Statistical mechanics of classical and quantum computational complexity

C. R. Laumann,[1] R. Moessner,[2] A. Scardicchio,[3] and S. L. Sondhi[4]

[1] *Department of Physics, Princeton University, Princeton, NJ 08544, USA*
[2] *Max Planck Institut fur Physik Komplexer Systeme, 01187 Dresden, Germany*
[3] *Abdus Salam International Centre for Theoretical Physics, Strada Costiera 11, 34014 Trieste, Italy*
[4] *Department of Physics, Princeton University, Princeton, NJ 08544*

The quest for quantum computers is motivated by their potential for solving problems that defy existing, classical, computers. The theory of computational complexity, one of the crown jewels of computer science, provides a rigorous framework for classifying the hardness of problems according to the computational resources, most notably time, needed to solve them. Its extension to quantum computers allows the relative power of quantum computers to be analyzed. This framework identifies families of problems which are likely hard for classical computers ("NP-complete") and those which are likely hard for quantum computers ("QMA-complete") by indirect methods. That is, they identify problems of comparable worst-case difficulty without directly determining the individual hardness of any given instance. Statistical mechanical methods can be used to complement this classification by directly extracting information about particular families of instances—typically those that involve optimization—by studying random ensembles of them. These pose unusual and interesting (quantum) statistical mechanical questions and the results shed light on the difficulty of problems for large classes of algorithms as well as providing a window on the contrast between typical and worst case complexity. In these lecture notes we present an introduction to this set of ideas with older work on classical satisfiability and recent work on quantum satisfiability as primary examples. We also touch on the connection of computational hardness with the physical notion of glassiness.

I. INTRODUCTION

A large and exciting effort is underway to build quantum computers. While the roots of this effort lie in the deep insights of pioneers such as Feynman and Deutsch, what triggered the growth was the discovery by Shor that a quantum computer could solve the integer factoring problem efficiently — a feat currently beyond the reach of classical computers. In addition to the desire to create useful devices intrinsically more powerful than existing classical computers, the challenge of creating large quantum systems subject to precise control has emerged as the central challenge of quantum physics in the last decade.

The first and primary objective - that of enhanced computational power - has in turn spurred the founding of quantum computer science and the development of a rigorous theory of the (potential) power of quantum computers: quantum complexity theory. This theory builds on the elegant ideas of classical complexity theory to classify computational problems according to the resources needed to solve them as they become large. The distinction between polynomial scaling of resources, most notably time, and super-polynomial scaling (e.g. exponential) generates a robust distinction between easy and hard problems.

While this distinction is easily made in principle, in practice complexity theory often proceeds by the powerful technique of assigning guilt by association: more precisely, that of classifying problems by mapping between them. This allows the isolation of sets of problems that encapsulate the difficulty of an entire class: for example, the so-called satisfiability problem (SAT) captures the difficulty of all problems whose solution can be easily checked by a classical computation; the quantum satisfiability problem (QSAT) does the same for quantum computers, as we will explain later in these notes. The solution of these problems would thus enable the solution of the vast set of of all checkable problems. This implication is a powerful argument that both SAT and QSAT must be hard.

This kind of indirect reasoning is very different from the way physicists normally approach problems: one of the purposes of these notes is to help physics readers appreciate the power of the computer science approach. However, the direct approach of examining actual problem instances and attempting to come up with algorithms for them is, of course, also important and this is where physicists are able to bring their own methods to bear. Specifically, physicists have applied themselves to the task of trying to understand problems such as the two satisfiability problems. These can be expressed as optimization problems and thus look much like the Hamiltonian problems the field is used to. Even more specifically, they have studied ensembles of these problems with a variety of natural probability measures in order to reveal features of the hard instances.

As is familiar from the statistical mechanical theory of disordered systems such as spin glasses, studying a random ensemble brings useful technical simplifications that allow the structure of a typical instance to be elucidated with less trouble than for a specific capriciously picked instance. This has enabled the identification of phase transitions as parameters defining the ensembles are changed - exactly the kind of challenge to warm a statistical physicist's heart. A further major product of such work, thus far largely in the classical realm, has been the identification of obstacles to the solution of such typical instances by large classes of algorithms and the construction of novel algorithms that avoid these pitfalls. We note that this focus on typical instances also usefully complements the standard results of complexity theory which are necessarily controlled by the worst cases - instances that would take the longest to solve but which may be very unusual. This is then an independent motivation for studying such ensembles.

Certainly, the flow of ideas and technology from statistical mechanics to complexity theory has proven useful. In return, it is useful to reflect on what complexity theory has to say about physical systems. Here the central idea - whose precise version is the Church-Turing hypothesis - is that a physical process is also a computation in that it produces an output from an input. Thus if a complexity theoretic analysis indicates that a problem is hard, any physical process that encodes its solution must take a long time. More precisely, the existence of hard optimization problems implies the existence of a class of Hamiltonians whose ground states can only be reached in a time that scales exponentially in the volume of the system *irrespective* of the processes used.

