Linear Time Algorithm for Quantum 2SAT

Itai Arad[1], Miklos Santha[2], Aarthi Sundaram[3], and Shengyu Zhang[4]

1   Centre for Quantum Technologies, National University of Singapore, Singapore
    arad.itai@fastmail.com
2   Centre for Quantum Technologies, National University of Singapore,
    Singapore; and
    CNRS, IRIF, Université Paris Diderot 75205 Paris, France
    miklos.santha@gmail.com
3   Centre for Quantum Technologies, National University of Singapore, Singapore
    aarthims@gmail.com
4   Department of Computer Science and Engineering, The Chinese University of
    Hong Kong, Shatin, N.T., Hong Kong
    syzhang@cse.cuhk.edu.hk

Abstract

A canonical 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 Q_{ij} on a system of n qubits, and the task is to decide whether the Hamiltonian $H = \sum Q_{ij}$ has a 0-eigenvalue, or it is larger than $1/n^c$ for some $c = O(1)$. 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 the 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. While Bravyi has shown that the quantum 2-SAT problem has a classical polynomial-time algorithm, the running time of his algorithm is $O(n^4)$. In this paper we give a classical algorithm with linear running time in the number of local projectors, therefore achieving the best possible complexity.

1   Introduction

Various formulations of the satisfiability problem of Boolean formulae arguably constitute the center piece 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 prove that none exists when the formula is unsatisfiable. In the case of the k-SAT problem, where k is a positive integer, in each clause the number of literals is at most k. While k-SAT is an NP-complete problem [4, 12, 17] when $k \geq 3$, the 2-SAT problem is well-known to be efficiently solvable.

Polynomial time algorithms for 2-SAT come in various flavors. Let us suppose that the input formula has n variables and m clauses. The algorithm of Krom [15] 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 [9] has linear time complexity in m, as well as the elegant procedure of Aspvall, Plass and Tarjan [1] based on computing strongly connected components in a graph. A particularly simple randomized procedure of complexity $O(n^2)$ is described by Papadimitriou [18].

For our purposes the Davis-Putnam procedure [6] 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 [5], forms even today the basis for the most efficient SAT solvers. 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 contains only 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 satisfiabilty 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 contradiction we declare ϕ 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 by itself a Hermitian operator acting non-trivially on at most k qubits. Local Hamiltonians model the local interactions between quantum spins. Of central importance is the minimal eigenstate of the Hamiltonian, known as the *ground state*, and its associated eigenvalue, known as the *ground energy*. The ground state governs much of the low temperature physics of the system, such as quantum phase transitions and collective quantum phenomena [19, 20]. Finding the 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. This connection with complexity theory is of physical significance. Indeed, with the advent of quantum information theory and quantum complexity theories, it has become clear that the complexity of finding the ground state and its energy is intimately related to its entanglement structure. In recent years, much attention has been devoted into understanding this structure, revealing a rich an intricate behaviour such as area laws [8] and topological order [13].

The connection between classical k-SAT and quantum local Hamiltonian was formalized by Kitaev [14] who introduced the k-local Hamiltonian problem: one is given a k-local Hamiltonian H, along with two constants $a < b$ such that $b - a > 1/n^\alpha$ for some constant α. It is promised that the ground energy of H is at most a (the YES case) or is at least b (the NO case), and the task is to decide which case holds. Broadly speaking, given a quantum state $|\psi\rangle$, the energy of a local term $\langle\psi|h_i|\psi\rangle$ is 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. Therefore, in spirit, the k-local Hamiltonian problem corresponds to MAX-k-SAT, and indeed Kitaev has shown [14] that the 5-local Hamiltonian problem 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 larger than $b = 1/n^{\alpha}$ for some constant α. Notice that in the YES case, the energy of all projectors at the ground state is necessarily 0, since 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 the global ground state is also a ground state of every local term. Bravyi [2] has shown that quantum k-SAT was 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 Nagaj [10].

