THEORY OF COMPUTING, Volume 14 (1), 2018, pp. 1–27
www.theoryofcomputing.org

Linear-Time Algorithm for Quantum 2SAT

Itai Arad* Miklos Santha*† Aarthi Sundaram*‡ Shengyu Zhang*§

Received May 16, 2016; Revised August 22, 2017; Published March 9, 2018

Abstract: A well-known result about satisfiability theory is that the 2-SAT problem can be solved in linear time, despite the NP-hardness of the 3-SAT problem. In the quantum 2-SAT problem, we are given a family of 2-qubit projectors Π_{ij} on a system of n qubits, and the task is to decide whether the Hamiltonian $H = \sum \Pi_{ij}$ has a 0-eigenvalue, or all eigenvalues are greater than $1/n^{\alpha}$ for some $\alpha = O(1)$. The problem is not only a natural extension of the classical 2-SAT problem to the quantum case, but is also equivalent to the problem of finding a ground state of 2-local frustration-free Hamiltonians of spin $1/2$, a well-studied model believed to capture certain key properties in modern condensed matter physics. Bravyi has shown that the quantum 2-SAT problem has a deterministic algorithm of complexity $O(n^4)$ in the algebraic model of computation where every arithmetic operation on complex numbers can be performed in unit time, and n is the number of variables. In this paper we give a deterministic algorithm in the algebraic model with running time $O(n+m)$, where m is the number of local projectors, therefore achieving the best possible complexity in that model. We also show that if in the input every number has a constant size representation then the bit complexity of our algorithm is $O((n+m)M(n))$, where $M(n)$ denotes the complexity of multiplying two n-bit integers.

ACM Classification: F.2

AMS Classification: 68Q25, 81P68

Key words and phrases: quantum satisfiability, davis-putnam procedure, linear time algorithm

A conference version of this paper appeared in the Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP'16), 2016 [1].

*Supported by the Singapore Ministry of Education and the National Research Foundation, also under the Tier 3 Grant MOE2012-T3-1-009; by the European Commission IST STREP project Quantum Algorithms (QALGO) 600700; and the French ANR Blanc Program Contract ANR-12-BS02-005.

†Also thanks STIAS for their hospitality during October and November 2015.

‡Also thanks the Centre for Quantum Technologies for their support during my graduate studies when this work was done.

§Supported in part by RGC of Hong Kong (Project no. CUHK419413 and 14239416).

DOI: 10.4086/toc.2018.v014a001

1 Introduction

Various formulations of the satisfiability problem of Boolean formulae arguably constitute the centerpiece of classical complexity theory. In particular, a great amount of attention has been paid to the SAT problem, in which we are given a formula in the form of a conjunction of *clauses*, where each clause is a disjunction of *literals* (variables or negated variables), and the task is to find a satisfying assignment if there is one, or determine that none exists when the formula is unsatisfiable. In the case of the k-SAT problem, where k is a positive integer, the number of literals in each clause is at most k. While k-SAT is an NP-complete problem [7, 16, 21] when $k \geq 3$, the 2-SAT problem is well-known to be efficiently solvable.

Polynomial time algorithms for 2-SAT come in various flavours. Let us suppose that the input formula has n variables and m clauses. The algorithm of Krom [19] based on the resolution principle and on transitive closure computation decides if the formula is satisfiable in time $O(n^3)$ and finds a satisfying assignment in time $O(n^4)$. The limited backtracking technique of Even, Itai and Shamir [11] has linear time complexity in m, as well as the elegant procedure of Aspvall, Plass and Tarjan [2] based on computing strongly connected components in a graph. A particularly simple randomized procedure of complexity $O(n^2)$ is described by Papadimitriou [22].

For our purposes the Davis-Putnam procedure [9] is of singular importance. This is a resolution-principle based general SAT solving algorithm, which with its refinement due to Davis, Putnam, Logemann and Loveland [8], forms the basis for the most efficient SAT solvers even today. While on general SAT instances it works in exponential time, on 2-SAT formulae it is of polynomial complexity.

The high level description of the procedure for 2-SAT is relatively simple. Let us suppose that our formula ϕ only contains clauses with two literals. Pick an arbitrary unassigned variable x_i and assign $x_i = 0$. The formula is simplified: a clause $(\bar{x}_i \vee x_j)$ becomes true and therefore can be removed, and a clause $(x_i \vee x_j)$ forces $x_j = 1$. This can be, in turn, propagated to other clauses to further simplify the formula until a contradiction is found or no more propagation is possible. If no contradiction is found and the propagation stops with the simplified formula ϕ_0, then we recurse on the satisfiability of ϕ_0. Otherwise, when a contradiction is found, that is, at some point the propagation assigns two different values to the same variable, we reverse the choice made for x_i, and propagate the new choice $x_i = 1$. If this also leads to a contradiction we declare ϕ to be unsatisfiable, otherwise we recurse on the result of this propagation, the simplified formula ϕ_1.

There is a deep and profound link between k-SAT formulas and k-local Hamiltonians, the central objects of condensed matter physics. A k-local Hamiltonian on n qubits is a Hermitian operator of the form $H = \sum_{i=1}^{m} h_i$, where each h_i is individually a Hermitian operator acting non-trivially on at most k qubits. Local Hamiltonians model the local interactions between quantum spins. Of central importance are the eigenstates of the Hamiltonian that correspond to its minimal eigenvalue. These are called *ground states*, and their associated eigenvalue is known as the *ground energy*. Ground states govern much of the low temperature physics of the system, such as quantum phase transitions and collective quantum phenomena [24, 25]. Finding a ground state of a local Hamiltonian shares important similarities with the k-SAT problem: in both problems we are trying to find a global minimum of a set of local constraints. The connection with complexity theory is also of physical significance. With the advent of quantum information theory and quantum complexity theories, it has become clear that the complexity of finding a ground state and its energy is intimately related to its entanglement structure. In recent years, much attention has been devoted to understanding this structure, revealing a rich and intricate behaviour such as area laws [10] and topological order [17].

The connection between classical k-SAT and quantum local Hamiltonians was formalized by Kitaev [18] who introduced the k-local Hamiltonian problem. We are given a k-local Hamiltonian H, along with two constants $a < b$ such that $b - a > 1/n^{\alpha}$ for some constant α, and a promise that the ground energy is at most a (the YES case) or is at least b (the NO case). Our task is to decide which case holds. Given a quantum state $|\psi\rangle$, the energy of a local term $\langle\psi|h_i|\psi\rangle$ can be viewed as a measure of how much $|\psi\rangle$ "violates" h_i, hence the ground energy is the quantum analog of the minimal number of violations in a classical k-SAT formula. Therefore, in spirit, the k-local Hamiltonian problem corresponds to MAX-k-SAT, and indeed Kitaev has shown [18] that 5-local Hamiltonian is QMA-complete, where the complexity class QMA is the quantum analogue of classical class MA, the probabilistic version of NP.

The problem *quantum* k-SAT, the quantum analogue of k-SAT, is a close relative of the k-local Hamiltonian problem. Here we are given a k-local Hamiltonian that is made of k-local *projectors*, $H = \sum_{i=1}^{m} Q_i$, and we are asked whether the ground energy is 0 or it is greater than $b = 1/n^{\alpha}$ for some constant α. Notice that in the YES case, the energy of each projector at a ground state is necessarily 0 because, by definition, projectors are non-negative operators. Classically, this corresponds to a perfectly satisfiable formula. Physically, this is an example of a *frustration-free* Hamiltonian, in which a global ground state is also a ground state of every local term. Bravyi [4] has shown that quantum k-SAT is QMA_1-complete for $k \geq 4$, where QMA_1 stands for QMA with one-sided error (that is on YES instances the verifier accepts with probability 1). The QMA_1-completeness of quantum 3-SAT was recently proven by Gosset and Nagaj [12].

This paper is concerned with the quantum 2-SAT problem, which we will also denote simply by Q2SAT. One major result concerning this problem is due to Bravyi [4], who has proven that it belongs to the complexity class P. More precisely, he has proven that Q2SAT can be decided by a deterministic algorithm in time $O(n^4)$ in the *algebraic model of computation*, where every arithmetic operation on complex numbers can be performed in unit time. In addition, on satisfiable instances he could construct a ground state that has a polynomial classical description. In the case of Q2SAT, the Hamiltonian is given as a sum of 2-qubits projectors; each projector is defined on a 4-dimensional Hilbert space and can therefore be of rank 1, 2 or 3. In this paper, we give an algorithm for Q2SAT of *linear* complexity.

Theorem 1.1. *In the algebraic model of computation there is a deterministic algorithm for* Q2SAT *whose running time is $O(n+m)$, where n is the number of variables and m is the number of local terms in the Hamiltonian.*

Our algorithm shares the same trial and error approach of the Davis-Putnam procedure for classical 2-SAT, but handles many of the difficulties arising in the quantum setting. Firstly, a ground state of the input to Q2SAT may be entangled, a feature that classical 2-SAT does not have. Thus the idea of setting some qubit to a certain state and propagating from there does not have foundation in the first place. Indeed, if a rank-3 projection leaves the only allowed state entangled, then any ground state is entangled in those two qubits. We account for this by showing a *product-state theorem*, which asserts that for any frustration-free Q2SAT instance H that contains only rank-1 and rank-2 projectors, there always exists a ground state in the form of a tensor product of *single-qubit* states.

This structural theorem grants us the following approach: We try some candidate solution $|\psi\rangle_i$ on a qubit i, and propagate it along the graph. If no contradiction is found, it turns out that we can detach the explored part and recurse on the rest of the graph. If a contradiction is found, then we can identify two candidates $(i, |\psi\rangle_i)$ and $(j, |\phi\rangle_j)$ such that either assigning $|\psi\rangle_i$ to qubit i or assigning $|\phi\rangle_j$ to qubit j is correct, if there exists a solution at all. More details follow next.

To illustrate the main idea of our algorithm, let us assume, for simplicity, that the system is only made of rank-1 projectors. Consider, then a rank-1 projector in the system, say, $\Pi_{12} = |\psi\rangle\langle\psi|$ over qubits 1 and 2. The product-state theorem implies that it suffices to search for a product ground state. Thus on the first two qubits, we are looking for states $|\alpha\rangle, |\beta\rangle$ such that $(\langle\alpha| \otimes \langle\beta|)\Pi_{12}(|\alpha\rangle \otimes |\beta\rangle) = 0$, which is equivalent to $\langle\alpha| \otimes \langle\beta| \cdot |\psi\rangle = 0$. In other words, we look for a product state $|\alpha\rangle \otimes |\beta\rangle$ that is perpendicular to $|\psi\rangle$. Assume that we have assigned qubit 1 with the state $|\alpha\rangle$ and we are looking for a state $|\beta\rangle$ for qubit 2. The important point, which enables us to solve Q2SAT efficiently, is that just like in the classical case, there are only two possibilities: (i) for any $|\beta\rangle$, the state $|\alpha\rangle \otimes |\beta\rangle$ is perpendicular to $|\psi\rangle$, or (ii) there is only one state $|\beta\rangle$ (up to an overall complex phase), for which $(\langle\alpha| \otimes \langle\beta|) \cdot |\psi\rangle = 0$. The first case happens if and only if $|\psi\rangle$ is by itself a product state of the form $|\psi\rangle = |\alpha^\perp\rangle \otimes |\xi\rangle$, where $|\alpha^\perp\rangle$ is perpendicular to $|\alpha\rangle$ and $|\xi\rangle$ is arbitrary. If the second case happens, we say that the state $|\alpha\rangle$ is propagated to the state $|\beta\rangle$ by the constraint state $|\psi\rangle$.

The above dichotomy enables us to propagate a product state $|s\rangle$ on part of the system until we either reach a contradiction, or find that no further propagation is possible and we are left with a smaller Hamiltonian H'. This smaller Hamiltonian consists of a subset of the original projectors *without introducing new projectors*. This crucial fact implies that, analogous to the classical case, the original Hamiltonian H is satisfiable if and only if the smaller Hamiltonian H' is satisfiable.

We still need to specify how the state $|\alpha\rangle$ is chosen to initialize the propagation. An idea is to begin with projectors $|\psi\rangle\langle\psi|$ for which $|\psi\rangle$ is a product state $|\gamma\rangle \otimes |\delta\rangle$. In such cases a product state solution must either have $|\gamma^\perp\rangle$ at the first qubit or $|\delta^\perp\rangle$ at the second. To maintain a linear running time, we propagate these two choices simultaneously until one of the propagations stops without contradiction, in which case the corresponding qubit assignment is made final. If both propagations end with contradictions, the input is rejected.

The more interesting case of the algorithm happens when we have only entangled rank-1 projectors. What should our initial state be then? We make an arbitrary assignment (say, $|0\rangle$) to any of the still unassigned qubits and propagate this choice. If the propagation ends without contradiction, we recurse. If a contradiction is found then we confront a challenging problem. In the classical case we could reverse our choice, say $x_0 = 0$, and try the other possibility, $x_0 = 1$. But in the quantum case we have an infinite number of potential assignment choices. The solution is found by the following observation: Whenever a contradiction is reached, it can be attributed to a cycle of entangled projectors in which the assignment has propagated from qubit i along the cycle and returned to it with another value (see Figure 1(a)). Then using the technique of "sliding," which was introduced in Ref. [15], one can show that this cycle is equivalent to a system of one double edge and a "tail" (see Figure 1(b)). Using a simple structure lemma, we are guaranteed that at least one of the projectors of the double edge can be turned into a product state projector, which, as in the previous stage, gives us only two possible free choices.