This sounds a lot like what physicists mean by glassiness. We remind the reader that physical systems typically exhibit symmetric, high temperature or large quantum fluctuation phases, with a characteristic equilibration time that is independent of the size of the system. At critical points, or in phases with broken continuous symmetries, algebraic dependences are the norm. But glassy systems exhibit much slower relaxation and thus present a challenge to experimental study in addition to theoretical understanding. Indeed, there is no settled understanding of laboratory glassiness. Consequently, the complexity theoretic arguments that imply the existence of glassy Hamiltonians, both in the classical and quantum cases, ought to be interesting to physicists. That said, we hasten to add that the connection is not so simple for two reasons. First, complexity theoretic results do not always hold if we restrict the degrees of freedom to live in Euclidean space - say on regular lattices - and require spatial locality. Thus hard Hamiltonians in complexity theory can look unphysical to physicists. Nonetheless, there are many interesting low dimensional (even translation invariant) problems which are hard in the complexity theoretic sense [2, 10]. Second, the physical processes intrinsic to a given system can sometimes be slow for reasons of locality or due to energetic constraints which are ignored when one is considering the full set of algorithms that can solve a given optimization problem. Still, we feel

this is a direction in which Computer Science has something to say to Physics and we refer readers to a much more ambitious manifesto along this axis by Aaronson [1] for stimulation.

In the bulk of these notes we provide an introduction to this complex of ideas, which we hope will enable readers to delve further into the literature to explore the themes that we have briefly outlined above. In the first part, we provide a tutorial on the basics of complexity theory including sketches of the proofs of the celebrated proofs of NP and QMA completeness for the satisfiability and quantum satisfiability problems. In the second part, we show how statistical methods can be applied to these problems and what information has been gleaned. Unsurprisingly, the quantum part of this relies on recent work and is less developed than the classical results. In the concluding section we list some open questions stemming from the quantum results to date.

II. COMPLEXITY THEORY FOR PHYSICISTS

Complexity theory classifies how "hard" it is to compute the solution to a problem as a function of the *input size* N of the problem instance. As already mentioned above, algorithms are considered *efficient* if the amount of time they take to complete scales at most polynomially with the size of the input and *inefficient* otherwise. The classification of algorithms by asymptotic efficiency up to polynomial transformations is the key to the robustness of complexity theoretic results, which includes the independence from an underlying model of computation.

In this line of reasoning, if P $\neq$ NP, as is the current consensus, there are natural classes of problems which cannot be solved in polynomial time by any computational process, including any physical process which can be simulated by computer. Complexity theory provides its own guide to focusing our attention on a certain set of NP problems, those termed NP-complete, which capture the full hardness of the class NP. In particular, the problem of Boolean satisfiability of 3-bit clauses, 3-SAT, is NP-complete and therefore can encode the full hardness of the class NP.

The advent of the quantum computer modifies the above reasoning only slightly. It appears that quantum computers are somewhat more powerful than their classical counterparts, so that we must introduce new quantum complexity classes to characterize them. Nonetheless, analogous statements hold within this new framework: quantum polynomial (BQP) is larger than classical polynomial (P) but not powerful enough to contain classical verifiable (NP), nor quantum verifiable (QMA). If we wish to study the particularly hard quantum problems, we may turn to the study of QMA-complete problems such as LOCAL HAMILTONIAN and the closely related QSAT.

In this section, we provide a concise review of the key concepts for the above argument culminating in a discussion of worst-case hardness and the Cook-Levin theorem, showing the existence of NP-complete problems, and the quantum analogues due to Kitaev and Bravyi. This story motivates and complements the statistical study of 'typical' instances of 3-SAT, 3-QSAT and other classical and quantum hard optimization problems, which are discussed in the following sections.

A. Problems, instances, computers and algorithms

The success of complexity theory as a classification scheme for the "hardness" of problems is in part due to the careful definitions employed. Here we sketch the most important aspects of these concepts and leave the rigorous formalism for the textbooks, of which we particularly recommend Arora and Barak to the interested reader [9]. We have taken a particular path through the forest of variations on the ideas below and do not pretend to completeness.

Throughout these notes we focus on so-called *decision problems*, that is Yes/No questions such as "Does the Hamiltonian H have a state with energy less than E?" rather than more general questions like "What is the ground state energy of H?" This restriction is less dramatic than it seems – many general questions may be answered by answering a (reasonably short) sequence of related Yes/No questions, much like playing the game twenty questions – and it significantly simplifies the conceptual framework we need to introduce. Moreover, many of the essential complexity theoretic results arise already within the context of decision problems.

A decision problem, then, is a question that one may ask of a class of *instances*. For example, the DIVIDES problem asks "Does a divide b?" for integers a and b. Leaving the variables a and b unspecified, we clearly cannot yet answer this question. An *instance* of DIVIDES might be "Does 5 divide 15?" A moment's thought now reveals that a definitive answer exists: Yes. We refer to instances of a problem as Yes-instances (No-instances) if the answer is Yes (No). We follow computer science convention by giving problems fully capitalized names.[1]

What does it mean to solve a problem? We would not feel we had solved the problem if we could only answer a few specific instances. On the other hand, we certainly could not expect to have a book containing the (infinite) table

[1] We trust this will not give physics readers PROBLEMS.

of answers to all possible instances for easy reference. Thus, we want a general *algorithm* which, given an arbitrary instance, provides us with a step-by-step recipe by which we can compute the answer to the instance. Some physical object must carry out the algorithmic instructions and it is this object that we call a *computer* – whether it is a laptop running C code, ions resonating in an ion trap or a sibling doing long division with pencil and paper. Thus, a solution to a decision problem is an algorithm which can decide arbitrary instances of the problem when run on an appropriate computer. Such an algorithm for a decision problem is often called a *decision procedure*.