This paper is concerned with the quantum 2-SAT problem, which we will also denote simply by 2QSAT. One major result concerning this problem is due to Bravyi [2], who has proven that it belongs to the complexity class P. More precisely, he has proven that 2QSAT can be decided by a deterministic algorithm in time $O(n^4)$, together with a ground state that has a polynomial classical description. In the case of 2QSAT, 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 2QSAT of *linear* complexity.

▶ **Theorem 1.** *There is a deterministic algorithm for* 2QSAT *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 2SAT, but handles many difficulties arising in the quantum setting. First, a ground state of 2QSAT input may be entangled, a distinctive feature that classical 2SAT does not have. Thus the idea of setting some qubit to a certain state and propagating from there does not have a foundation in the first place. Indeed, if a rank-3 projector forces the only allowed state to be entangled, then any ground state is entangled in those two qubits. We overcome this by showing a *product-state theorem*, which asserts that for any frustration-free 2QSAT 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 this 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 suppose that the input contains only projectors of rank at most two. Such a system can be further simplified to a system consisting only of rank-1 projectors, by writing every rank-2 projector as a sum of two rank-1 projectors. Consider, for example, qubits 1 and 2 and a rank-1 projector $\Pi_{12} = |\psi\rangle\langle\psi|$ over these qubits. 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 crucial point, which enables us to solve 2QSAT 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 $|\psi\rangle = |\alpha^{\perp}\rangle \otimes |\xi\rangle$,
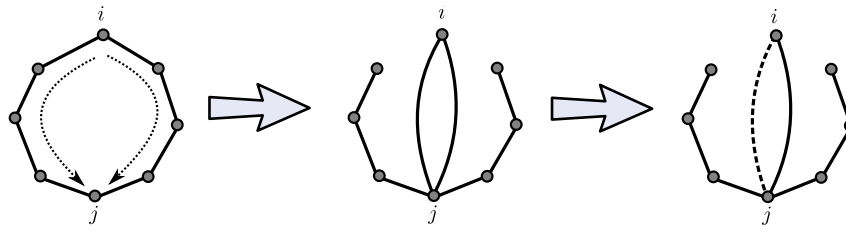
◼ **Figure 1** Handling a contradicting cycle: we slide edges that touch i along the two paths to j until we get a double edge with a 'tail'. We then use a structure lemma to deduce that at least one of these edges can be written as a product projector.

where $|\alpha^\perp\rangle$ is perpendicular to $|\alpha\rangle$ and $|\xi\rangle$ is arbitrary. If the second case happens, we say that state $|\alpha\rangle$ is propagated to state $|\beta\rangle$ by the constraint state $|\psi\rangle$.

This dichotomy enables us to propagate a product state $|s\rangle$ on part of the system until we reach a contradiction, or find that no further propagation is possible and we are left with a smaller Hamiltonian H_s. This smaller Hamiltonian consists of a subset of the original projectors, *without introducing new projectors*. It turns out that once an edge is checked for potential propagation, then no matter whether a propagation happens along the edge or not, the edge can be safely removed without changing the satisfiability. Thus the satisfiability of the original Hamiltonian H is the same as that of the smaller Hamiltonian H_s.

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 $|\alpha\rangle \otimes |\beta\rangle$. In such cases a product state solution must either have $|\alpha^\perp\rangle$ at the first qubit or $|\beta^\perp\rangle$ 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 contradiction, 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, $x_i = 1$. 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. Then using the techniques of 'sliding', which was introduced in Ref. [11], one can show that this cycle is equivalent to a system of one double edge and a 'tail' (see Fig. 1). 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 two possible free choices.

Let us state here that our algorithm works in the algebraic model of computation: we suppose that every arithmetic operation on complex numbers can be done in unit time.

Classically, Davis-Putnam [6] and DPLL algorithms [5] 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 and independently from our work and approximately at the same time, de Beaudrap and Gharibian [7] have also presented a linear time algorithm for quantum 2SAT.

The main difference between the two algorithms is how they deal with instances with only entangled rank-1 projectors. Contrarily to us, [7] handles these instances by using transfer matrix techniques to find discretizing cycles [16]. Most proofs are omitted from this version of the paper owing to space constraints.