As we have stated, our algorithm works in the algebraic model of computation: we suppose that every arithmetic operation on complex numbers can be done in unit time. There are several ways to work in a more realistic model. One possibility would be to consider complex numbers with bounded precision in which case exact computation is no more possible and therefore an error analysis should be

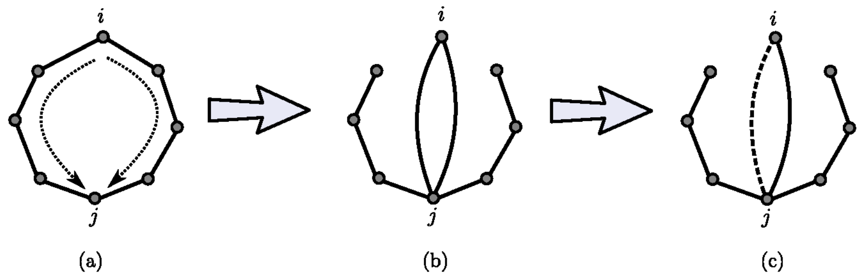


Figure 1: Handling a contradicting cycle: (a) we slide edges that touch i along the two paths to j until (b) we get a double edge with two “tails.” (c) we use a structure lemma to deduce that at least one of these edges can be written as a product projector (a dashed edge).

made. Bravyi [4] suggests considering bounded degree algebraic numbers, in which case the length of the representations and the cost of the operations can be analyzed by considering their bit-wise costs, termed the *bit complexity*. For the sake of completeness, following the thread of working within the framework of bounded algebraic numbers, we provide an analysis of the bit complexity of our algorithm in the final section of the paper. More precisely, we prove that if in the input every number has a constant size representation then the bit complexity of our algorithm is $O((n+m)\ M(n))$, where $M(n)$ denotes the complexity of multiplying two n-bit integers. Finding an algorithm with a better bit complexity seems to be a hard problem and we leave this challenge as an open problem.

Classically, Davis-Putnam [9] and DPLL algorithms [8] are widely-used heuristics, forming the basis of today’s most efficient solvers for general SAT. For quantum k-SAT, it could also be a good heuristic if we try to find product-state solutions, and in that respect our algorithm makes the first-step exploration.

Simultaneously albeit independently from our work de Beaudrap and Gharibian [3] also presented a linear time algorithm for quantum 2SAT. The main difference between the two algorithms is how they deal with instances having only entangled rank-1 projectors. Contrarily to us, [3] handles these instances by using transfer matrix techniques to find discretizing cycles [20].

2 Preliminaries

2.1 Notation

We will use the notation $[n] = \{1,\dots,n\}$. For a graph $G = (V,E)$, and for a subset $U \subseteq V$ of the vertices, we denote by $G(U)$ the subgraph induced by U. Our Hilbert space is defined over n qubits, and is written as $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_n$, where $\mathcal{H}_i$ is the two-dimensional Hilbert space of the i^{th} qubit. We shall often write $|\alpha\rangle_i$ to emphasize that the 1-qubit state $|\alpha\rangle$ lives in $\mathcal{H}_i$. Similarly, $|\psi\rangle_{ij}$ denotes a 2-qubit state that lives in $\mathcal{H}_i \otimes \mathcal{H}_j$. For a 1-qubit state $|\alpha\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$, we define its *perpendicular state* as $|\alpha^\perp\rangle = \bar{\alpha}_1|0\rangle - \bar{\alpha}_0|1\rangle$, where $\bar{\alpha}$ denotes the complex conjugate of α.

A 2-*local projector* on qubits $i \neq j$ is a projector of the form $\Pi_{ij} = \hat{\Pi}_{ij} \otimes \mathbb{I}_{\text{rest}}$, where $\hat{\Pi}_{ij}$ is a 2-qubit projector working on $\mathcal{H}_i \otimes \mathcal{H}_j$ and $\mathbb{I}_{\text{rest}}$ is the identity operator on the rest of the system. Similarly, a 1-*local projector* on qubit i is a projector of the form $\Pi_{ii} = \hat{\Pi}_{ii} \otimes \mathbb{I}_{\text{rest}}$, where $\hat{\Pi}_{ii}$ is a 1-qubit projector working on $\mathcal{H}_i$ and $\mathbb{I}_{\text{rest}}$ is the identity operator on the rest of the system. We define the *rank* of a 2-local projector $\Pi_{ij} = \hat{\Pi}_{ij} \otimes \mathbb{I}_{\text{rest}}$ to be the dimension of the subspace that $\hat{\Pi}_{ij}$ projects to, and we denote it by $\texttt{rank}(\Pi_{ij})$. The rank of a 1-local projector is defined analogously. The 2-local projectors of rank 3 and the 1-local projectors of rank 1 are considered to be of *maximal* rank. A 2-local projector $\Pi_{ij} = \hat{\Pi}_{ij} \otimes \mathbb{I}_{\text{rest}}$ of rank 1 where $\hat{\Pi}_{ij} = |\psi\rangle\langle\psi|_{ij}$ is called *entangled* if $|\psi\rangle$ is an entangled state, and it is called a *product* projector if $|\psi\rangle$ is a product state.Recall that a 2-qubit state $|\psi\rangle_{ij}$ is a product state if it can be written as a tensor product of 2 single-qubit states $|\psi\rangle_{ij} = |\psi_1\rangle_i \otimes |\psi_2\rangle_j$; and a 2-qubit state which is not a product state is an entangled state. Additionally, given a 2-local projector $\hat{\Pi}_{ij}$, it is possible to determine if it is entangled using the Peres-Horodecki criterion [14, 23].

2.2 The Q2SAT problem

In this paper, we define a 2-*local Hamiltonian* on an n-qubit system to be a Hermitian operator $H = \sum_{e \in I} \Pi_e$, for some $I \subseteq \{(i,j) \in [n] \times [n] : 1 \le i \le j \le n\}$, where Π_e is a 2-local or a 1-local projector, for $e \in I$. We suppose that $\texttt{rank}(\Pi_{ii}) = 1$, for all $(i,i) \in I$, and $0 < \texttt{rank}(\Pi_{ij}) < 4$, for all $(i,j) \in I$ when $i < j$. We also suppose that for every $i \in [n]$, there is some $e \in I$ such that Π_e acts on qubit i.

The *ground energy* of a Hamiltonian $H = \sum_{e \in I} \Pi_e$ is its smallest eigenvalue, and a *ground state* of H is an eigenvector corresponding to the smallest eigenvalue. The subspace of the ground states is called the *ground space*. A Hamiltonian is *frustration-free* if it has a ground state that is simultaneously also the ground state of each local term. As explained in the introduction, if the Hamiltonian is made of local projectors, it is frustration-free if and only if there is a state that is a mutual zero eigenstate of all projectors, which happens if and only if the ground energy is 0. Therefore, if $|\Gamma\rangle$ is a ground state of a frustration-free 2-local Hamiltonian then $\Pi_e|\Gamma\rangle = 0$ for all $e \in I$. We can also view each local projector as a *constraint* on at most two qubits, then a ground state of a frustration free Hamiltonian satisfies every constraint.

It turns out that for the representation of a 2-local Hamiltonian, it will be helpful to eliminate the rank-2 projectors by decomposing each one of them into a sum of two rank-1 projectors. For every $(i,j) \in I$ such that $\texttt{rank}(\Pi_{ij}) = 2$, let $\Pi_{ij} = \Pi_{ij,1} + \Pi_{ij,2}$, where $\Pi_{ij,1}$ and $\Pi_{ij,2}$ are rank-1 projectors. Such projectors can be found in constant time. We therefore suppose without loss of generality that H is specified by

$$H = \sum_{\texttt{rank}(\Pi_{ij}) \neq 2} \Pi_{ij} + \sum_{\texttt{rank}(\Pi_{ij}) = 2} (\Pi_{ij,1} + \Pi_{ij,2}),$$

which we call the *rank*-1 *decomposition* of H.

To the rank-1 decomposition we associate a weighted, directed multigraph with self-loops $G(H) = (V, E, w)$, which we call the *constraint graph* of H. By definition

$$V = \{i \in [n] : \exists j \in [n] \text{ such that } (i,j) \in I \text{ or } (j,i) \in I\}.$$

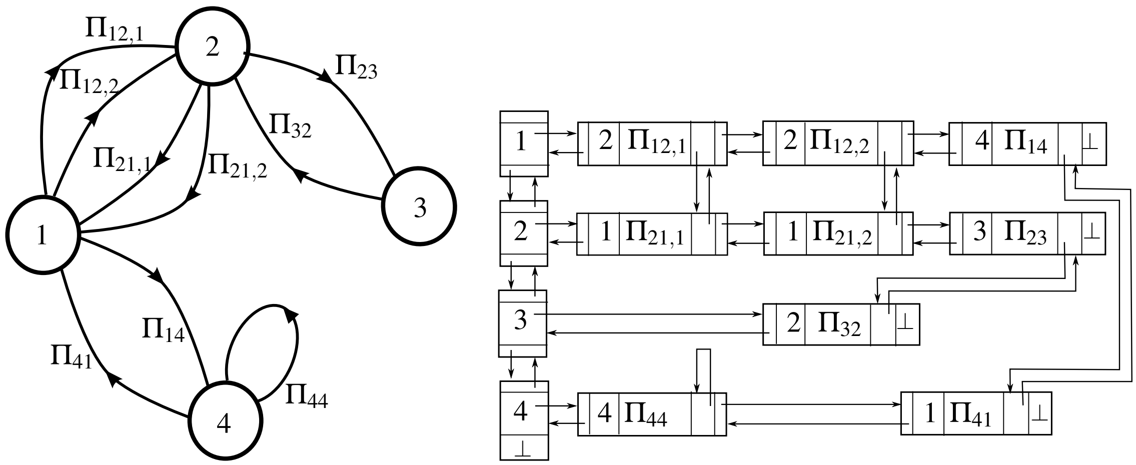
For every rank-3 and rank-1 projector acting on two qubits, there is an edge in each direction between the two nodes representing them. For every projector acting on a single qubit, there is a self-loop. Finally, for every rank-2 projector, there are two parallel edges, one in each direction, between the nodes representing its qubits. Because of the parallel edges, E is not a subset of $V \times V$. Formally, $E = E_1 \cup E_2$ where

$$E_1 = \{(i,j) \in [n] \times [n] : (i,j) \in I \text{ and } \mathtt{rank}(\Pi_{ij}) \in \{1,3\}, \text{ or } (j,i) \in I \text{ and } \mathtt{rank}(\Pi_{ji}) \in \{1,3\}\},$$

and

$$E_2 = \{(i,j,b) \in [n] \times [n] \times [2] : (i,j) \in I \text{ and } \mathtt{rank}(\Pi_{ij}) = 2, \text{ or } (j,i) \in I \text{ and } \mathtt{rank}(\Pi_{ji}) = 2\}.$$

We say that an edge $e \in E$ *goes from i to j* if $e \in \{(i,j),(i,j,1),(i,j,2)\}$. We define e^{rev}, the *reverse* of the edge e, by $(i,j)^{\mathsf{rev}} = (j,i), (i,j,1)^{\mathsf{rev}} = (j,i,1)$ and $(i,j,2)^{\mathsf{rev}} = (j,i,2)$ respectively. For a 2-qubit projector $\hat{\Pi}$ we define its *reverse* projector $\hat{\Pi}^{\mathsf{rev}}$ by $\hat{\Pi}^{\mathsf{rev}}|\alpha\rangle|\beta\rangle = \hat{\Pi}|\beta\rangle|\alpha\rangle$. For $i < j$ and $b \in [2]$, if $\Pi_{ij} = \hat{\Pi}_{ij} \otimes \mathbb{I}_{\text{rest}}$, then we set $\Pi_{ji} = \hat{\Pi}^{\mathsf{rev}}_{ij} \otimes \mathbb{I}_{\text{rest}}$, and $\Pi_{ji,b}$ is defined analogously. The *weight* of the edges (i,j) and (i,j,b) are defined as $w(i,j) = \Pi_{ij}$, and $w(i,j,b) = \Pi_{ij,b}$.



(a) (b)

Figure 2: (a) The constraint graph for Hamiltonian $H = \Pi_{12} + \Pi_{14} + \Pi_{23} + \Pi_{44}$ where $\mathtt{rank}(\Pi_{44}) = \mathtt{rank}(\Pi_{23}) = 1, \mathtt{rank}(\Pi_{12}) = 2$ and $\mathtt{rank}(\Pi_{14}) = 3$ using its rank-1 decomposition. (b) The adjacency list representation for the constraint graph $G(H)$

We will suppose that the input to our problem is the constraint graph $G(H)$ of the Hamiltonian, given in the standard adjacency list representation of weighted graphs, naturally modified for dealing with the parallel edges as shown in Figure 2. In this representation there is a doubly linked list of size at most n containing one element for each vertex, and the element i in this list is also pointing towards a doubly linked list containing an element for every edge going from i to j. For an edge (i,j), this element contains j, the non-trivial part of projector Π_{ij}, $\hat{\Pi}_{ij}$, and a pointer towards the next element in the list and for an edge (i,j,b) it also contains the value b. We also suppose that for every edge e, there is a double link between the elements representing e and e^{rev}. The problem Q2SAT is defined formally as follows.