Clearly it is less work to answer the DIVIDES problem for small numbers than for large. "Does 5 divide 15?" takes essentially no thought at all while "Does 1437 divide 53261346150?" would take a few moments to check. We therefore define the *input size* (or just *size*) of an instance as the number of symbols we need to specify the instance. In the DIVIDES problem, we could take the size as the number of symbols needed to specify the pair (a, b). The size N of $(5, 15)$ would be 6 while that of $(1437, 53261346150)$ is 18.

Computer scientists measure the *efficiency* of an algorithm by considering the asymptotic scaling of its resource consumption with the input size of the problem. More precisely, consider the finite but large collection of all possible problem instances whose size is no greater than N. For each of these instances, the algorithm will take some particular amount of time. For the finite collection at size N, there will be a worst-case instance which takes more time T than any of the others at that size. Complexity theory generally focusses on the scaling of this worst-case time T as a function of input size N as $N \to \infty$. Clearly, the slower the growth of T with N, the more efficient the algorithm for large inputs. Indeed, algorithmic procedures are considered efficient so long as T grows at most polynomially with N, for any particular polynomial we like. Thus both linear and quadratic growth are efficient, even though linear growth clearly leeds to faster computations, at least for sufficiently large N. Anything slower, such as $T = O(e^N)$, is inefficient.

For example, the most famous decision procedure for DIVIDES – long division – takes of order $T = O(\log b \times \log a) \leq O(N^2)$ arithmetic steps to perform the division and check the remainder is 0. That T grows with a and b logarithmically corresponds nicely to our intuition that bigger numbers are harder to divide, but not too much harder. It is instructive to consider a different, inefficient, algorithm for the same problem. Suppose we had not yet learned how to divide but knew how to multiply. We might try the following decision procedure: try to multiply a with every number c from 1 up to b and check if $ac = b$. This trial-and-error approach would take $T = O(\log a \times \log b \times b) \approx O(e^{cN})$ to try out all the possibilities up to b. Even for relatively small instances, this approach would quickly become prohibitively time consuming – simply enumerating all of the numbers of up to 30 digits at one per nanosecond would take longer than the age of the universe!

Finally, we lift our classification of algorithm efficiency to a classification of problem hardness: A problem is *tractable* if there exists an efficient algorithm for solving it and it is *intractable* otherwise. By this definition, DIVIDES is tractable (long division solves it efficiently), despite the existence of alternative slower algorithms. As we will discuss further in the following sections, we can rarely *prove* that no efficient algorithm exists for a given problem, but complexity theory nonetheless offers strong arguments that certain large classes of problems are intractable in this sense.

Computers are clearly central to the determination of the difficulty of problems – we classify problems according to the efficiency of the computational algorithms that exist for treating them. In addition to the time taken, we can measure the resource requirements in various implementation dependent ways – memory consumed, number of gates required, laser pulse bandwidth, quantity of liquid Helium evaporated. One might expect that the kind of computer that we use would greatly influence any complexity classification. At the very least, your laptop will be faster than your brother at dividing 1000 digit numbers. The beauty of the definition of efficiency by polynomial scaling is that many of these implementation dependent details drop out and we really can focus on the time efficiency as an overall measure of difficulty.[2]

The robustness of these definitions follows from one of the great ideas of complexity theory: up to polynomial overheads, any reasonable classical computer may be simulated by any other. This is known as the strong Church-Turing hypothesis and, as its name suggests, is only a conjecture. Nonetheless, it has been examined and confirmed for many particular models of classical computation[3] and is widely believed to hold more generally. This is the reason for defining efficiency up to polynomial scaling: since any computer can simulate the operation of any other up to polynomial overheads, all computers can solve the same problems efficiently. In Section II C below, we consider the most important classical *complexity classes* that arise from these coarse but robust definitions of efficiency.

The careful reader will have noticed that we restricted our statement of the Church-Turing hypothesis to *classical* computers. It is widely believed that classical computers *cannot* efficiently simulate quantum systems. Certainly,

---

[2] In practice the amount of memory or the number of cores in a workstation regularly limits its ability to do computations. Since in finite time, even a parallel computer can only do a finite amount of work or address a finite amount of memory, a polynomial bound on T also provides a polynomial bound on the space requirements. Likewise, finite parallelization only provides constant time improvements. More refined classifications can be made by restricting resource consumption more tightly but we will not consider them here.

[3] For example, Turing machines, Boolean circuit models and your laptop.

directly simulating Schrödinger's equation on a polynomially sized classical computer is problematic since the Hilbert space is exponentially large. On the other hand, if we had a quantum computer with which to do our simulation, the state space of our computer would also be a Hilbert space and we could imagine representing and evolving complex states of the system by complex states and evolutions of the quantum computer. This reasoning leads to the strong *quantum* Church-Turing hypothesis: that any reasonable quantum computer may be efficiently simulated by any other. With this hypothesis in hand, we may proceed to develop a robust classification of *quantum* complexity classes, as in Section II D.