2 Preliminaries

2.1 Notation

We will use the notation $[n] = \{1, \ldots, n\}$. 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 ith 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 $|\alpha^\perp\rangle = \alpha_1|0\rangle - \alpha_0|1\rangle$.

We shall denote local projectors either by Π_{ij}, or by Π_e, where $e = (i, j)$. When $i < j$, Π_{ij} is a 2-local projector on the qubits i, j; it can be written as $\Pi_{ij} = \hat{\Pi}_{ij} \otimes \mathbb{I}_{rest}$, where $\hat{\Pi}_{ij}$ is a projector working on $\mathcal{H}_i \otimes \mathcal{H}_j$ and $\mathbb{I}_{rest}$ is the identity operator on the rest of the system. Similarly, when $i = j$, $\Pi_{ii} = \hat{\Pi}_{ii} \otimes \mathbb{I}_{rest}$, where $\hat{\Pi}_{ii}$ is a projector defined in $\mathcal{H}_i$. Often, in order not to overload the notation, we shall use Π_{ij} instead of $\hat{\Pi}_{ij}$, even when acting on states in $\mathcal{H}_i \otimes \mathcal{H}_j$. Similarly, with a slight abuse the notation, we define the *rank* of a projector Π_e to be the dimension of the subspace that its local projector $\hat{\Pi}_e$ projects to, and it will be denoted by $\texttt{rank}(\Pi_e)$. We call a rank-1 projector $\Pi_e = |\psi\rangle\langle\psi|$, *entangled* if $|\psi\rangle$ is an entangled state, and *product* if $|\psi\rangle$ is a product state.

2.2 The 2QSAT problem

A quantum 2-SAT Hamiltonian on an n-qubit system is a Hermitian operator $H = \sum_{e \in I} \Pi_e$, for some $I \subseteq \{(i, j) \in [n] \times [n] : 1 \le i \le j \le n\}$. 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 $i < j$. The single-qubit projectors of H as well as its 2-qubit projectors of rank-3 are called *maximal* rank.

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 also simultaneously the ground state of all local terms. 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 quantum 2-SAT Hamiltonian, $\Pi_e|\Gamma\rangle = 0$ for all $e \in I$. We can also view each local projector as a *constraint* on at most two qubits, then a ground state satisfies every constraint.

It turns out that for the representation of the 2QSAT 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

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

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

To the rank-1 decomposition we associate a weighted, directed multigraph with self-loops $G(H) = (V, E, w)$, 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 in each direction between 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 respectively

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

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 $e \in \{(i, j), (i, j, 1), (i, j, 2)\}$. For a projector Π acting on two qubits, we define its *reverse* projector Π^{rev} by $\Pi^{rev}|\alpha\rangle|\beta\rangle = \Pi|\beta\rangle|\alpha\rangle$, and for $i \leq j$ and $b \in [2]$, we set $\Pi_{ji} = \Pi_{ij}^{rev}$ and $\Pi_{jib} = \Pi_{ijb}^{rev}$. Then for an edge (i, j), its *weight* is defined as $w(i, j) = \Pi_{ij}$, and analogously for an edge (i, j, b), we set $w(i, j, b) = \Pi_{ijb}$.

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. In this representation there is a 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 linked list containing an element for every edge (i, j) or (i, j, b). For an edge (i, j), this element contains j, the projector Π_{ij} and a pointer towards the next element in the list, for an edge (i, j, b) it also contains the value b. The problem 2QSAT is defined formally as follows.

2QSAT
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 2QSAT Hamiltonian composed has a ground state which 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. [3]. The difference here is that we specifically attribute the 2-qubits states in the product state to rank-3 projectors. Just as in Ref. [3], our derivation begins with Theorem 1 of Ref. [3], which we give below. It relies on the notion of a *genuinely entangled* state in an n-qubit system, which is a pure state that is not a product state with respect to any bi-partition of the system. Then Theorem 1 in [3] states

▶ **Proposition 2.** *A 2-local frustration-free Hamiltonian on n qubits which has a genuinely entangled ground state always has a product ground state, whenever $n \geq 3$.*

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

▶ **Fact 3.** *Any 2-dimensional subspace V of the 2-qubit space $\mathbb{C}^2 \otimes \mathbb{C}^2$ contains at least one product state, which can be found in constant time.*

The proofs of Fact 3 and our Product State Theorem are omitted owing to space constraints.