Q2SAT

Input: The constraint graph $G(H)$ of a 2-local Hamiltonian H, given in the adjacency list representation.

Output: A solution if H is frustration free, "H is unsatisfiable" if it is not.

2.3 Simple ground states

Our algorithm is based crucially on the following *product state theorem*, which says that any frustration-free Q2SAT Hamiltonian has a ground state that is a product state of single qubit and two-qubit states, where the latter only appear in the support of rank-3 projectors. A slightly weaker claim of that form has already appeared in Theorem 2 of Ref. [5]. The difference here is that we specifically attribute the 2-qubit states in such a state to rank-3 projectors. Just as in Ref. [5], our derivation relies on the notion of a *genuinely entangled state:*

Definition 2.1 (Genuinely entangled states)**.** A state $|\psi\rangle$ over n qubits is genuinely entangled if for any bi-partition of the qubits into two subsets A,B, it cannot be written as a product state $|\psi\rangle = |\psi_A\rangle \otimes |\psi_B\rangle$, where $|\psi_A\rangle$, $|\psi_B\rangle$ are defined on the qubits of A and B respectively.

Using this definition, Theorem 1 of [5] is restated below.

Proposition 2.2. *Any* 2*-local frustration-free Hamiltonian on* $n \geq 3$ *qubits that has a genuinely entangled ground state also has a ground state, thats is a product of one-qubit and two-qubits states.*

We will also need the following fact about 2-dimensional subspaces in $\mathbb{C}^2 \otimes \mathbb{C}^2$.

Proposition 2.3. *Any* 2*-dimensional subspace* V *of the* 2*-qubit space* $\mathbb{C}^2 \otimes \mathbb{C}^2$ *contains at least one product state.*

Proof. Take a basis $\{|\psi\rangle, |\phi\rangle\}$ of the two-dimensional subspace $V^\perp$, the orthogonal complement of V. Our goal is to find a product state $|\alpha\rangle \otimes |\beta\rangle \in V$ such that $\langle\psi|(|\alpha\rangle \otimes |\beta\rangle) = \langle\phi|(|\alpha\rangle \otimes |\beta\rangle) = 0$. To that aim, expand $|\psi\rangle, |\alpha\rangle$ and $|\beta\rangle$ in the standard basis as $|\psi\rangle = \sum_{ij} \psi_{ij}|ij\rangle$, $|\phi\rangle = \sum_{ij} \phi_{ij}|ij\rangle$, and $|\alpha\rangle = \sum_i \alpha_i|i\rangle$, $|\beta\rangle = \sum_j \beta_j|j\rangle$. Then we need to find coefficients α_i and β_j such that $\sum_{ij} \phi^*_{ij} \cdot \alpha_i\beta_j = 0$ and $\sum_{ij} \psi^*_{ij} \cdot \alpha_i\beta_j = 0$. We can move to a matrix notation, in which ψ^*_{ij}, ϕ^*_{ij} are the entries of 2×2 matrices Ψ, Φ, and α_i, β_j are the coordinates of the 2-vectors $\underline{\alpha}, \underline{\beta}$. In that notation, we are looking for vectors $\underline{\alpha}, \underline{\beta}$ such that

$$\underline{\alpha}^T \Psi \underline{\beta} = \underline{\alpha}^T \Phi \underline{\beta} = 0\,. \tag{2.1}$$

If the matrix Φ is singular, we pick $\underline{\beta}$ inside its the null space, and choose $\underline{\alpha}$ such that $\underline{\alpha}^T\Psi\underline{\beta} = 0$. Otherwise, when Φ is non-singular, we let $\underline{\beta}$ be a right eigenvector of the matrix $\Phi^{-1}\Psi$, i. e., $\Phi^{-1}\Psi\underline{\beta} = c\underline{\beta}$, where c is some eigenvalue. Then $\Psi\underline{\beta} = c\Phi\underline{\beta}$, and therefore to satisfy equation (2.1), we can choose $\underline{\alpha}$ such that $\underline{\alpha}^T\Phi\underline{\beta} = 0$. $\square$

For later use we note that the above proof is constructive, implying that the product state can be found in constant time. Our product state theorem is stated as follows.

Theorem 2.4. *Any frustration-free* Q2SAT *Hamiltonian* $H = \sum_{e \in I} \Pi_e$ *has a ground state that is a tensor product of one qubit and two-qubit states, where two-qubit states only appear in the support of rank-*3 *projectors.*

Proof. Consider a frustration-free 2-local Hamiltonian H and let $|\Gamma\rangle$ be a ground state of H. Much like any natural number can be written as a product of prime numbers, using Definition 2.1, any state over n qubits can be written as a product state of one or more genuinely entangled states. In particular, $|\Gamma\rangle$ can be written as a product state

$$|\Gamma\rangle = |\alpha^{(1)}\rangle \otimes |\alpha^{(2)}\rangle \otimes \cdots \otimes |\alpha^{(r)}\rangle ,$$

where each $|\alpha^{(i)}\rangle$ is a genuinely entangled state defined on a subset $S^{(i)}$ of qubits. Notice if H contains a rank-3 projector $\Pi_{jk} = \mathbb{I} - |\psi\rangle\langle\psi|_{jk}$, then necessarily *every* ground state of H will contain $|\psi\rangle_{jk}$ at a tensor product with the rest of the system. Therefore, if $|\psi\rangle_{jk}$ is entangled, there must exist a subset $S^{(i)} = \{j,k\}$ in the above decomposition with $|\alpha^{(i)}\rangle = |\psi\rangle_{jk}$. Similarly, if $|\psi\rangle_{jk}$ is a product state, there must exist two subsets $S^{(i_1)} = \{j\}$, and $S^{(i_2)} = \{k\}$. Consequently, if $S^{(i)}$ has more than two qubits, there cannot be any rank-3 projector defined on these qubits.

Consider now a subset $S^{(i)}$ that either: (i) contains three or more qubits, or (ii) contains exactly two qubits, but there does not exists a rank-3 projector that is defined on these qubits. Define $H^{(i)}$ to be the Hamiltonian that is the sum of all projectors whose support is inside $S^{(i)}$. By definition and from the above paragraph, $H^{(i)}$ does not contain a rank-3 projector. Therefore we can use either Proposition 2.2 (when $S^{(i)}$ has three or more qubits) or Proposition 2.3 (when $S^{(i)}$ has two qubits) to deduce that in addition to $|\alpha^{(i)}\rangle$, $H^{(i)}$ has a ground state $|\beta^{(i)}\rangle$ that is a product state of *single* qubit states:

$$|\beta^{(i)}\rangle = |\beta_1^{(i)}\rangle \otimes |\beta_2^{(i)}\rangle \otimes \cdots . \tag{2.2}$$

The remaining $S^{(i)}$ are either single qubits subsets, or 2-qubits subsets that match the support of entangled rank-3 projectors. For all these cases, we define $|\beta^{(i)}\rangle = |\alpha^{(i)}\rangle$.

We now claim that the state $|\beta\rangle = |\beta^{(1)}\rangle \otimes \cdots \otimes |\beta^{(r)}\rangle$, which is a product of one-qubit and two-qubit states, is a ground state of the full Hamiltonian $H = \sum_{e \in I} \Pi_e$, including terms that act across the different $S^{(i)}$. For that, we need to show that $\Pi_e|\beta\rangle = 0$ for every projector Π_e in H. If the support of Π_e is inside one of the S_i subsets, then by definition $\Pi_e|\beta^{(i)}\rangle = 0$ and therefore $\Pi_e|\beta\rangle = 0$. Assume then that Π_e is supported on a qubit from $S^{(i)}$ and a qubit from $S^{(j)}$ with $i \neq j$. We now consider 3 cases:

1. If both $S^{(i)}$ and $S^{(j)}$ contain only one qubit then $\Pi_e|\beta^{(i)}\rangle \otimes |\beta^{(j)}\rangle = \Pi_e|\alpha^{(i)}\rangle \otimes |\alpha^{(j)}\rangle = 0$.

2. If $S^{(i)}$ is made of one qubit but $S^{(j)}$ has two or more qubits, then expand $|\alpha^{(j)}\rangle = \lambda_0|0\rangle \otimes |y_0\rangle + \lambda_1|1\rangle \otimes |y_1\rangle$. Here, the standard basis vectors $|0\rangle, |1\rangle$ are defined on the qubit of S_j that is in the support of Π_e, while $|y_0\rangle, |y_1\rangle$ are defined on the rest of the qubits in S_j, and are not necessarily orthogonal. The vectors $\lambda_0|y_0\rangle, \lambda_1|y_1\rangle$ are by assumption linearly independent, as otherwise $|\alpha_j\rangle$

could have been written as a product state, violating the assumption that it is genuinely entangled. Then, as the condition $\Pi_e|\alpha^{(i)}\rangle \otimes |\alpha^{(j)}\rangle = 0$ is equivalent to

$$\big(\Pi_e|\alpha^{(i)}\rangle \otimes |0\rangle\big) \otimes \lambda_0|y_0\rangle + \big(\Pi_e|\alpha^{(i)}\rangle \otimes |1\rangle\big) \otimes \lambda_1|y_1\rangle = 0\,,$$

we conclude that $\Pi_e|\alpha^{(i)}\rangle \otimes |0\rangle = \Pi_e|\alpha^{(i)}\rangle \otimes |1\rangle = 0$. Therefore, Π_e annihilates the subspace $|\alpha^{(i)}\rangle \otimes \mathbb{C}^2$, and in particular it annihilates $|\beta^{(i)}\rangle \otimes |\beta^{(j)}\rangle$ because $|\beta^{(i)}\rangle = |\alpha^{(i)}\rangle$.

3. The third case in which both $S^{(i)}$ and $S^{(j)}$ contain two or more qubits cannot happen. Indeed, in this case we expand the parts of $|\alpha^{(i)}\rangle, |\alpha^{(j)}\rangle$ that are in the support of Π_e in the standard basis, and repeating the argument from the previous case, we conclude that Π_e must annihilate 4 independent vectors. It therefore cannot be a rank-1 or a rank-2 projector.

This completes the proof of the theorem. □

2.4 Assignments

Let $H = \sum_{e\in I} \Pi_e$ be a 2-local Hamiltonian. By Theorem 2.4, if H is frustration free then it has a ground state that is the tensor product of 1-qubit and 2-qubit entangled states, where the latter only appear in pairs of qubits in the support of rank-3 projectors. To build up a ground state of this form, our algorithm will use partial assignments (or simply, *assignments*, for short). An *assignment* s is a mapping from $[n]$. For every $i \in [n]$, the value $s(i)$ is either a 1-qubit state $|\alpha\rangle$, or a 2-qubit entangled state $|\gamma\rangle_{ij}$ for some $j \neq i$, or the symbol $\ominus$. If $s(i) = |\alpha\rangle$ or $s(i) = |\gamma\rangle_{ij}$, then this value is assigned to qubit variable i, and in the latter case the entangled state is shared with variable j. If in an assignment $s(i) = |\gamma\rangle_{ij}$, we require that $s(j) = |\gamma\rangle_{ij}$. The symbol $\ominus$ is used for unassigned variables. It is common practice to consider normalized quantum states, that is states $|\alpha\rangle$ such that $\langle\alpha|\alpha\rangle = 1$. However, in the course of our algorithm, we will deal with and assign to the variables un-normalized states. This does not affect the accuracy of the algorithm but can result in an un-normalized ground state as the output. It is possible to additionally normalize the ground state without affecting the running time of the algorithm.

We define the *support* of s by $\mathrm{supp}(s) = \{i \in [n] : s(i) \neq \ominus\}$. The assignment s is *empty* if $\mathrm{supp}(s) = \emptyset$. We denote the empty assignment by $s_\ominus$. For assignments s and s', we say that s' is an *extension* of s, if for every i, such that $s(i) \neq \ominus$, we have $s'(i) = s(i)$. An assignment is *total* if $s(i) \neq \ominus$, for all i. Clearly, an assignment defines a product state of 1-qubit and 2-qubits states on qubits in its support. We denote this state by $|s\rangle$. We say that an assignment s *satisfies* a projector Π_e, or simply that it satisfies the edge e if, for any total extension s' of s, we have $\Pi_e|s'\rangle = 0$.

For $H = \sum_{e\in I} \Pi_e$ given in rank-1 decomposition, and an assignment s, we define the *reduced Hamiltonian* H_s of s as

$$H_s = H - \sum_{s \text{ satisfies } e} \Pi_e\,.$$

We will denote the constraint graph $G(H_s)$ of the reduced Hamiltonian H_s by $G_s = (V_s, E_s)$. We call an assignment s a *pre-solution* if it has a total extension s' satisfying every constraint in H, and we call s a *solution* if s itself satisfies every constraint in H. Obviously, an assignment is a solution if and only if G_s is the empty graph. An assignment s is *closed* if $\mathrm{supp}(s) \cap V_s = \emptyset$.