B. Polynomial reductions and worst-case behavior

Reduction is the most important tool in complexity theory. A decision problem A reduces to another problem B if there is a polynomial time algorithm which can transform instances of A into instances of B such that Yes-instances (No-instances) of A map to Yes-instances (No-instances) of B. In this case, B is at least as hard as A: any algorithm which could efficiently decide B would be able to efficiently decide A as well. Just use the transformation to convert the given instance of A into an instance of B and then apply the efficient algorithm for B.

Reductions formalize the interrelationships between problems and allow us to show that new problems are actually part of known classes. Obviously, if we can reduce a problem A to a problem B that we know how to solve efficiently, we have just shown how to solve A efficiently as well. Conversely, if we have a problem C which we believe is *intractable* – that is, not solvable by an efficient algorithm – and we can reduce it to another problem D, that suggests D should also be intractable. Using this logic we can try to show that all kinds of interesting problems ought to be intractable if we can find one to start with.

C. Classical: P and NP

The most important complexity class is known as P – this is the class of decision problems which a classical computer can decide efficiently. More precisely, a decision problem is in P if there exists an algorithm that runs in polynomial time as a function of the input size of the instance and outputs Yes or No depending on whether the instance is a Yes-instance or No-instance of the problem. From a logical point of view, to show that a given problem is in P we need to provide an efficient procedure to decide arbitrary instances. Colloquially, P is the class of problems that are easy to solve.

We have already discussed one example, the DIVIDES problem, for which long division constitutes a polynomial time decision procedure. Another example is given by the energy evaluation problem: "Does a specific configuration σ of a classical Ising Hamiltonian $H = \sum J_{ij}\sigma_i\sigma_j$ have energy less than E?" Here the instance is specified by a configuration made of N bits, a Hamiltonian function with N^2 coupling terms and a threshold energy E (where all real numbers are specified with some fixed precision). Since we can evaluate the energy $H(\sigma)$ using of order N^2 multiplications and additions and then compare it to E, this problem is also in P.

The second most important complexity class is NP: this is the class of decision problems for which there exists a scheme by which Yes-instances may be efficiently verified by a classical computation. We may think of this definition as a game between a prover and a verifier in which the prover attempts, by hook or by crook, to convince the verifier that a given instance is a Yes-instance. The prover provides the verifier with a proof of this claim which the verifier can efficiently check and either Accept or Reject. NP places no restrictions on the power of the prover – only that Yes-instances must have Acceptable proofs, No-instances must not have Acceptable proofs and that the verifier can decide the Acceptability of the proof efficiently. We note that there is an intrinsic asymmetry in the definition of NP: we do not need to be able to verify that a No-instance is a No-instance.

For example, the problem "Does the ground state of the Hamiltonian $H = \sum J_{ij}\sigma_i\sigma_j$ have energy less than E?" has such an efficient verification scheme. If the prover wishes to show that a given H has such low energy states, he can prove it to the verifier by providing some configuration σ which he claims has energy less than E. The skeptical verifier may efficiently evaluate $H(\sigma)$ using the energy evaluation algorithm outlined above and if indeed $H(\sigma) < E$, the skeptic would Accept the proof. If H did not have such low energy states, then no matter what the prover tried, he would be unable to convince the verifier to Accept.

At first brush, NP seems a rather odd class – why should we be so interested in problems whose Yes-instances may be efficiently checked? Of course, any problem we can decide efficiently (in P) can be checked efficiently (because we can simply decide it!). What of problems outside of NP? These do not admit efficient verification schemes and thus certainly cannot have efficient decision procedures. Moreover, even if, by some supernatural act of intuition (not unusual in theoretical physics), we guess the correct answer to such a problem, we would not be able to convince

anybody else that we were correct. There would be no efficiently verifiable proof! Thus, NP is the class of problems that we could ever hope to be convinced about.

Since 1971, the outstanding question in complexity theory (worth a million dollars since the new millennium), has been "Is P = NP?" This would be an astonishing result: it would state that all of the difficulty and creativity required to come up with the solution to a tough problem could be automated by a general purpose algorithm running on a computer. The determination of the truth of theorems would reduce to the simple matter of asking your laptop to think about it. Since most scientists believe that there are hard problems, beyond the capability of general purpose algorithms, the consensus holds that P $\neq$ NP.[4]

D. Quantum: BQP and QMA

The most important quantum complexity class is BQP – this is the class of decision problems which a quantum computer can decide efficiently with bounded error (the B in the acronym). Since general quantum algorithms have intrinsically stochastic measurement outcomes, we have no choice but to allow for some rate of false-positive and false-negative outcomes. As long as these rates are bounded appropriately (say by 1/3), a few independent repetitions of the quantum computation will exponentially suppress the probability of determining the incorrect result. Thus, BQP is the quantum analogue of P and plays a similar role in the classification of decision problems. Since a quantum computer can simulate any classical computation, P is contained in BQP.[5]

The most important example of a BQP problem that is not known to be in P is integer factoring. As a decision problem, this asks "Given N and M, does the integer N have a factor p with $1 < p \leq M$?" In the 90's, Peter Shor famously proved that factoring is in BQP by developing a quantum factoring algorithm. There is no proof that factoring is classically hard (outside of P) – nonetheless, many of the cryptography schemes on which society relies for secure communication over the internet are only secure if it is. Shor's algorithm renders all of these schemes useless if a large scale quantum computer is ever built.