▶ **Theorem 4** (Product State Theorem). *Any frustration-free 2QSAT Hamiltonian has a ground state which is a tensor product of one qubit and two-qubit states, where two-qubit states only appear in the support of rank-3 projectors.*

2.4 Assignments

Let $H = \sum_{e \in I}$ be a 2-local Hamiltonian. By Theorem 4, if H is frustration free then it has a ground state which is the tensor product of 1-qubit and 2-qubit entangled states, where the latter only appear in pairs of qubits corresponding to rank-3 projectors. To build up a ground state of such form, our algorithm will use partial assignments, or shortly assignments. 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 a symbol from the set $\{\ominus, X\}$. 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. The symbol $\ominus$ is used for unassigned variables, and the symbol X is used when several values are assigned to some variable.

We define the *support* of s by $\mathsf{supp}(s) = \{i \in [n] : s(i) \neq \ominus\}$. The assignment s is *empty* if $\mathsf{supp}(s) = \emptyset$. When there is no danger of confusion, we will denote the empty assignment also by $\ominus$. We say that an assignment is *coherent* if for every i, we have $s(i) \neq X$, and whenever $s(i) = |\gamma\rangle_{ij}$, we also have $s(j) = |\gamma\rangle_{ji}$. For coherent 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 $s'(i) = s(i)$. A coherent assignment is *total* if $s(i) \neq \ominus$, for all i. Clearly, a coherent assignments 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 a coherent 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 a coherent 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 a coherent assignment s a *pre-solution* if it has a total extension s' satisfying every constraint in H, and we call s is 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. A coherent assignment s is *closed* if $\mathsf{supp}(s) \cap V_s = \emptyset$.

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. [16], and can in fact be traced back also to Bravyi's original work. Here, we shall adopt the following definition

▶ **Definition 5** (Propagation). Let $\Pi_e = |\psi\rangle\langle\psi|$ be a rank-1 projector acting on variables i, j, and let $|\alpha\rangle$ be either 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 a phase, there exists a unique 1-qubit state $|\beta\rangle$ such that $\Pi_e|\alpha\rangle_i \otimes |\beta\rangle_j = 0$. In such a case we say that $|\alpha\rangle$ is propagated to $|\beta\rangle$ along Π_e, or that Π_e propagated $|\alpha\rangle$ to $|\beta\rangle$.

We establish a sequence of Lemmas on propagation whose proofs are not discussed here owing to space contraints. The first lemma shows how the propagation properties of $\Pi_e = |\psi\rangle\langle\psi|$ are determined by entanglement in $|\psi\rangle$.

▶ **Lemma 6.** *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 $|\alpha\rangle_i \neq |\alpha'\rangle_i$ then $|\beta(\alpha)\rangle_j \neq |\beta(\alpha')\rangle_j$. This propagation can be calculated in constant time. 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 $|y^\perp\rangle_j$.*

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

▶ **Lemma 7** (Single qubit propagation). *Consider a frustration-free 2QSAT 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 $|\Gamma\rangle = |\alpha\rangle_i \otimes |rest\rangle$. Then:*
1. *If Π_e propagates $|\alpha\rangle_i$ to $|\beta\rangle_j$ then necessarily $|rest\rangle = |\beta\rangle_j \otimes |rest'\rangle$.*
2. *$|\Gamma\rangle$ is also a ground state of the 2QSAT Hamiltonian $H - \Pi_e$.*