3 Propagation

The crucial building block of our algorithm is the propagation of values by rank-1 projectors. This is the quantum analog of the classical propagation process when, for example, the clause $x_i \vee x_j$ propagates the value $x_i = 0$ to the value $x_j = 1$ in the sense that given $x_i = 0$, the choice $x_j = 1$ is the only possibility to make the clause true. In the quantum case this notion has already appeared in Ref. [20], and can, in fact, also be traced back to Bravyi's original work. Here, we shall adopt the following definition:

Definition 3.1 (Propagation)**.** Let $\Pi_e = |\psi\rangle\langle\psi|$ be a rank-1 projector acting on variables i, j, and let $|\alpha\rangle$ either be a 1-qubit state assigned to variable i, or a 2-qubit entangled state assigned to variables k, i for some $k \neq j$. We say that Π_e *propagates* $|\alpha\rangle$ if, up to an arbitrary complex phase $e^{i\theta}$, there exists a unique 1-qubit state $|\beta\rangle$ such that $\Pi_e|\alpha\rangle \otimes |\beta\rangle_j = 0$. In this case we say that $|\alpha\rangle$ is propagated to $|\beta\rangle$ along Π_e, or that Π_e propagated $|\alpha\rangle$ to $|\beta\rangle$.

The following lemma shows how the propagation properties of $\Pi_e = |\psi\rangle\langle\psi|$ are determined by the entanglement in $|\psi\rangle$.

Lemma 3.2. *Consider the rank-*1 *projector* $\Pi_e = |\psi\rangle\langle\psi|$*, defined on qubits* i, j*. If* $|\psi\rangle$ *is entangled, it propagates* every 1*-qubit state* $|\alpha\rangle_i$ *to a state* $|\beta(\alpha)\rangle_j$ *such that if* $|\alpha\rangle_i$ *is not a constant multiple of* $|\alpha'\rangle_i$ *then* $|\beta(\alpha)\rangle_j$ *is not a constant multiple of* $|\beta(\alpha')\rangle_j$*. When* $|\psi\rangle$ *is a product state* $|\psi\rangle = |x\rangle_i \otimes |y\rangle_j$*, the projector* Π_e *does not propagate states that are proportional to* $|x^\perp\rangle_i$*, while all other states are propagated to* $|y^\perp\rangle_j$*.*

Proof. Assume that $|\psi\rangle$ is entangled and consider the state $|\alpha\rangle$. Our task is to show that there always exists a unique $|\beta\rangle$ (up to an overall constant) such that $\Pi(|\alpha\rangle \otimes |\beta\rangle) = 0$, and that different $|\alpha\rangle$ vectors yield different $|\beta\rangle$ vectors.

Expanding $|\psi\rangle$, $|\alpha\rangle$, and $|\beta\rangle$ in the standard basis as $|\psi\rangle = \sum_{i,j} \psi_{ij}|i\rangle \otimes |j\rangle$; $|\alpha\rangle = \sum_i \alpha_i|i\rangle$; $|\beta\rangle = \sum_j \beta_j|j\rangle$, the condition $\Pi_e(|\alpha\rangle \otimes |\beta\rangle) = 0$ translates to $\sum_{i,j} \psi^*_{ij}\alpha_i\beta_j = 0$. Assuming that $|\psi\rangle$ is entangled, one can easily verify that the 2×2 matrix (ψ^*_{ij}) is non-singular. Then using the simple fact that in a two-dimensional space every non-zero vector has exactly one non-zero vector (up to an overall scaling) to which it is orthogonal, it is straightforward to deduce that for every non-zero vector (α_0, α_1) there is a unique (up to scaling) non-zero vector (β_0, β_1) such that $\sum_{i,j} \psi^*_{ij}\alpha_i\beta_j = 0$. Moreover, (β_0, β_1) can be calculated in constant time, and different (α_0, α_1) necessarily yield different (β_0, β_1).

The case when $|\psi\rangle$ is a product state is straightforward. □

As in the case of Proposition 2.3, we note that the proof above is constructive, and therefore the propagation can be calculated in constant time.

We now present two lemmas that describe the *global* structure of a ground state of the system, when part of it is known to be a tensor product of 1-qubit or 2-qubits states, which are then propagated by some Π_e.

Lemma 3.3 (Single qubit propagation)**.** *Consider a frustration-free* Q2SAT *system* $H = \sum_{e\in I} \Pi_e$ *with a rank-*1 *projector* $\Pi_e = |\psi\rangle\langle\psi|$ *between qubits* i, j*, and assume that* H *has a ground state of the form* $|\Gamma\rangle = |\alpha\rangle_i \otimes |\text{rest}\rangle$*, where* $|\alpha\rangle_i$ *is defined on qubit* i *and* $|\text{rest}\rangle$ *on the rest of the system. If* Π_e *propagates* $|\alpha\rangle_i$ *to* $|\beta\rangle_j$ *then necessarily* $|\text{rest}\rangle = |\beta\rangle_j \otimes |\text{rest}'\rangle$*, where the state* $|\text{rest}'\rangle$ *is defined on all the qubits of the system except for* i *and* j*.*

Proof. For the first claim assume that Π_e propagates $|\alpha\rangle_i$ to $|\beta\rangle_j$. We may expand the state $|\text{rest}\rangle$ as

$$|\text{rest}\rangle = |\beta\rangle_j \otimes |\text{rest}_1\rangle + |\beta^\perp\rangle_j \otimes |\text{rest}_2\rangle ,$$

where the states $|\text{rest}_1\rangle, |\text{rest}_2\rangle$ are defined on all the qubits of the system except for i and j, and are not necessarily normalized. Plugging this expansion into the condition $\Pi_e|\Gamma\rangle = 0$, we obtain the equation

$$(\Pi_e|\alpha\rangle_i|\beta\rangle_j) \otimes |\text{rest}_1\rangle + (\Pi_e|\alpha\rangle_i|\beta^\perp\rangle_j) \otimes |\text{rest}_2\rangle = 0 .$$

Π_e propagates $|\alpha\rangle_i$ to $|\beta\rangle_j$, so we have $\Pi_e|\alpha\rangle_i|\beta\rangle_j = 0$ and $\Pi_e|\alpha\rangle_i|\beta^\perp\rangle_j \neq 0$. Therefore, the above equation implies that $|\text{rest}_2\rangle = 0$, and we may set $|\text{rest}'\rangle = |\text{rest}_1\rangle$. □

Lemma 3.4 (Entangled 2-qubits propagation)**.** *Consider a frustration-free* Q2SAT *system H with a rank-*1 *projector $\Pi_e = |\psi\rangle\langle\psi|_{ij}$ between qubits i, j. Assume that H has a ground state of the form $|\Gamma\rangle = |\phi\rangle_{ik} \otimes |\text{rest}\rangle$, where $|\phi\rangle$ is an entangled state on qubits i,k with $k \neq j$ and $|\text{rest}\rangle$ is defined on all qubits except i and k. The following facts hold:*

1. *$|\psi\rangle$ is a product state $|\psi\rangle = |x\rangle_i|y\rangle_j$.*
2. *Π_e propagates $|\phi\rangle_{ik}$ to $|y^\perp\rangle_j$ and necessarily, up to an overall phase, $|\text{rest}\rangle = |y^\perp\rangle_j \otimes |\text{rest}'\rangle$ where $|\text{rest}'\rangle$ is defined on all qubits except i, k and j.*

Proof. Write $|\phi\rangle_{ik}$ in a Schmidt decomposition $|\phi\rangle_{ik} = \lambda_1|\alpha\rangle_i \otimes |\beta\rangle_k + \lambda_2|\alpha^\perp\rangle_i \otimes |\beta^\perp\rangle_k$, and note that both $\lambda_1, \lambda_2 \neq 0$, because $|\phi\rangle_{ik}$ is entangled. Plugging this into the condition $\Pi_e|\Gamma\rangle = 0$, we get

$$\Pi_e|\Gamma\rangle = \lambda_1|\beta\rangle_k \otimes \Pi_e\big(|\alpha\rangle_i \otimes |\text{rest}\rangle\big) + \lambda_2|\beta^\perp\rangle_k \otimes \Pi_e\big(|\alpha^\perp\rangle_i \otimes |\text{rest}\rangle\big) = 0 .$$

As $|\beta\rangle_k$ and $|\beta^\perp\rangle_k$ are linearly independent, we can conclude that

$$\Pi_e\big(|\alpha\rangle_i \otimes |\text{rest}\rangle\big) = \Pi_e\big(|\alpha^\perp\rangle_i \otimes |\text{rest}\rangle\big) = 0 .$$

To prove the first claim assume, by way of contradiction, that $|\psi\rangle$ is entangled. Then by Lemma 3.2, Π_e propagates $|\alpha\rangle$ and $|\alpha^\perp\rangle$ to two *different* states, say, $|\gamma_1\rangle_j \neq |\gamma_2\rangle_j$. However by Lemma 3.3, it follows that $|\text{rest}\rangle$ must be both in the form $|\gamma_1\rangle_j \otimes |\text{rest}'\rangle$ and $|\gamma_2\rangle_j \otimes |\text{rest}'\rangle$—which leads to a contradiction.

For the second claim, assume that $|\psi\rangle = |x\rangle_i \otimes |y\rangle_j$ is a product state. We find that

$$\Pi_e\big(|\alpha\rangle_i \otimes |\text{rest}\rangle\big) = \Pi_e\big(|\alpha^\perp\rangle_i \otimes |\text{rest}\rangle\big) = 0$$

because both states, $|\alpha\rangle_i \otimes |\text{rest}\rangle$ and $|\alpha^\perp\rangle_i \otimes |\text{rest}\rangle$, are ground states of the single projector Hamiltonian $\tilde{H} = \Pi_e$. Using Lemma 3.2 and Lemma 3.3, together with the fact that at least one of the states $|\alpha\rangle_i, |\alpha^\perp\rangle_i$ is different from $|x^\perp\rangle_i$, we conclude that $|\text{rest}\rangle = |y^\perp\rangle_j \otimes |\text{rest}'\rangle$. □

Let H be a 2-local Hamiltonian in rank-1 decomposition, let s be an assignment, and let $G_s = (V_s, E_s)$ be the constraint graph of the reduced Hamiltonian H_s. We would like to describe in G_s the result of the *iterated propagation* process where a value given to variable i is propagated along all possible projectors, followed by the propagated values being propagated on their turn. This is repeated until no value assigned during this process can be propagated further. The propagation can start either when the initial value is already assigned by s, that is when $s(i) = |\delta\rangle$ for $|\delta\rangle \in \{|\alpha\rangle, |\gamma\rangle_{ij}\}$, where $|\alpha\rangle$ is some 1-qubit state and $|\gamma\rangle_{ij}$ some 2-qubit state, or when $s(i) = \ominus$, in which case we shall choose an arbitrary 1-qubit state $|\alpha\rangle$ and assign it to i.

Let s, i and $|\delta\rangle$ be such that $s(i) \in \{\ominus, |\delta\rangle\}$. We say that in the constraint graph G_s an edge $e \in E_s$ from i to j *propagates* $|\delta\rangle$ if Π_e propagates it, and we denote by $\mathsf{prop}(s,e,|\delta\rangle)$ the state $|\delta\rangle$ is propagated to. Now we can generalize the notion of propagation in G_s from edges to paths. Let $i = i_0, i_1, \ldots, i_k$ be vertices in V_s, and let e_j be an edge from i_j to i_{j+1}, for $j = 0, \ldots, k-1$. Let $s(i) \in \{\ominus, |\delta\rangle\}$, and set $|\alpha_0\rangle = |\delta\rangle$. Let $|\alpha_1\rangle, \ldots, |\alpha_k\rangle$ be states such that the propagation of $|\alpha_j\rangle$ along Π_{e_j} is $|\alpha_{j+1}\rangle$, for $j = 0, \ldots, k-1$. Then we say that the path $p = (e_0, \ldots, e_{k-1})$ from i_0 to i_k *propagates* $|\delta\rangle$, and we set $\mathsf{prop}(s, p, |\delta\rangle) = |\alpha_k\rangle$. We say that a vertex $j \in V_s$ is *accessible* by propagating $|\delta\rangle$ from i if either $j = i$ or there is a path from i to j that propagates $|\delta\rangle$. We denote by $V_s^{\mathsf{prop}}(i, |\delta\rangle)$ the set of such vertices, and by $\mathsf{ext}_s^{\mathsf{prop}}(i, |\delta\rangle)$ the extension of s by the values given to the vertices in $V_s^{\mathsf{prop}}(i, |\delta\rangle)$ by iterated propagation.