The quantum analogue of NP is the class QMA, Quantum Merlin-Arthur, which is the class of decision problems whose Yes-instances can be efficiently checked by a quantum computer given a quantum state as a proof (or witness). The colorful name comes from the description of this class in terms of a game: Merlin, all-powerful but less than trustworthy, wishes to prove to Arthur, a fallible but well-intentioned individual who happens to have access to a quantum computer, that a particular instance of a problem is a Yes-instance. Merlin, using whatever supernatural powers he enjoys, provides Arthur with a quantum state designed to convince Arthur of this claim. Arthur then uses his quantum computer to decide, with some bounded error rate (say 1/3), whether to accept or reject the proof.

There are three primary differences between NP and QMA: 1) the verifier is a quantum computer, 2) the proof is a quantum state, and, 3) the verification is allowed a bounded error rate. The first two differences provide the class with its additional quantum power; that the verifier is allowed a bounded error rate is necessary due to quantum stochasticity, but not believed to be the source of its additional power. We note that the particular error bound is again somewhat arbitrary – Arthur can exponentially improve the accuracy of a noisy verification circuit by requesting Merlin provide him multiple copies of the proof state and running his verifier multiple times [3]. Thus, even a verifier which falsely accepts No-instances with probability up to $1/2-1/\text{poly}(N)$ while accepting valid proofs with probability 1/2 only slightly larger can be turned into an efficient bounded error QMA verifier through repetition.

An example of a QMA problem is given by the k-LOCAL HAMILTONIAN problem:

Input:: A quantum Hamiltonian $H = \sum_m A_m$ composed of M bounded operators, each acting on k qubits of an N qubit Hilbert space. Also, two energies $a < b$, separated by at worst a polynomially small gap $b-a > 1/\text{poly}(N)$.

Output:: Does H have an energy level below a?

Promise:: Either H has an energy level below a or all states have energies above b.

Here we have introduced the notion of a 'promise' in a decision problem. Promises are a new feature in our discussion: they impose a restriction on the instances that a questioner is allowed to present to a decision procedure. The restriction arises because the algorithms and verification procedures we use to treat promise problems need not be correct when presented with instances that do not satisfy the promise – an efficient solver for LOCAL HAMILTONIAN could in fact fail on Hamiltonians with ground state energies in the *promise gap* between a and b and we would still consider LOCAL HAMILTONIAN solved.

---

[4] This may of course be the bias of the scientists who don't want to be replaced by omniscient laptops.

[5] For the expert, we note that a closer analogue of BQP is BPP, the class of decision problems which can be efficiently decided by a randomized classical algorithm with bounded error. In an attempt to minimize the onslaught of three letter acronyms, we have left this complication out.

Heuristically, it is clear why we need the promise gap for LOCAL HAMILTONIAN to be QMA: suppose we had a quantum verifier which took a quantum state $|\psi\rangle$ and tried to measure its energy $\epsilon = \langle\psi|H|\psi\rangle$ through a procedure taking time T. Time-energy uncertainty suggests that we should not be able to resolve ϵ to better than $1/T$. Thus, if T is to be at most polynomially large in N, the verifier would not be able to determine whether an ϵ exponentially close to a is above or below the threshold.

The actual construction of a verification circuit for the LOCAL HAMILTONIAN problem is somewhat more subtle than simply 'measuring' the energy of a given state. As we will provide a very closely related construction for the QSAT problem below, we do not include the verifier for LOCAL HAMILTONIAN in these notes and instead refer the interested reader to Ref. 3.

The quantum analogue of the classical claim that P $\neq$ NP is that BQP $\neq$ QMA – a conjecture that is strongly believed for many of the same reasons as in the classical case.

E. NP-Completeness: Cook-Levin

In the early 1970s, Cook and Levin independently realized that there are NP problems whose solution captures the difficulty of the entire class NP. These are the so-called *NP-complete* problems. What does this mean?

A problem is NP-complete if it is both a) in NP (efficiently verifiable) and b) any problem in NP can be reduced to it efficiently. Thus, if we had an algorithm to solve an NP-complete problem efficiently, we could solve any problem whatsoever in NP efficiently. This would prove P = NP with all of the unexpected consequences this entails. Assuming on the contrary that P $\neq$ NP, any problem which is NP-complete must be intractable.