▶ **Lemma 8** (Entangled 2-qubits propagation). *Consider a frustration-free 2QSAT system H with a rank-1 projector $\Pi_e = |\psi\rangle\langle\psi|$ between qubits i, j. Assume that H has a ground state of the form $|\Gamma\rangle = |\phi\rangle_{ik} \otimes |rest\rangle$, where $|\phi\rangle$ is an entangled state on qubits i, k with $k \neq j$. Then:*
1. *$|\psi\rangle$ is a product state $|\psi\rangle = |x\rangle|y\rangle$.*
2. *Π_e propagates $|\phi\rangle$ to $|y^\perp\rangle$ and necessarily $|rest\rangle = |y^\perp\rangle_j \otimes |rest'\rangle$.*
3. *$|\Gamma\rangle$ is also a ground state of the 2QSAT Hamiltonian $H - \Pi_e$.*

Let H be a 2QSAT Hamiltonian in rank-1 decomposition, let s be a coherent 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 when a value given to variable i is first propagated along all possible projectors, followed by these propagated values being propagated on their turn, and so on until no more values assigned during this process can be propagated further. The propagation can start 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 it can get started when $s(i) = \ominus$, in which case we shall explicitly choose a 1-qubit state $|\alpha\rangle$ and assign it to i.

Now, let s, i and $|\delta\rangle$ be such that $s(i) \in \{\ominus, |\delta\rangle\}$. We say that an edge $e \in E_s$, in the constraint graph G_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. We further 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.

Let us suppose that $s' = \mathsf{ext}_s^{\mathsf{prop}}(i, |\delta\rangle)$ is also coherent. 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

---

Algorithm 1 Propagation$(s, G_s, i, |\delta\rangle)$

$s(i) := |\delta\rangle$

create a list L and a queue Q, and **put** i into Q

while s is coherent and Q is not empty **do**

    remove the head j of Q

    for all edge e from j to k **do**

        remove e from E_s

        if e propagates $s(j)$ **then**

$$s(k) := \begin{cases} \mathsf{prop}(s, e, s(j)) & \text{if } s(k) = \ominus \\ X & \text{if } s(k) \notin \{\ominus, \mathsf{prop}(s, e, s(j))\} \end{cases}$$

            enqueue k

        if e is not propagating and k is not in L **then put** k into L

    remove j from V_s

if s is not coherent **return** "unsuccessful"

for all k in L **do**

    for all edges e from k to ℓ **do**

        if ℓ was removed from V_s **then remove** e

    if all edges outgoing from k were removed **then remove** k from V_s

---

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', none of the edges in E_3 is satisfied. Therefore $G_{s'}$ is nothing but $G(V_s \setminus V_s^{\mathsf{prop}}(i, |\delta\rangle))$ without the isolated vertices, 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. The leaves of the tree are vertices in $V_s \setminus V_s^{\mathsf{prop}}(i, |\delta\rangle)$. The algorithm Propagation uses a temporary queue Q to implement this process.

▶ **Lemma 9** (Propagation Lemma). *Let Propagation$(s, G_s, i, |\delta\rangle)$ be called when H_s doesn't have rank-3 constraints, and $s(i) \in \{\ominus, |\delta\rangle\}$. Let s' and $G' = (V', E')$ be the outcome of the procedure. Then:*

1. *If Propagation$(s, G_s, i, |\delta\rangle)$ doesn't 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 which is an extension of s and for which $z(i) = |\delta\rangle$.*

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

4 The main algorithm

4.1 Description of the algorithm

We now give in broad strokes the description of our algorithm called 2QSATSolver. 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 $\ominus$, and graphs G_0 and G_1 in the adjacency list representation, initialized to $G(H)$. The algorithm consists of four phases, and except the first one, each phase consists of

---

Algorithm 2 2QSATSolver$(G(H))$

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

MaxRankRemoval()   ▷ Remove maximal rank constraints

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

while there exists in G_0 a product edge with constraint $|\alpha_0^\perp\rangle_{i_0} \otimes |\alpha_1^\perp\rangle_{i_1} \langle\alpha_0^\perp|_{i_0} \otimes \langle\alpha_1^\perp|_{i_1}$
do
     ParallelPropagation$(i_0, |\alpha_0\rangle, i_1, |\alpha_1\rangle)$ ▷ Remove product constraints

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

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

---

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 when several values are assigned to the same variable during a necessary propagation process.