The set $V_s^{\mathsf{prop}}(i, |\delta\rangle)$ divides the edges E_s into three disjoint subsets: the edges E_1 of the induced subgraph $G(V_s^{\mathsf{prop}}(i, |\delta\rangle))$, the edges E_2 between the induced subgraphs $G(V_s^{\mathsf{prop}}(i, |\delta\rangle))$ and $G(V_s \setminus V_s^{\mathsf{prop}}(i, |\delta\rangle))$, and the edges E_3 of the induced subgraph $G(V_s \setminus V_s^{\mathsf{prop}}(i, |\delta\rangle))$. While the edges in $E_1 \cup E_2$ are satisfied by $s' = \mathsf{ext}_s^{\mathsf{prop}}(i, |\delta\rangle)$, none of the edges in E_3 is satisfied by it. Therefore $G_{s'}$ is nothing but $G(V_s \setminus V_s^{\mathsf{prop}}(i, |\delta\rangle))$ and it can be constructed by the following process. Given s and i, the edges in $E_1 \cup E_2$ can be traversed via a breadth first search rooted at i. The levels of the tree are decided dynamically: at any level the next level is composed of those vertices whose value is propagated from the current level. A vertex of $G_{s'}$ is of degree 0 if all its adjacent edges are in E_2, these vertices will be removed from $V_{s'}$. The algorithm Propagation uses a temporary queue Q to implement this process.

Procedure 1 $\mathsf{Propagation}(s, G_s, i, |\delta\rangle)$

```
s(i) := |δ⟩
create a queue Q and put i into Q
while Q is not empty do
    remove the head j of Q
    for all edges e from j to k  do
        if e propagates s(j) then
            if s(k) ∉ {⊖, prop(s,e,s(j))} then abort and return "unsuccessful"
            if s(k) = ⊖ then s(k) := prop(s,e,s(j))
            enqueue k
        remove e and e^rev from E_s
        if  the list pointed to by k is empty  then remove k from V_s
    remove j from V_s
```

Lemma 3.5. (**Propagation Lemma**) *Let Propagation*$(s, G_s, i, |\delta\rangle)$ *be called when* H_s *does not have rank-*3 *constraints, and* $s(i) \in \{\ominus, |\delta\rangle\}$. *Let* s' *and* $G' = (V', E')$ *be the outcome of the procedure. The following hold true:*

1. *If Propagation$(s, G_s, i, |\delta\rangle)$ does not return "unsuccessful" then $s' = \mathsf{ext}_s^{\mathsf{prop}}(i, |\delta\rangle)$ and $G' = G_{s'}$. Moreover, if s is a pre-solution then s' is a pre-solution, and if s is closed then s' is also closed.*

2. *If Propagation(s, G_s, i) returns "unsuccessful" then there is no solution z that is an extension of s and for which $z(i) = |\delta\rangle$.*

3. *The complexity of the procedure is $O(|E_s| - |E_{s'}|)$.*

Proof. The assignments made during the breadth first search correspond exactly to the the paths propagating $|\delta\rangle$ from i, therefore the extension of s created by the process is indeed $s' = \mathsf{ext}_s^{\mathsf{prop}}(i, |\delta\rangle)$. The while loop removes the edges between vertices in $V_s^{\mathsf{prop}}(i, |\delta\rangle)$ and the edges between vertices in $V_s^{\mathsf{prop}}(i, |\delta\rangle)$ and in $V_s \setminus V_s^{\mathsf{prop}}(i, |\delta\rangle)$. It also removes the vertices in $V_s^{\mathsf{prop}}(i, |\delta\rangle)$ and the vertices in $V_s \setminus V_s^{\mathsf{prop}}(i, |\delta\rangle)$ whose degree became 0. Therefore in $H_{s'}$ for every qubit there is a local projector that acts on this qubit, and we have $G' = G_{s'}$.

Let us suppose that s is a pre-solution, and let z be an extension of s where z is a solution and it is a product state on the vertices in V_s. By Theorem 2.4 there exists such a solution because H_s does not have rank-3 constraints. We define the assignment z' by

$$z'(j) = \begin{cases} s'(j) & \text{if } \ j \in \operatorname{supp}(s'), \\ z(j) & \text{otherwise.} \end{cases}$$

Then z' is a solution that is an extension of s', and therefore s' is a pre-solution. If s is closed then so is s' as it is only the vertices in $V_s^{\mathsf{prop}}(i, |\delta\rangle)$ that get assigned during the process, and they are not included into $V_{s'}$.

Let us now suppose that the procedure returns "unsuccessful." Then, there is a vertex $k \in V_s^{\mathsf{prop}}(i, |\delta\rangle)$, and two paths p and p' in G_s from i to k such that $\mathsf{prop}(s, p, |\delta\rangle) = |\beta\rangle$, $\mathsf{prop}(s, p', |\delta\rangle) = |\beta'\rangle$ and $|\beta\rangle \neq |\beta'\rangle$. Let us also suppose that there exists a solution z that is an extension of s and for which $z(i) = |\delta\rangle$. Then, by the repeated use of Lemma 3.3 along with a single use of Lemma 3.4 when $|\delta\rangle$ is a 2-qubit entangled state, we conclude that $z(k)$ is simultaneously equal to $|\beta\rangle$ and to $|\beta'\rangle$—which is a contradiction.

Finally Statement 3 follows because every step of the procedure can be naturally charged to an edge in $E_s \setminus E_{s'}$, and every edge is charged only a constant number of times. □

4 The main algorithm

4.1 Description of the algorithm

Now we give a high level description of our algorithm, Q2SATSolver. It takes as input the adjacency list representation of the constraint graph $G(H)$ of a 2-local Hamiltonian H in rank-1 decomposition. The algorithm uses four global variables: assignments s_0 and s_1 initialized to $s_\ominus$, and graphs G_0 and G_1 in the adjacency list representation, initialized to $G(H)$. The algorithm consists of four phases, and except for the first one, each phase consists of several stages, where essentially one stage corresponds to one Propagation process. In the case of an unsatisfiable Hamiltonian the algorithm at some point outputs "H is unsatisfiable" and stops. This happens when either the maximal rank constraints are already unsatisfiable, or at some later point several values are assigned to the same variable during a propagation process that should necessarily succeed to obtain a satisfying assignment.

In the case of a frustration-free Hamiltonian, at the beginning and at the end of each stage, we will have $s_0 = s_1$ and $G_0 = G_1 = G_{s_0}$. In the first two phases only (s_0, G_0) develops, and is copied to (s_1, G_1) at the end of the phase. In the last two phases, (s_0, G_0) and (s_1, G_1) develop independently, but only the result of one of the two processes is retained and is copied into the other variable at the end of the phase. This parallel development of the two processes is necessary for complexity considerations, as it ensures that some potentially useless work done during the stage is proportional to the useful work of the stage.

In the first phase, the procedure MaxRankRemoval satisfies, if at all possible, every constraint of maximal rank. In the second phase, all these assignments are propagated, which, if the phase is successful, results in a closed assignment s such that H_s has only rank-1 constraints. In the third phase the procedure ParallelPropagation satisfies the product constraints one by one and propagates the assigned values. To satisfy a product constraint, the only two possible choices are tried and propagated in parallel. In the fourth phase, the remaining entangled constraints are taken care of, again, one by one. To satisfy an entangled constraint an arbitrary value, which we choose to be $|0\rangle$, is tried and propagated. In the case of an unsuccessful propagation, we are able to efficiently find a product constraint implied by the entangled constraints considered during the propagation, and therefore it becomes possible to proceed as in phase three. In the case of a successful propagation, we are left with a satisfying assignment and the empty constraint graph. Theorem 1.1 is an immediate consequence of the following result.

Algorithm 2 Q2SATSolver$(G(H))$

$s_0 = s_1 := s_\ominus$, $G_0 = G_1 := G(H)$ ▷ Initialize global variables

MaxRankRemoval() ▷ Phase 1: Remove maximal rank constraints

while there exist $i \in V_0$ such that $s(i) \neq \ominus$ **do** ▷ Phase 2: Propagate all assigned values
  PROPAGATE$(s_0, G_0, i, s_0(i))$
  if the propagation returns "unsuccessful" **output** "H is unsatisfiable"
  $s_1 := s_0, G_1 := G_0$

while there exists a product constraint $\Pi_{i_0 i_1} = |\alpha_0^\perp\rangle\langle\alpha_0^\perp|_{i_0} \otimes |\alpha_1^\perp\rangle\langle\alpha_1^\perp|_{i_1}$ in G_0 **do**
  ParallelPropagation$(i_0, |\alpha_0\rangle, i_1, |\alpha_1\rangle)$ ▷ Phase 3: Remove product constraints

while G_0 is not empty **do** ▷ Phase 4: Remove entangled constraints
  ProbePropagation(i) for some vertex i

output $|s\rangle$ for any total extension s of s_0.

Theorem 4.1. *Let $G(H) = (V, E)$ be the constraint graph of a 2-local Hamiltonian. Then, the following facts hold:*

1. *If H is frustration-free, the algorithm Q2SATSolver$(G(H))$ outputs a ground state $|s\rangle$.*

2. *If H is not frustration-free, the algorithm Q2SATSolver$(G(H))$ outputs "H is unsatisfiable."*

3. *The running time of the algorithm is* $O(|V|+|E|)$.

Theorem 4.1 will be proven in Section 4.5.

4.2 Max rank removal

The MaxRankRemoval procedure is conceptually very simple. As every maximal rank constraint has a unique solution (up to a global phase), it uses this assignment for each constraint, and then checks if this is globally consistent.

Procedure 3 MaxRankRemoval()

for all $i \in V_0$ such that $\mathtt{rank}(\Pi_{ii}) = 1$ **let** $|\phi\rangle$ be a state satisfying Π_{ii} **do**
  $s_0(i) := |\phi\rangle$
for all $i \in V_0$, for all edge $e \in E_0$ from i to j such that $\mathtt{rank}(\Pi_e) = 3$ **do**
  let $|\gamma\rangle$ be the unique state that satisfies Π_e

  if $|\gamma\rangle = |\alpha\rangle_i |\beta\rangle_j$ is a product state **then**
    if $s_0(i) \notin \{\ominus, |\alpha\rangle\}$ **then output** "H is unsatisfiable"
    if $s_0(i) = \ominus$ **then** $s_0(i) := |\alpha\rangle$

  if $|\gamma\rangle$ is an entangled state **then**
    if $s_0(i) \notin \{\ominus, |\gamma\rangle_{ij}\}$ **then output** "H is unsatisfiable"
    if $s_0(i) = \ominus$ **then** $s_0(i) := |\gamma\rangle_{ij}$

remove from E_0 every edge e such that Π_e is satisfied by s_0.
remove every isolated vertex from G_0
$s_1 := s_0$, $G_1 := G_0$

Lemma 4.2. *Let* s_0, G_0, s_1, G_1 *be the outcome of* MaxRankRemoval. *Then the following holds true:*

1. *If MaxRankRemoval does not output "H is unsatisfiable" then* s_0 *satisfies every maximal rank constraint,* $G_0 = G(H_{s_0})$, $s_0 = s_1$, *and* $G_0 = G_1$. *Moreover, if* H *is satisfiable then* s_0 *is a pre-solution.*

2. *If MaxRankRemoval outputs "H is unsatisfiable" then H is unsatisfiable.*

3. *The complexity of the procedure is* $O(|V|+|E|)$.

Proof. If the procedure does not output "H is unsatisfiable" then indeed s_0 satisfies all maximal rank constraints. The removal of the necessary edges and vertices ensures that $G_0 = G(H_{s_0})$, and obviously $s_0 = s_1, G_0 = G_1$. If H is satisfiable, then it has a ground state for some total assignment s. This s is an extension of s_0 because there is a unique way to satisfy the maximal rank constraints.

Maximal rank projectors are such that there is a unique assignment to their qubits that satisfies them. The first part of the procedure creates the assignment that assigns these necessary values. If several different values are assigned to some variable then H is unsatisfiable. Similarly, if s_0 assigns an entangled 2-qubit state between variables i and k, and there is an entangled rank-1 constraint between i and j, then by Lemma 3.4 it is impossible to extend s_0 into a satisfying assignment, and therefore H is unsatisfiable. This proves Statement 2.

The procedure can be executed by a constant number of vertex and edge traversals for s_0, and similarly for s_1. □

4.3 The ParallelPropagation procedure

The procedure ParallelPropagation is called when s_0 is a closed assignment, and G_{s_0} contains an edge with a product constraint. As there are only two ways to satisfy a product constraint, these are tried and propagated in parallel. If one of these propagations terminates successfully, the other is stopped, which ensures that the overall work done is proportional to the progress made.