Let us sketch a proof of the Cook-Levin theorem showing the existence of NP-complete problems. In particular, we will show that classical 3-satisfiability, 3-SAT, is NP-complete. 3-SAT is the decision problem which asks whether a given Boolean expression composed of the conjunction of clauses, each involving at most 3 binary variables, has a satisfying assignment. Re-expressed as an optimization problem, 3-SAT asks, "Does the energy function

$$H = \sum_m E_m(\sigma_{m_1}, \sigma_{m_2}, \sigma_{m_3}), \tag{1}$$

acting on N binary variables σ_i in which each local energy term E_m takes values 0 or 1, have a zero energy (satisfying) ground state?"[6]

1. 3-SAT is in NP

First, it is clear that 3-SAT is itself efficiently verifiable and therefore in NP. If a prover wishes to prove that a particular instance H is satisfiable, she could provide a verifier a zero energy configuration. The verifier would take this configuration and evaluate its energy (using arithmetic in a polynomial number of steps) and thus be able to check the validity of the claim. If H is satisfiable, such a configuration exists. On the other hand, if H is not satisfiable, the prover would not be able to convince the verifier otherwise because all states would have energy greater than zero.

2. 3-SAT is NP-complete

The tricky part is to show that 3-SAT is as hard as the entire class NP. We need to show that *any* possible NP problem can be reduced to a 3-SAT problem by a polynomial transformation. What is the only thing that all NP problems have in common? By definition, they all have polynomial size verification procedures which take as input a proposed proof that an instance is a Yes-instance and output either Accept or Reject based on whether the proof is valid. This verification procedure is what we will use to provide the reduction to 3-SAT.

Let us think of the verification procedure for a particular instance A of some NP problem as a polynomially sized Boolean circuit as in Figure 1. The input wires encode the proposed proof that A is a Yes-instance and the output wire tells us whether to Accept or Reject the proof. The gates in the figure are simply the usual Boolean logic gates such as NAND and NOR, which take two input bits and provide one output bit. Any Boolean circuit may be written using such binary operations with arbitrary fan-out, so we assume that we can massage the verification circuit into

---

[6] The interactions E_m in 3-SAT are usually defined to penalize exactly one of the $2^3 = 8$ possible configurations of its input variables – but allow each of the $\binom{N}{3}$ possible 3-body interactions to appear multiple times in the sum. Thus, our definition is equivalent up to absorbing these terms together, which modifies the excited state spectrum but not the counting of zero energy satisfying states.

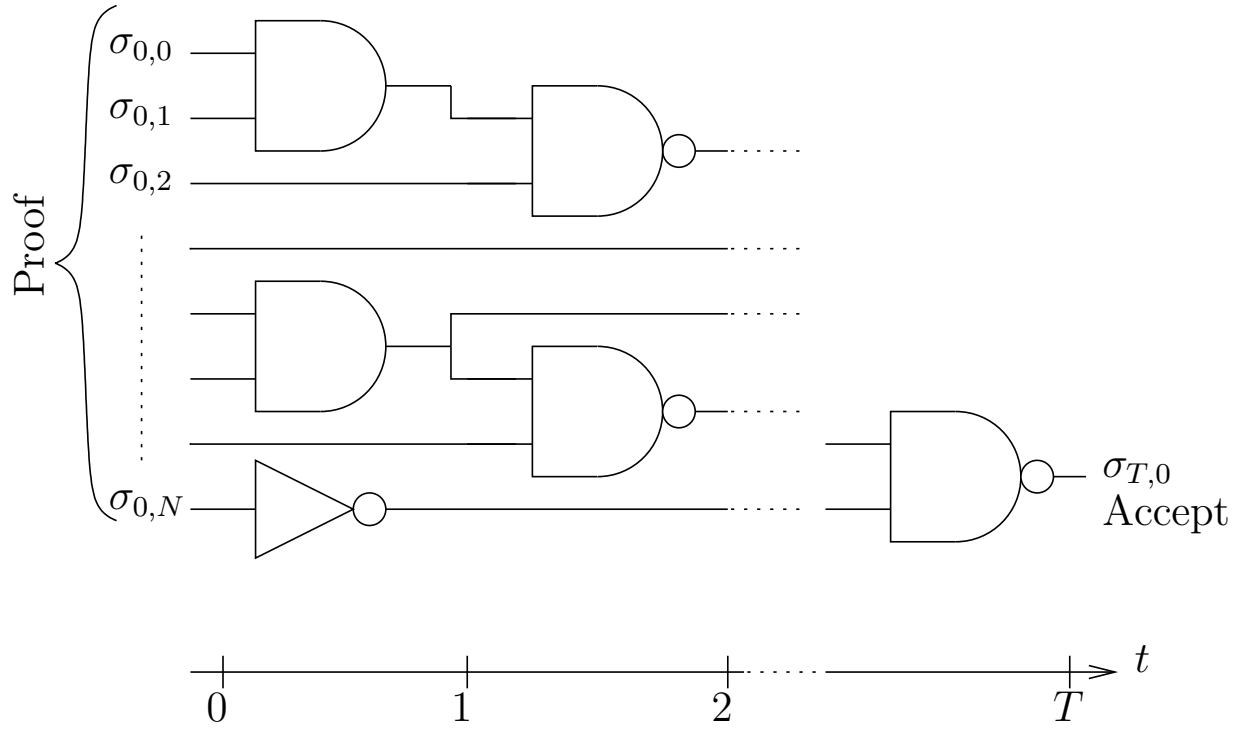
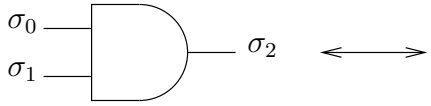


FIG. 1. Circuit representing an NP verifier. The circuit depends on the particular instance and must be efficiently constructible by a polynomial time circuit drawing algorithm.