In the case of a frustration-free Hamiltonian, at the beginning and 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, ensuring that the useless work done is proportional to the useful work.

In the first phase the procedure MaxRankRemoval satisfies, if possible, all constraints of maximal rank. In the second phase all these assignments are propagated, which, if successful, result 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 a constraint, an arbitrary value is tried and propagated. In 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 case of success we are left with a satisfying assignment and the empty constraint graph. Theorem 1 is an immediate consequence of the following result.

▶ **Theorem 10.** *Let $G(H) = (V, E)$ be the constraint graph of a 2-local Hamiltonian. Then:*
1. *If H is frustration-free, the algorithm 2QSATSolver$(G(H))$ outputs a ground state $|s\rangle$.*
2. *If H is not frustration-free, the algorithm 2QSATSolver$(G(H))$ outputs "H is unsatisfiable".*
3. *The running time of the algorithm is $O(|V| + |E|)$.*

The proofs of the series of lemmas in the following sections is omitted. Theorem 10 will be proven in Section 4.5.

Algorithm 3 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($s_1, G_1, i_1, |\alpha_1\rangle$)
> **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($s_1, G_1, i_1, |\alpha_1\rangle$)
> $s_1 := s_0, \ G_1 := G_0$

4.2 Max rank removal

The MaxRankRemoval procedure is conceptually very simple. Since every maximal rank constraint has a unique solution (up to a global phase), it makes this assignment for each constraint, and then checks if this is globally consistent. The description of the procedure and the proof of Lemma 11 are omitted.

▶ **Lemma 11.** *Let s_0, G_0, s_1, G_1 be the outcome of MaxRankRemoval. Then:*

1. *If MaxRankRemoval doesn't output "H is unsatisfiable" then s_0 is coherent, it satisfies every maximal rank constraint, $G_0 = G(H_{s_0})$ and $s_0 = s_1, G_0 = G_1$. Moreover, if H is satisfiable then s_0 is a pre-solution.*
2. *If MaxRankRemoval outputs "H is unsatisfiable" then indeed H is unsatisfiable.*
3. *The complexity of the procedure is $O(|V| + |E|)|$.*

4.3 Algorithm ParallelPropagation

The procedure ParallelPropagation is called when s_0 is a closed assignment, and in G_{s_0} there is a product edge. Since 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.

▶ **Lemma 12.** *Let ParallelPropagation be called when s_0 is closed, H_{s_0} doesn't have rank-3 constraints, $G_0 = G_{s_0}$, in G_0 there exists a product edge from i_0 to i_1 with constraint $|\alpha_0^\perp\rangle \otimes |\alpha_1^\perp\rangle$, and $s_1 = s_0, G_1 = G_0$. Let s_0', s_1', G_0', G_1' be the outcome of the procedure. Then:*

1. *If ParallelPropagation doesn't output "H is unsatisfiable" then s_0' is a proper closed extension of s_0, $G_0' = G_{s_0'}$, and $s_1' = s_0', G_1' = G_0'$. Moreover, if s is a pre-solution then s_0' is a pre-solution.*
2. *If ParallelPropagation outputs "H is unsatisfiable" then indeed H is unsatisfiable.*
3. *The complexity of the procedure is $O(|E_{s_0}| - |E_{s_0'}|)$.*

4.4 Algorithm ProbePropagation

The procedure ProbePropagation is evoked when s_0 is a closed assignment, and in G_{s_0} there are only entangled constraints. It picks an arbitrary vertex in $i \in V_s$, assigns $|0\rangle$ (an arbitrary value) to it, and propagates this choice. In the lucky case of successful propagation this is repeated. Otherwise, we reach a contradiction: there is some $j \in V_s$, such that two propagating paths assign different values to it. We prove below the Sliding Lemma which already appeared in Ref. [11]. It implies that when $i_0 \to i_1 \to \ldots \to i_k$ is a propagating path of

---

Algorithm 4 ProbePropagation(i)