Procedure 4 ParallelPropagation$(i_0, |\alpha_0\rangle, i_1, |\alpha_1\rangle)$

Run in parallel Propagation$(s_0, G_0, i_0, |\alpha_0\rangle)$ and Propagation$(s_1, G_1, i_1, |\alpha_1\rangle)$ step by step
until one of them terminates successfully **or** both terminate unsuccessfully

if both propagations terminate unsuccessfully **then**
  output "H is unsatisfiable"
else let Propagation$(s_0, G_0, i_0, |\alpha_0\rangle)$ terminate first (the other case is symmetric)
  undo Propagation$(s_1, G_1, i_1, |\alpha_1\rangle)$
  $s_1 := s_0,\ G_1 := G_0$

Lemma 4.3. *Let ParallelPropagation be called when s_0 is closed, H_{s_0} does not have rank-3 constraints, $G_0 = G_{s_0}$, there exists a product edge in G_0 from i_0 to i_1 with the constraint $|\alpha_0^\perp\rangle\langle\alpha_0^\perp|_{i_0} \otimes |\alpha_1^\perp\rangle\langle\alpha_1^\perp|_{i_1}$, $s_1 = s_0$ and $G_1 = G_0$. Let s'_0, s'_1, G'_0, G'_1 be the outcome of the procedure. Then the following holds:*

1. *If ParallelPropagation does not output "H is unsatisfiable" then s'_0 is a proper closed extension of s_0, $G'_0 = G_{s'_0}$, $s'_1 = s'_0$ and $G'_1 = G'_0$. Moreover, if s is a pre-solution then s'_0 is a pre-solution.*

2. *If ParallelPropagation outputs "H is unsatisfiable" then H is unsatisfiable.*

3. *The complexity of the procedure is $O(|E_{s_0}| - |E_{s'_0}|)$.*

Proof. If the procedure does not output "H is unsatisfiable" then at least one of the parallel propagations terminates successfully, say Propagation$(s_0, G_0, i_0, |\alpha_0\rangle)$. Then s'_0 is a proper extension of s_0 because s_0 is closed and therefore $s_0(i_0) = \ominus$. Obviously $s'_1 = s'_0$ and $G'_1 = G'_0$, and all other claims follow from the Propagation Lemma.

As H_{s_0} does not have rank-3 constraints, by Theorem 2.4 if it is frustration free, it has a product ground state. In H_{s_0} there exists a product edge from i_0 to i_1 with constraint $|\alpha_0^\perp\rangle\langle\alpha_0^\perp|_{i_0} \otimes |\alpha_1^\perp\rangle\langle\alpha_1^\perp|_{i_1}$, therefore only the assignments that have either $|\alpha_0\rangle$ assigned to variable i_0 or $|\alpha_1\rangle$ assigned to variable i_1

can be a solution. But if both propagations output "unsuccessful," then, by the Propagation Lemma no such assignment can satisfy H_{s_0}. Therefore H_{s_0} is not frustration free, and neither is H.

For the complexity analysis observe that the unsuccessful or unterminated propagation of the parallel processes makes at most as many steps as the successful one. This is the reason for performing the two propagations step by step in parallel. Undoing this propagation can be performed in the same order of time as the propagation itself, for example, by copying the removed edges into temporary lists. The claim on the complexity of the successful propagation follows from the Propagation Lemma. □

4.4 The ProbePropagation procedure

The procedure ProbePropagation is evoked when s_0 is a closed assignment, and G_{s_0} has only entangled constraints. The general outline of the procedure is it follows. It begins by picking an arbitrary vertex $i \in V_s$, assigning $|0\rangle$ (an arbitrary value) to it, and propagating this choice. In the lucky case of a successful propagation, this call to the procedure ends. Otherwise, we reach a contradiction: there are two paths p_1, p_2, starting from qubit i and ending at some qubit j, such that the propagation along p_1 assigns the state $|\beta_1\rangle$ to qubit j, whereas the propagation along p_2 assigns $|\beta_2\rangle$ to it—and these two states are not proportional to each other. These two paths form a cycle, which we call a *contradicting cycle*, as illustrated in Figure 1(a). Let us label the two paths by $p_1 := (i = i_0 \to i_1 \to \cdots \to i_k = j)$ and $p_2 := (i = i'_0 \to i'_1 \to \cdots \to i'_\ell = j)$. Below, we introduce an operation called "sliding," which enables us to take the rank-1 projector on i and i_1 and "slide" it along the path p_1 so that it acts on i and j. Similarly, we slide the projector of i and i'_1, along p_2, ending this way with two rank-1 projectors acting on qubits i and j, as shown in Figure 1(b). Assuming these two projectors are different (which, as we show, must be the case because of the contradiction we initially obtained), their ground space is two dimensional, and so is its complement. It follows from Proposition 2.3, that the complement subspace must contain a product state, hence the two projectors can be replaced by two other projectors, one of which is a product projector, *without changing their ground space*. We have therefore re-introduced a product constraint into the system, which can then be handled by the ParallelPropagation procedure.

We continue by describing in detail the sliding operation, which first appeared in a similar form in Ref. [15]. We formulate it as a lemma, and give its proof to ensure that it can be done efficiently in our algebraic computation model.

Lemma 4.4 (Sliding Lemma)**.** *Consider a system on* 3 *qubits* i, j *and* k*, together with two rank-*1 *projectors* $\Pi_1 := |\psi_1\rangle\langle\psi_1|_{ij}$ *on qubits* (i, j) *and* $\Pi_2 := |\psi_2\rangle\langle\psi_2|_{jk}$ *on qubits* (j, k)*. If* $|\psi_2\rangle$ *is entangled, then we can find another rank-*1 *projector* $\Pi_3 := |\psi_3\rangle\langle\psi_3|_{ik}$ *on qubits* (i, k) *such that the ground space of* $\Pi_1 + \Pi_2$ *is identical to the ground space of* $\Pi_2 + \Pi_3$*. In addition, if a single qubit state* $|\alpha\rangle_i$ *is propagated by* $\Pi_1 + \Pi_2$ *to* $|\beta\rangle_k$*, it is also propagated to* $|\beta\rangle_k$ *directly via* Π_3*.*

Proof. Expand $|\psi_2\rangle_{jk}$ in terms of the standard basis on qubit k as $|\psi_2\rangle_{jk} = |x\rangle_j|0\rangle_k + |y\rangle_j|1\rangle_k$. The states $|x\rangle_j, |y\rangle_j$ are not necessarily normalized or orthogonal to each other, but they must be linearly independent, otherwise $|\psi_2\rangle$ can be written as a product state. Consequently, we can use Gaussian elimination to find the transformation T on qubit j such that $T|x\rangle = |1\rangle$ and $T|y\rangle = -|0\rangle$. Note that T must be unique and non-singular, and that $T|\psi_2\rangle_{jk} = |1\rangle|0\rangle - |0\rangle|1\rangle$ is the anti-symmetric state. Let $|\tilde{\psi}_1\rangle_{ij} = T|\psi_1\rangle_{ij}$ and $|\tilde{\psi}_2\rangle_{jk} = T|\psi_2\rangle_{jk}$ respectively, and use them to define the rank-1 projectors $\tilde{\Pi}_1 = |\tilde{\psi}_{1ij}\rangle\langle\tilde{\psi}_{1ij}|, \tilde{\Pi}_2 = |\tilde{\psi}_2\rangle\langle\tilde{\psi}_2|_{jk}$. Any state in the ground space of $\tilde{\Pi}_1 + \tilde{\Pi}_2$ must be invariant under

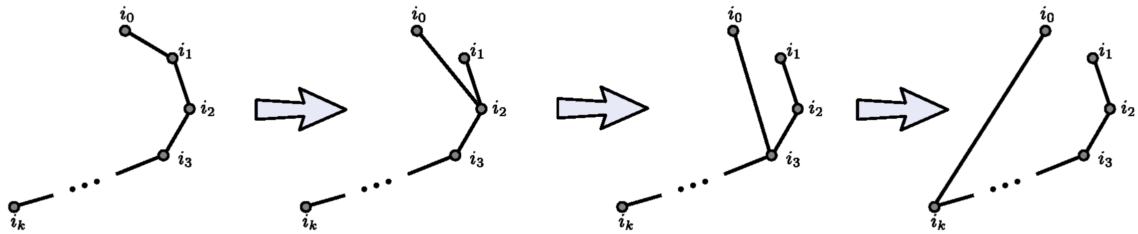


Figure 3: The sliding of the edge (i_0, i_1) over the path $i_1 \to i_2 \to \cdots \to i_k$, until it becomes the edge (i_0, i_k).

a swapping of qubits j,k because $\tilde{\Pi}_2$ projects into the anti-symmetric subspace. Therefore, defining $|\psi_3\rangle_{ik} = |\tilde{\psi}_1\rangle_{ik}$, and $\Pi_3 = |\psi_3\rangle\langle\psi_3|_{ik}$, the ground space of $\tilde{\Pi}_1 + \tilde{\Pi}_2$ is identical to the ground space of $\Pi_3 + \tilde{\Pi}_2$. Now, applying the inverse transformation T^{-1} on qubit j, the projector $\tilde{\Pi}_2$ returns to Π_2, while Π_3 remains unchanged. As both T and T^{-1} are non-singular, it follows that ground space of $\Pi_1 + \Pi_2$ is identical to the ground space to $\Pi_2 + \Pi_3$.

For the second claim, assume by way contradiction that Π_3 does not propagate $|\alpha\rangle_i$ to $|\beta\rangle_k$. Then there is a 1-qubit state $|\gamma\rangle \neq |\beta\rangle$, such that $\Pi_3(|\alpha\rangle_i|\gamma\rangle_k) = 0$. As Π_2 is an entangled rank-1 projector, it propagates $|\gamma\rangle_k$ to some state $|\delta\rangle_j$ (see Lemma 3.2). Therefore, the state $|\alpha\rangle_i|\delta\rangle_j|\gamma\rangle_k$ is a ground state of $\Pi_2 + \Pi_3$, as well as of $\Pi_1 + \Pi_2$. However, this contradicts the assumption that the latter propagates $|\alpha\rangle_i$ to $|\beta\rangle_k$. □

Using the sliding lemma iteratively on a path $p := i_0 \to i_1 \to \cdots \to i_k$ that is made of entangled rank-1 projectors $\Pi_{i_0,i_1}, \ldots \Pi_{i_{k-1},i_k}$, we can transform the first projector Π_{i_0,i_1} to a projector Π_{i_0,i_2}, and then to Π_{i_0,i_3}, and so on until we obtain Π_{i_0,i_k}, as illustrated in Figure 3. Let us denote the underlying 2-qubit state in Π_{i_0,i_k}, by $|\mathsf{slide}(p)\rangle$, i. e.,

$$\Pi_{i_0,i_k} := |\mathsf{slide}(p)\rangle\langle\mathsf{slide}(p)|\,.$$

Then we reach the following corollary.

Corollary 4.5. *Given a path $p := i_0 \to i_1 \to \cdots \to i_k$ of entangled rank-1 projectors $\Pi_{i_0,i_1}, \ldots, \Pi_{i_{k-1},i_k}$, apply the sliding lemma iteratively to obtain $|\mathsf{slide}(p)\rangle_{i_0,i_k}$. Then the ground space of*

$$\Pi_{i_0,i_1} + \Pi_{i_1,i_2} + \cdots + \Pi_{i_{k-1},i_k}$$

is equal to the ground space of

$$\Pi_{i_1,i_2} + \cdots + \Pi_{i_{k-1},i_k} + |\mathsf{slide}(p)\rangle\langle\mathsf{slide}(p)|\,.$$

Moreover, if $|\alpha\rangle_{i_0}$ is propagated to $|\beta\rangle_{i_k}$ along p, then it is also propagated directly by $|\mathsf{slide}(p)\rangle\langle\mathsf{slide}(p)|$.

We are now ready to state the ProbePropagation procedure and analyze it formally.

Procedure 5 $\mathsf{ProbePropagation}(i)$

> $\mathsf{Propagation}(s_0, G_0, i, |0\rangle)$.
> **if** the propagation is successful **then** $s_1 := s_0$, $G_1 := G_0$
> **else**
>   Let j such that $|s_0(j)| > 1$
>   **find** two paths p_1 and p_2 in G_0 from i to j such that $\mathsf{prop}(s_0, p_1, |0\rangle) \neq \mathsf{prop}(s_0, p_2, |0\rangle)$
>   **find** a product state $|\alpha^\perp\rangle \otimes |\beta^\perp\rangle$ in the two-dimensional subspace $\mathrm{span}\{|\mathsf{slide}(p_1)\rangle, |\mathsf{slide}(p_2)\rangle\}$
>   **undo** $\mathsf{Propagation}(s_0, G_0, i, |0\rangle)$
> $\mathsf{ParallelPropagation}(j, |\alpha\rangle, j, |\beta\rangle)$

Lemma 4.6. *Let ProbePropagation be called when s_0 is closed, H_{s_0} has only rank-1 entangled constraints, $G_0 = G_{s_0}, s_1 = s_0$ and $G_1 = G_0$. Let s_0', s_1', G_0', G_1' be the outcome of the procedure. Then the following holds:*

1. *If ProbePropagation does not output "H is unsatisfiable" then s_0' is a proper closed extension of s_0, $G_0' = G_{s_0'}, s_1' = s_0'$ and $G_1' = G_0'$. Moreover, if s is a pre-solution then s_0' is a pre-solution.*

2. *If the call to ParallelPropagation outputs "H is unsatisfiable" then H is unsatisfiable.*

3. *The complexity of the procedure is $O(|E_{s_0}| - |E_{s_0'}|)$.*

Proof. If the procedure does not output "H is unsatisfiable" then either $\mathsf{Propagation}(s_0, G_0, i, |0\rangle)$ or one of the parallel propagations (say $\mathsf{Propagation}(s_0, G_0, i, |\alpha\rangle)$) terminates successfully. Thus s_0' is a proper extension of s_0 because s_0 is closed and therefore $s_0(i) = \ominus$. Obviously $s_1' = s_0'$ and $G_1' = G_0'$, and all other claims follow from the Propagation Lemma.