σ_0	σ_1	σ_2	E
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

FIG. 2. Interpretation of Boolean AND gate as three-body interaction.

the form shown. Now we will construct an instance of 3-SAT encoding the operation of this circuit. That is, if the instance is satisfiable, then there exists a proof that the verifier accepts showing that the original NP problem is a Yes-instance and conversely, if the instance is not satisfiable, then no such proof exists and the original NP problem is a No-instance.

The 3-SAT instance is very simple to construct if we simply change our point of view on the picture in Figure 1. Instead of viewing it as a Boolean circuit operating from left to right, let us view it as the interaction graph for a collection of $O(N \times T)$ binary bond variables – one for each of the wires in the circuit: the input bits of the proof, the output bit and each of the intermediate variables. Each gate then specifies a 3-body interaction E_m for the adjacent variables which we define to take the value 0 for configurations in which the variables are consistent with the operation of the gate and 1 otherwise. See Figure 2. We now add a final 1-body term on the output bit of the verification circuit which penalizes the Reject output. We now have an Ising-like model with polynomially many 3-body interactions and non-negative energy.

That's it. If the 3-SAT instance described by Figure 1 has a zero energy ground state, then there is a configuration of the wire variables such that the circuit operates correctly and produces an Accept output. In this state, the input wires represent a valid proof showing that the original instance was a Yes-instance. On the other hand, if the 3-SAT instance is not satisfiable, no state exists such that the circuit operates correctly and the output produced is always

REJECT. Thus we have shown that all problems in NP can be efficiently reduced to 3-SAT.

Now that we have one problem, 3-SAT, which is NP-complete, it is straightforward to show the existence of many other NP-complete problems: we need only find reductions from 3-SAT to those problems. Indeed, a veritable menagerie of NP-complete problems exists (see *e.g.*, [18]) including such famous examples as the traveling salesman problem and graph coloring. A more physics oriented example is that of determining the ground state energy of the $\pm J$ Ising model in 3 or more dimensions [10].

F. QMA-Completeness: Kitaev

The complexity class QMA provides the quantum analogue to NP and, just like NP, it contains complete problems which capture the difficulty of the entire class. Kitaev first introduced the QMA-complete problem 5-LOCAL HAMILTONIAN in the early '00s and proved its completeness using a beautiful idea due to Feynman: that of the history state, a superposition over computational histories. The quantum Cook-Levin proofs are somewhat more complicated than the classical case and we will only sketch them here (see [12] for more details). For simplicity and to connect with the statistical study undertaken in the later sections, we restrict our attention to the slightly simpler problem of k-QSAT, which is QMA_1-complete for $k \geq 4$. QMA_1 is the variant of QMA in which the verification error is one-sided: Yes-instances may be verified with no errors while invalid proofs still occasionally get incorrectly accepted.

First, let us define k-QSAT a bit more carefully:

Input:: A quantum Hamiltonian $H = \sum_m \Pi_m$ composed of M projectors, each acting on at most k qubits of an N qubit Hilbert space.

Promise:: Either H has a zero energy state or all states have energy above a promise gap energy $\Delta > 1/\mathrm{poly}(N)$.

Question:: Does H have a zero energy ground state?

Now, we sketch the proof that QSAT is QMA_1-complete.

1. *QSAT is QMA_1*

To show that QSAT is QMA_1, we need to find an efficient quantum verification scheme such that (a) there exist proofs for Yes-instances which our verifier always accepts and (b) any proposed proof for a No-instance will be rejected with probability at least $\epsilon = 1/\mathrm{poly}(N)$. This rather weak requirement on the bare false-Acceptance rate can be bootstrapped into an arbitrarily accurate verification scheme by repetition, as sketched in Section II D above.

Given an instance $H = \sum_m \Pi_m$, the obvious proof is for Merlin to provide a state $|\Psi\rangle$ which he alleges is a zero energy state. Arthur's verification procedure will be to check this claim. The verifier works by measuring each of the Π in some pre-specified order on the state. That this can be done efficiently follows from the fact that Π acts on no more than k qubits and therefore its measurement can be encoded in an N-independent number of quantum gates. Clearly, if $|\Psi\rangle$ is a zero-energy state, it is a zero-energy eigenstate of each of the Π and therefore all of these measurements will produce 0 and the verifier accepts. This checks condition (a) above and we say our verification scheme is complete.[7]

On the other hand, if H is a No-instance, it has a ground state energy above the promise gap Δ and $|\Psi\rangle$ necessarily has overlap with the positive eigenspaces of at least some of the Π. It is a short computation to show that the probability that all of the measurements return 0 will then be bounded above by $1 - \Delta/N^k \sim 1 - 1/\mathrm{poly}(N)$. Thus, No-instances will be rejected with probability at least $\epsilon = 1/\mathrm{poly}(N)$ and our verification scheme is sound.

2. *QSAT is QMA_1-complete*

Just as in the classical Cook-Levin proof, we need to show that *any* QMA_1 problem can be reduced to solving an instance of QSAT. We again exploit the only thing that all QMA_1 problems have in common: their quantum verification algorithm. We will take the quantum circuit representing this verifier and construct from it a QSAT

[7] The feature that one can do these measurements by local operations and that they provide probability 1 verification of ground states is a special feature of the QSAT Hamiltonian which allows it somewhat to evade the heuristic expectations of time-energy uncertainty.