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 $|\alpha^\perp\rangle_i \otimes |\beta^\perp\rangle_j$ in the two dimensional space $\mathsf{span}\{\mathsf{slide}(p_1), \mathsf{slide}(p_2)\}$
  undo Propagation($s_0, G_0, i, |0\rangle$)
  ParallelPropagation($i, |\alpha\rangle, j, |\beta\rangle$)

---

entangled rank-1 projectors, the ground space of the Hamiltonian $\Pi_{i_0,i_1} + \Pi_{i_1,i_2} + \ldots + \Pi_{i_{k-1},i_k}$ is equal to the ground state of the Hamiltonian $\Pi_{i_0,i_k} + \Pi_{i_1,i_2} + \ldots + \Pi_{i_{k-1},i_k}$, where Π_{i_0,i_k} is a new projector defined on the qubits (i_0, i_k) that replaces the projector Π_{i_0,i_1}. Graphically, this can be viewed as if we are sliding the (i_0, i_1) edge on the path $i_1 \to \ldots \to i_k$. Therefore, if we have two propagating paths starting at i and ending at j, they define two projectors on qubits (i, j). As we shall see, if these two paths are contradicting then necessarily the two projectors are different, which by Lemma 3 implies the existence of a product constraint between (i_0, i_k) variables. In this case, we can proceed by calling the procedure ProbePropagation.

▶ **Lemma 13** (Sliding Lemma). *Consider a system on 3 qubits i, j and k. Suppose that we have a two rank-1 constraints $\Pi_1 = |\psi_1\rangle\langle\psi_1|_{ij}$ on qubits (i, j) and $\Pi_2 = |\psi_2\rangle\langle\psi_2|_{jk}$ on qubits (j, k). If $|\psi_2\rangle$ is entangled, there is another rank-1 constraint $\Pi_3 = |\psi_3\rangle\langle\psi_3|_{ik}$ 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 $|\beta\rangle_k$, then it is also propagated to $|\beta\rangle_k$ directly via Π_3.*

Applying Lemma 13 iteratively, we reach the following corollary

▶ **Corollary 14.** *Let $H = \sum_{e \in I} H_e$ be a 2-local Hamiltonian in rank-1 decomposition. Let $i_0, i_1, \ldots i_k$ be vertices in V, and let e_j be an edge from i_j to i_{j+1}, for $j = 0, \ldots, k-1$ such that the rank-1 constraints Π_{e_j} are entangled. Then there exists a 2-qubit entangled state $|\gamma\rangle$ between i_0 and i_k such that the ground space of $\sum_{j=0}^{k-1} \Pi_{e_j}$ is identical to the ground space of $\sum_{j=1}^{k-1} \Pi_{e_j} + |\gamma\rangle\langle\gamma|_{i_0,i_k}$. Moreover, if $|\alpha\rangle_{i_0}$ is propagated to $|\beta\rangle_{i_k}$ along the path, then it is also propagated directly by $|\gamma\rangle\langle\gamma|_{i_0,i_k}$.*

We will denote the state $|\gamma\rangle$ in the conclusion of the corollary by $\mathsf{slide}(p)$.

▶ **Lemma 15.** *Let ProbePropagation be called when s_0 is closed, H_{s_0} has only rank-1 entangled constraints, $G_0 = G_{s_0}$, and $s_1 = s_0, G_1 = G_0$. Let s_0', s_1', G_0', G_1' be the outcome of the procedure. Then:*
1. *If ProbePropagation doesn't output "H is unsatisfiable" then s_0' is a proper closed extension of s_0, $G_0' = G_{s_0'}$, and $s_1' = s_0', G_1' = G_0'$. Moreover, if s is a pre-solution then s_0' is a pre-solution.*
2. *If ParallelPropagation outputs "H is unsatisfiable" then indeed H is unsatisfiable.*
3. *The complexity of the procedure is $O(|E_{s_0}| - |E_{s_0'}|)$.*

4.5 Analysis of the algorithm

Proof of Theorem 10. If H is frustration free then by Lemma 11 MaxRankRemoval outputs a pre-solution s_0 that satisfies every maximal rank constraint. By the Propagation Lemma, at

the end of Phase two, in addition s_0 is a closed assignment. By Lemma 12 ParallelPropagation outputs s_0 such that there are only entangled constraints in H_s. By Lemma 15 at the end of the algorithm in addition H_s is empty, and therefore s is a solution.