Let us suppose that all three propagations are unsuccessful. By Corollary 4.5, any solution for H_{s_0} also satisfies the system obtained after sliding the constraints from i along paths p_1 and p_2, with the new constraints $|\mathsf{slide}(p_1)\rangle\langle\mathsf{slide}(p_1)|_{ij}$ and $|\mathsf{slide}(p_2)\rangle\langle\mathsf{slide}(p_2)|_{ij}$. Using Proposition 2.3, as $|\alpha^\perp\rangle_i \otimes |\beta^\perp\rangle_j$ lies in $\mathrm{span}\{|\mathsf{slide}(p_1)\rangle, |\mathsf{slide}(p_2)\rangle\}$, any state orthogonal to the latter subspace will also be orthogonal to the former. Hence, any solution for H_{s_0} should also satisfy the product constraint $|\alpha^\perp\rangle\langle\alpha^\perp|_i \otimes |\beta^\perp\rangle\langle\beta^\perp|_j$. Then, Lemma 4.3 implies that H_{s_0}, and by extension H is unsatisfiable if the call to $\mathsf{ParallelPropagation}$, made to satisfy $|\alpha^\perp\rangle\langle\alpha^\perp|_i \otimes |\beta^\perp\rangle\langle\beta^\perp|_j$, fails.

For the complexity analysis the interesting case is when the first propagation, that we call $\mathtt{Propagation}_{\text{failure}}$, is unsuccessful but one of the two parallel propagations is successful. Let us call this successful one $\mathtt{Propagation}_{\text{success}}$. The main observation here is that every propagating edge in $\mathtt{Propagation}_{\text{failure}}$ will also be propagating in $\mathtt{Propagation}_{\text{success}}$, because by Lemma 3.2 entangled edges always propagate. The paths p_1 and p_2 can be found in time proportional to the size of the subgraph visited by $\mathtt{Propagation}_{\text{failure}}$. Indeed, observe that the edges of the two paths, except the last edge of one of the two, are edges in the tree created by the breadth first search underlying $\mathtt{Propagation}_{\text{failure}}$. The way from a vertex to the root of the tree can be found by maintaining, for each vertex in the tree, a pointer towards its parent. The product state $|\alpha\rangle \otimes |\beta\rangle$ can be found in constant time by Proposition 2.3. Therefore, by applying Lemma 4.3, the complexity is indeed $O(|E_{s_0}| - |E_{s_0'}|)$. □

4.5 Analysis of the algorithm

Proof of Theorem 4.1. If H is frustration free then by Lemma 4.2 MaxRankRemoval outputs a pre-solution s_0 that satisfies every maximal rank constraint. By the Propagation Lemma, at the end of Phase 2, s_0 is additionally a closed solution. By Lemma 4.3 ParallelPropagation outputs s_0 such that H_s contains only entangled constraints. By Lemma 4.6 at the end of the algorithm H_s is now empty, and therefore s is a solution.

If the algorithm does not output "H is unsatisfiable" then by Lemma 4.2, by the Propagation Lemma, and by Lemma 4.3 and Lemma 4.6 it outputs an assignment s such that G_s is the empty graph, and therefore s is a solution.

The complexity of MaxRankRemoval by Lemma 4.2 is $O(|E|)$. After the second phase, the propagation of the assigned values during MaxRankRemoval, the copying of s_0 and G_0 into respectively s_1 and G_1 can be done by executing the same propagation steps this time with s_1 and G_1. The complexity of the rest of the algorithm by the Propagation Lemma, Lemma 4.3 and Lemma 4.6 is a telescopic sum. Observe that the constants hidden in the big-O notation in the terms of this sum are all the same, they are equal to the absolute constant in the complexity analysis of the algorithm Propagation referenced in the Propagation Lemma. Therefore the telescopic sum evaluates to $O(|E|)$, and the overall complexity of the algorithm is $O(|E|)$. $\square$

5 Bit complexity of Q2SATSolver

As mentioned in the introduction, the algorithm Q2SATSolver has been analyzed in the algebraic model of computation, where arithmetic operations on complex numbers are assumed to consume unit time. This was done mainly in order to simplify the presentation. However, in order to assess the running time of the algorithm in any realistic scenario, we must also take into account the actual cost of the arithmetic operations. This is not a completely trivial task: on the one hand, the continuous nature of a Q2SAT Hamiltonian implies that it should be represented using complex numbers with an exponentially high accuracy. But on the other hand, we also want the representation to be as efficient as possible, in order to minimize the total cost of each arithmetic operation. One way to approach this problem is to consider the input in the framework of *bounded algebraic numbers* and analyze the complexity of the algorithm in this setting. Essentially, this would require calculating the bit-wise cost of representing the input in this framework and performing the arithmetic operations of the algorithm on it. In other words, we need to calculate the *bit complexity* of the algorithm. The techniques used in this section are based on standard methods in algebraic computational complexity, and further details can be found in [6].

Representing the input and the output

As we would like to work in the framework of algebraic numbers, it helps to first list the kinds of number fields that will be used to represent the input and output. The simplest field to consider is that of rational numbers, $\mathbb{Q}$. To bound the size of the entries from $\mathbb{Q}$, a $k_{\mathbb{Q}}$*-number*, for some integer $k > 0$, is defined below.

Definition 5.1 ($k_{\mathbb{Q}}$-number)**.** A rational number u is a $k_{\mathbb{Q}}$-number if there exists integers u_1 and u_2 of at most k-bits such that $u = u_1/u_2$.

A $k_{\mathbb{Q}}$-number u_1/u_2 will be represented as the tuple (u_1, u_2) requiring $2k$ bits. As quantum projectors and states use complex entries, we also require numbers from $\mathbb{Q}(\mathrm{i})$, the extension field of $\mathbb{Q}$ with $\mathrm{i} = \sqrt{-1}$. This leads to the following definition of a $k_{\mathbb{Q}(\mathrm{i})}$*-number*:

Definition 5.2 ($k_{\mathbb{Q}(\mathrm{i})}$-number)**.** A complex number $a + b\mathrm{i} \in \mathbb{Q}(\mathrm{i})$ is a $k_{\mathbb{Q}(\mathrm{i})}$-number if its coefficients a and b are $k_{\mathbb{Q}}$-numbers.

A $k_{\mathbb{Q}(\mathrm{i})}$-number $a+b\mathrm{i}$ will be represented as the tuple (a,b) for $k_{\mathbb{Q}}$-numbers a and b, thereby requiring $4k$ bits in total. When the square root, $\sqrt{t}$, of a square-free integer t (that is, an integer which is divisible by no perfect square other than 1) is generated during the course of the algorithm, we shall consider the field extension $\mathbb{Q}(\mathrm{i}, \sqrt{t})$ of $\mathbb{Q}(\mathrm{i})$ and the corresponding $k_{\mathbb{Q}(\mathrm{i},\sqrt{t})}$*-numbers*, defined below.

Definition 5.3 ($k_{\mathbb{Q}(\mathrm{i},\sqrt{t})}$-number)**.** The number $a_1 + a_2\sqrt{t} + a_3\mathrm{i} + a_4\sqrt{t}\mathrm{i} \in \mathbb{Q}(\mathrm{i}, \sqrt{t})$ is a $k_{\mathbb{Q}(\mathrm{i},\sqrt{t})}$-number if its coefficients, a_1, a_2, a_3 and a_4, are $k_{\mathbb{Q}}$-numbers.

Clearly, when the integer t can be represented as a k'-bit integer, a $k_{\mathbb{Q}(\mathrm{i},\sqrt{t})}$-number can be represented using $8k + k'$ bits by the tuple (a_1, a_2, a_3, a_4, t) for $k_{\mathbb{Q}}$-numbers a_1, a_2, a_3, a_4.

The input Hamiltonian is a set of m different 2-qubit projectors whose non-trivial parts are 4×4 complex matrices. As mentioned in Section 2.2, for a 2-qubit projector $\Pi = \hat{\Pi} \otimes \mathbb{I}_{\text{rest}}$, the input specifies the non-trivial part $\hat{\Pi}$ which is uniquely determined by its image subspace, $\mathsf{img}(\hat{\Pi})$. We consider each projector is given as a $k_{\mathbb{Q}(\mathrm{i})}$*-projector* which is defined below for some constant $k > 0$.

Definition 5.4 ($k_{\mathbb{Q}(\mathrm{i})}$-projector)**.** Consider a 2-qubit projector $\hat{\Pi}$ of rank-r to be specified by r independent, but not necessarily orthogonal, 4×1 vectors $\{v_1, \dots, v_r\}$ such that $\mathsf{img}(\hat{\Pi}) = \mathsf{span}\{v_1, \dots, v_r\}$. $\hat{\Pi}$ is a $k_{\mathbb{Q}(\mathrm{i})}$-projector if $v_1, v_2, \dots, v_r$ are given in the standard basis with $k_{\mathbb{Q}(\mathrm{i})}$-number coefficients.

Then, the Hamiltonian can be represented by m different $k_{\mathbb{Q}(\mathrm{i})}$-projectors, leading to a total space consumption of at most $m \times 4 \times 4 \times 1 \times 4k = 64mk$ bits. The ground state output will be a tensor product of single qubit and 2-qubit states which are length 2 and 4 complex vectors, respectively. Each entry of these vectors could belong to $\mathbb{Q}, \mathbb{Q}(\mathrm{i})$ or $\mathbb{Q}(\mathrm{i}, \sqrt{t})$, for different square free integers t. We will show below that there can be at most n different integers t, and that the state assigned to each variable consumes $O(n)$ bits. Recall from Section 2.4 that the algorithm proceeds by assigning un-normalized states without affecting its accuracy. We also suppose that the basis vectors describing the image subspace of an input projector need not be normalized.

Cost of arithmetic operations

A useful fact on the bit complexity of basic arithmetic operations that will be used repeatedly is stated below. Let $M(k, k')$ be the time required to multiply a k-bit integer with a k'-bit integer and let $M(k) = M(k,k)$. The currently known most efficient algorithm for integer multiplication uses Fourier transforms bounding $M(k) = k \log k 8^{\Theta(\log^* k)}$ [13] where

$$\log^* k = \min\{w \in \mathbb{N} : \log\log \overset{w\text{ times}}{\cdots\cdots} \log k \leq 1\}.$$

Fact 5.5. *Let a be a $k_{\mathbb{Q}}$-number and b a $k'_{\mathbb{Q}}$-number. Adding, subtracting, multiplying or dividing a and b can be performed in time $O(M(k,k'))$, and the result is an $O(k+k')_{\mathbb{Q}}$-number.*

Now we consider the bit complexity incurred during the course of the Q2SATSolver algorithm.

Theorem 5.6. *Let H be a* Q2SAT *Hamiltonian on n qubits consisting of m projectors with complex entries, each given as two $k_{\mathbb{Q}(\mathrm{i})}$-numbers, for some constant $k > 0$. Then the bit complexity of Q2SATSolver(H) is $O((n+m)\,M(n))$.*

Proof. The straightforward approach is to calculate the bit complexity of each operation that manipulates the projectors and assignments. In the first phase of MaxRankRemoval, the unique state satisfying each 2-qubit projector of rank-3 can be found using Gaussian elimination which for an $O(1)$-sized matrix is equivalent to a constant number of multiplications. Using Fact 5.5, the 2-qubit state found will hence be an $O(k)_{\mathbb{Q}(\mathrm{i})}$-number.

The second and third phases of the algorithm repeatedly propagate values across rank-1 constraints. The bit complexity of propagation is now analyzed. Given a product constraint $|\alpha\rangle\langle\alpha| \otimes |\beta\rangle\langle\beta|$, it requires finding $|\alpha^{\perp}\rangle$ and $|\beta^{\perp}\rangle$ using complex conjugates. Given an entangled constraint and a state assigned to one end of the constraint, the propagated state can be found by solving a system of linear equations. Assuming the initial state and the projector use $k_{\mathbb{Q}(\mathrm{i})}$-numbers, the propagated state will be represented using $2k_{\mathbb{Q}(\mathrm{i})}$-numbers. To propagate ℓ steps, each step using $k_{\mathbb{Q}(\mathrm{i})}$-number projectors, starting with a $k'_{\mathbb{Q}(\mathrm{i})}$-number state, the final propagated state will use $(k'+\ell k)_{\mathbb{Q}(\mathrm{i})}$-numbers. Hence, an $O(n)$ step propagation will result in a final state represented with $O(n)_{\mathbb{Q}(\mathrm{i})}$-numbers when $k = O(1)$ and $k' = O(n)$. This is linear in the number of qubits and takes at most $O(n)M(n,k)$ time. This covers the second and third phases of the algorithm.