If the algorithm doesn't output "H is unsatisfiable" then by Lemma 11, by the Propagation Lemma, and by Lemmas 12 and 15 it outputs a coherent assignment s such that G_s is the empty graph, and therefore s is a solution.

The complexity of MaxRankRemoval by Lemma 11 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, and Lemmas 12 and 15 is a telescopic sum which sums up to also $O(|E|)$. ◀

References

1 B. Aspvall, M. Plass, and R. E. Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Inf. Process. Lett.*, 8(3):121–123, 1979. Erratum: Information Processing Letters 14(4): 195 (1982). `doi:10.1016/0020-0190(79)90002-4`.

2 S. Bravyi. Efficient algorithm for a quantum analogue of 2-SAT. In K. Mahdavi, D. Koslover, and L. L. Brown, editors, *Contemporary Mathematics*, volume 536. American Mathematical Society, 2011. URL: `http://arxiv.org/abs/quant-ph/0602108`.

3 J. Chen, X. Chen, R. Duan, Z. Ji, and B. Zeng. No-go theorem for one-way quantum computing on naturally occurring two-level systems. *Physical Review A*, 83(5):050301, 2011.

4 S. Cook. The complexity of theorem proving procedures. In *Proceedings of the Third Annual ACM Symposium*, pages 151–158, New York, 1971. ACM.

5 M. Davis, G. Logemann, and D. Loveland. A machine program for theorem-proving. *Commun. ACM*, 5(7):394–397, July 1962. `doi:10.1145/368273.368557`.

6 M. Davis and H. Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, July 1960. `doi:10.1145/321033.321034`.

7 N. de Beaudrap and S. Gharibian. A linear time algorithm for quantum 2-SAT. *CoRR*, abs/1508.07338, 2015. To appear in 31st Conference on Computational Complexity. URL: `http://arxiv.org/abs/1508.07338`.

8 J. Eisert, M. Cramer, and M. Plenio. Area laws for the entanglement entropy – a review. *Reviews of Modern Physics*, 82(277), 2010.

9 S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM J. Comput.*, 5(4):691–703, 1976. `doi:10.1137/0205048`.

10 D. Gosset and D. Nagaj. Quantum 3-sat is qma1-complete. *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, 0:756–765, 2013. `doi:10.1109/FOCS.2013.86`.

11 Z. Ji, Z. Wei, and B. Zeng. Complete characterization of the ground-space structure of two-body frustration-free hamiltonians for qubits. *Physical Review A*, 84:042338, 2011.

12 R. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. URL: `http://www.cs.berkeley.edu/~luca/cs172/karp.pdf`.

13 A. Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, 2003. `doi:10.1016/S0003-4916(02)00018-0`.

14 A. Kitaev, A. Shen, and M. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society, Boston, MA, USA, 2002.

15 M. 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`.

16 C. Laumann, R. Moessner, A. Scardicchio, and S. Sondhi. Phase transitions and random quantum satisfiability. *Quantum Information & Computation*, 10(1), 2010.

17 L. Levin. Universal sequential search problems. *Problems of Information Transmission*, 9(3):265–266, 1973.

18 C. Papadimitriou. On selecting a satisfying truth assignment (extended abstract). In *32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 1-4 October 1991*, pages 163–169, 1991. `doi:10.1109/SFCS.1991.185365`.

19 S. Sachdev. *Quantum phase transitions*. Wiley Online Library, 2007.

20 G. Vidal, J.-I. Latorre, E. Rico, and A. Kitaev. Entanglement in quantum critical phenomena. *Phys. Rev. Lett.*, 90:227902, Jun 2003. `doi:10.1103/PhysRevLett.90.227902`.