The last phase with ProbePropagation will deal with one or more disconnected components, each made of only entangled constraints. The complexity of this phase is of interest when a contradiction arises during propagation in a component and a satisfying assignment has to be found by sliding constraints and using Proposition 2.3. Consider the way sliding across a constraint is done as described in Lemma 4.4. Finding the transformation and its inverse both require a constant number of multiplications and the new constraint after sliding will be given using $(ck)_{\mathbb{Q}(\mathrm{i})}$-numbers for some constant c. Sliding along $O(n)$ constraints will finally result in constraints represented by $O(n)_{\mathbb{Q}(\mathrm{i})}$-numbers and consumes $O(n)M(n,k)$ time.

Computing the product constraint in the resulting 2-dimensional subspace requires computing an eigenvector as per Proposition 2.3 leading to the possibility of an irrational square root $\sqrt{r}$ being introduced into the representation, for some square-free integer r. This highlights the necessity of moving to the field extension $\mathbb{Q}(\mathrm{i},\sqrt{r})$. Assuming the two parallel rank-1 projectors were initially given using $k'_{\mathbb{Q}(\mathrm{i})}$-numbers, the product constraint will require $O(k')_{\mathbb{Q}(\mathrm{i},\sqrt{r})}$-numbers to describe it. Also, $\sqrt{r}$ is generated as the irrational part of the eigenvalue of a 2×2 matrix whose entries are $O(k')_{\mathbb{Q}(\mathrm{i})}$-numbers due to which t will be an $O(k')$-bit integer. Hence, the representation of each $O(k')_{\mathbb{Q}(\mathrm{i},\sqrt{t})}$-number will require $O(k')$ bits. Any propagation after this point does not introduce new irrational square roots. The component, if satisfiable after ℓ propagation steps, will have assignments given by $O(k'+\ell k)_{\mathbb{Q}(\mathrm{i},\sqrt{t})}$-numbers. An $O(n)$ step propagation in this setting will result in this component using $O(n)_{\mathbb{Q}(\mathrm{i},\sqrt{t})}$-numbers for the assignment with the operations bounded by $O(nM(n,n)) = O(nM(n))$ time when $k' = O(nk) = O(n)$.

Note that each disconnected component at the beginning of this phase may introduce different irrational square roots into their partial assignments.

Recall that since propagation is carried forward with unnormalised states, this is the only phase that may introduce square roots in the final assignment. The final output keeps all of these separate representations and does not homogenize them into a single representation. To conclude, each propagating, sliding or eigenvector finding step adds an overhead of at most $M(n,k) < M(n)$ where bit complexity is concerned. Using Theorem 4.1, the bit complexity of Q2SATSolver is therefore $O((n+m)M(n))$. □

Remark 5.7. The above explanation started by assuming the base representation field as that of the Gaussian rationals, $\mathbb{Q}(\mathrm{i})$, and extends to the appropriate $\mathbb{Q}(\mathrm{i},\sqrt{r})$ field for some square-free integer r when required. It is also possible to consider any algebraic number field as the base field and extend it appropriately using techniques from computational algebraic number theory [6]. Of course, any overheads in performing arithmetic operations and calculating field extensions will have to be accounted for accordingly.

References

[1] ITAI ARAD, MIKLOS SANTHA, AARTHI SUNDARAM, AND SHENGYU ZHANG: Linear time algorithm for quantum 2SAT. In *Proc. 43rd Internat. Colloq. on Automata, Languages and Programming (ICALP'16)*, volume 55 of *LIPIcs*, pp. 15:1–15:14. Schloss Dagstuhl, 2016. [doi:10.4230/LIPIcs.ICALP.2016.15, arXiv:1508.06340] 1

[2] BENGT ASPVALL, MICHAEL F. PLASS, AND ROBERT ENDRE TARJAN: A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Inf. Process. Lett.*, 8(3):121–123, 1979. [doi:10.1016/0020-0190(79)90002-4] 2

[3] NIEL DE BEAUDRAP AND SEVAG GHARIBIAN: A linear time algorithm for quantum 2-SAT. In *Proc. 31st IEEE Conf. on Computational Complexity (CCC'16)*, volume 50 of *LIPIcs*, pp. 27:1–27:21. Schloss Dagstuhl, 2016. [doi:10.4230/LIPIcs.CCC.2016.27, arXiv:1508.07338] 5

[4] SERGEY BRAVYI: Efficient algorithm for a quantum analogue of 2-SAT. *Contemporary Mathematics*, 536:33–48, 2011. [doi:10.1090/conm/536, arXiv:quant-ph/0602108] 3, 5

[5] JIANXIN CHEN, XIE CHEN, RUANYAO DUAN, ZHENGFENG JI, AND BEI ZENG: No-go theorem for one-way quantum computing on naturally occurring two-level systems. *Phys. Rev. A*, 83(5):050301, 2011. [doi:10.1103/PhysRevA.83.050301, arXiv:1004.3787] 8

[6] HENRI COHEN: *A Course in Computational Algebraic Number Theory*. Volume 138 of *Graduate Texts in Mathematics*. Springer, 1993. [doi:10.1007/978-3-662-02945-9] 21, 24

[7] STEPHEN A. COOK: The complexity of theorem-proving procedures. In *Proc. 3rd STOC*, pp. 151–158. ACM Press, 1971. [doi:10.1145/800157.805047] 2

[8] MARTIN DAVIS, GEORGE LOGEMANN, AND DONALD W. LOVELAND: A machine program for theorem-proving. *Commun. ACM*, 5(7):394–397, 1962. [doi:10.1145/368273.368557] 2, 5

[9] MARTIN DAVIS AND HILARY PUTNAM: A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, 1960. [doi:10.1145/321033.321034] 2, 5

[10] JENS EISERT, MARCUS CRAMER, AND MARTIN B. PLENIO: Area laws for the entanglement entropy. *Rev. Mod. Phys.*, 82(1):277–306, 2010. [doi:10.1103/RevModPhys.82.277, arXiv:0808.3773] 3

[11] SHIMON EVEN, ALON ITAI, AND ADI SHAMIR: On the complexity of timetable and multicommodity flow problems. *SIAM J. Comput.*, 5(4):691–703, 1976. Preliminary version in FOCS'75. [doi:10.1137/0205048] 2

[12] DAVID GOSSET AND DANIEL NAGAJ: Quantum 3-SAT is QMA_1-complete. *SIAM J. Comput.*, 45(3):1080–1128, 2016. Preliminary version in FOCS'13. [doi:10.1137/140957056, arXiv:1302.0290] 3

[13] DAVID HARVEY, JORIS VAN DER HOEVEN, AND GRÉGOIRE LECERF: Even faster integer multiplication. *J. Complexity*, 36:1–30, 2016. [doi:10.1016/j.jco.2016.03.001, arXiv:1407.3360] 22

[14] MICHAŁ HORODECKI, PAWEŁ HORODECKI, AND RYSZARD HORODECKI: Separability of mixed states: necessary and sufficient conditions. *Phys. Lett. A*, 223(1–2):1–8, 1996. [doi:10.1016/S0375-9601(96)00706-2, arXiv:quant-ph/9605038] 6

[15] ZHENGFENG JI, ZHAOHUI WEI, AND BEI ZENG: Complete characterization of the ground-space structure of two-body frustration-free Hamiltonians for qubits. *Phys. Rev. A*, 84(4):042338, 2011. [doi:10.1103/PhysRevA.84.042338, arXiv:1010.2480] 4, 18

[16] RICHARD M. KARP: Reducibility among combinatorial problems. In *Complexity of Computer Computations*, The IBM Research Symposia Series, pp. 85–103. Plenum Press, New York, 1972. [doi:10.1007/978-1-4684-2001-2_9] 2

[17] ALEXEI YU. KITAEV: Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, 2003. [doi:10.1016/S0003-4916(02)00018-0, arXiv:quant-ph/9707021] 3

[18] ALEXEI YU. KITAEV, ALEXANDER SHEN, AND MIKHAIL N. VYALYI: *Classical and Quantum Computation*. Amer. Math. Soc., 2002. [doi:10.1090/gsm/047] 3

[19] MELVEN R. KROM: The decision problem for a class of first-order formulas in which all disjunctions are binary. *Mathematical Logic Quarterly*, 13(1–2):15–20, 1967. [doi:10.1002/malq.19670130104] 2

[20] CHRISTOPHER R. LAUMANN, RODERICH MOESSNER, ANTONELLO SCARDDICHIO, AND SHIVAJI L. SONDHI: Random quantum satisfiability. *Quantum Inf. Comput.*, 10(1):1–15, 2010. Link at ACM DL. [arXiv:0903.1904] 5, 11

[21] LEONID ANATOLEVICH LEVIN: Universal sequential search problems. *Problems of Information Transmission*, 9(3):265–266, 1973. Available at Math-Net.Ru. 2

[22] CHRISTOS H. PAPADIMITRIOU: On selecting a satisfying truth assignment (extended abstract). In *Proc. 32nd FOCS*, pp. 163–169. IEEE Comp. Soc. Press, 1991. [doi:10.1109/SFCS.1991.185365] 2

[23] ASHER PERES: Separability criterion for density matrices. *Phys. Rev. Lett.*, 77(8):1413–1415, 1996. [doi:10.1103/PhysRevLett.77.1413, arXiv:quant-ph/9604005] 6

[24] SUBIR SACHDEV: *Quantum phase transitions*. Wiley Online Library, 2007. [doi:10.1002/9780470022184.hmm108] 2

[25] GUIFRE VIDAL, JOSÉ IGNACIO LATORRE, ENRIQUE RICO, AND ALEXEI YU. KITAEV: Entanglement in quantum critical phenomena. *Phys. Rev. Lett.*, 90(22):227902, 2003. [doi:10.1103/PhysRevLett.90.227902, arXiv:quant-ph/0211074] 2

AUTHORS

Itai Arad
Assistant professor
Technion - Israel Institute of Technology
arad.itai@fastmail.com
https://phys.technion.ac.il/en/people?view=person&id=871

Miklos Santha
Directeur de Recherche
Université Paris Diderot
miklos.santha@gmail.com
https://www.irif.fr/~santha/

Aarthi Sundaram
Hartree postdoctoral fellow
Joint Centre for Quantum Information and Computer Science
University of Maryland
aarthims@gmail.com
http://quics.umd.edu/people/aarthi-sundaram

Shengyu Zhang
Assistant professor
The Chinese University of Hong Kong
syzhang@cse.cuhk.edu.hk
http://www.cse.cuhk.edu.hk/~syzhang

ABOUT THE AUTHORS

ITAI ARAD was born and raised in a Kibbutz in the south of Israel, where the howling old owl in the woods is hunting the horny back toad. He did his undergraduate studies in mathematics and physics at Tel Aviv University (B. Sc. 1996), and later his Ph. D. studies at the Weizmann Institute of Science under the supervision of Itamar Procaccia in the subject of turbulence (Ph. D. 2001). After graduation he decided to study astrophysics, by taking a postdoc position at the Institute of Astronomy at the University of Cambridge, UK. He finally found his favourite research area with the help of Dorit Aharonov, who introduced him to the subject of Quantum Computation. After a postdoc in Berkeley with Umesh Vaziarani and some happy years at the Centre of Quantum Technologies (CQT) in Singapore, he is now an Assistant Professor at the faculty of Physics at the Technion in Israel. His research interests lie mainly in the intersection of condensed matter physics and quantum information, including Hamiltonian Complexity, and questions about area laws and tensor networks.

MIKLOS SANTHA received his Diploma in mathematics in 1979 from Eötvös University in Budapest, and his Ph. D. in mathematics in 1983 from the Université Paris 7. His advisor was Jacques Stern. Since 1988 he has been a CNRS researcher, currently at the Université Paris Diderot, IRIF. He is also a principal investigator at CQT in Singapore.

AARTHI SUNDARAM hails from India, the land of diversity and contrasts. She received a B. E. in Computer Science and M. Sc. in Mathematics from the Birla Institute of Technology and Science at their Goa campus in 2012. An elective course offered by Dr. Radhika Vatsan, during her undergraduate studies, introduced her to the world of Quantum Computing and provided the impetus to pursue research in this subject. She completed her Ph. D. at the Centre of Quantum Technologies (CQT) in 2017 in Singapore under the guidance of Miklos Santha. She is currently a Hartree Postdoctoral Fellow at the Joint Center for Quantum Information and Computer Science at the University of Maryland. Her research interests span classical and quantum complexity theory, including quantum algorithms, cryptography, Hamiltonian Complexity, and ideas pertaining to a quantum polynomial hierarchy. Outside of work she is an avid Formula 1 and racing enthusiast, and loves to peer through her camera lens to capture landscapes, architecture and moments.

SHENGYU ZHANG received his B. S. in Mathematics at Fudan University in 1999, his M. S. in Computer Science at Tsinghua University under the supervision of Mingsheng Ying in 2002, and his Ph. D. in Computer Science at Princeton University under the supervision of Andrew Chi-Chih Yao in 2006. He then worked at NEC Laboratories America for a summer, and at the California Institute of Technology for two years as a postdoctoral researcher. In 2008, he joined The Chinese University of Hong Kong where he is now an associate professor. His main research interests are algorithms and complexity theories in various quantum and randomized models.