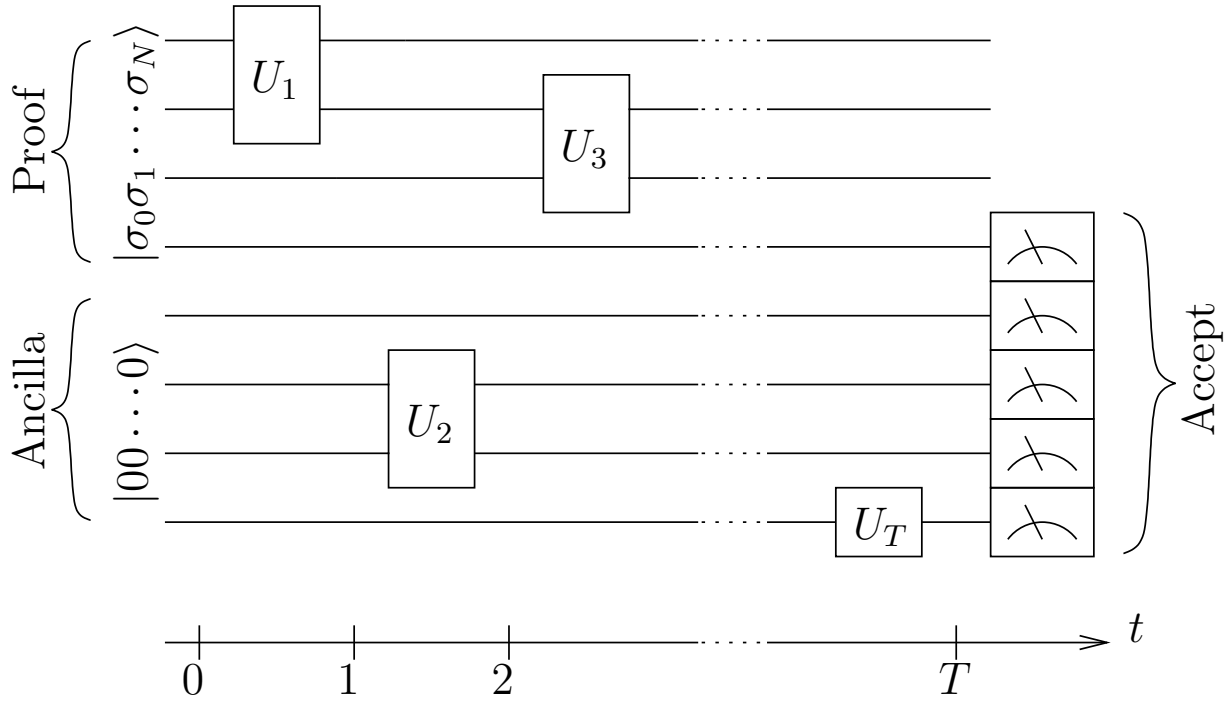


FIG. 3. QMA verification circuit. The circuit depends on the instance and must be constructible by an efficient algorithm given the instance. We have drawn the circuit so that there is a single local gate per time step, so $T = \text{poly}(N)$.

Hamiltonian whose ground state energy is directly related to the maximal acceptance probability of the verification circuit.

Let A be an arbitrary instance of a QMA_1 problem L. Then A has a polynomial sized quantum verification circuit as in Figure 3. This circuit takes as input a quantum state encoding a proof that A is a Yes-instance of L, along with some ancilla work qubits in a fiducial 0 state, then performs a sequence of T unitary one and two qubit gates and finally measures the output state of some subset of the qubits. If A is a Yes-instance, then there exists a valid input state such that all of the output bits will yield 0 with probability 1. Otherwise, at least one of the output bits will read 1 with probability polynomially bounded away from 0 for any input state $|\sigma_0 \cdots \sigma_N\rangle$.[8]

We now construct a single particle hopping Hamiltonian whose ground state encodes the operation of this quantum circuit. We introduce a *clock* particle hopping on a chain of length T and label its position states by $|t\rangle$. We endow this particle with an enormous 'spin': a full N qubit Hilbert space (dimension 2^N). As the particle hops, 'spin-orbit' coupling induces rotations on the N qubit space which correspond to the unitary gates in the verification circuit. To wit:

$$H_p = \frac{1}{2}\sum_{t=0}^{T-1}\Big(-|t+1\rangle\langle t|\otimes U_{t+1} - |t\rangle\langle t+1|\otimes U_{t+1}^\dagger + |t\rangle\langle t| + |t+1\rangle\langle t+1|\Big)$$

The terms of this Hamiltonian have been normalized and shifted such that each is a projector with energies 0 and 1, but otherwise it is just a 1-D hopping problem with Neumann boundary conditions. [9] Indeed, there is a simple basis transformation in which the spin-orbit coupling disappears entirely. This consists of rotating the basis of the position t spin space by a sequence of unitary transformations $U_1^\dagger U_2^\dagger \cdots U_t^\dagger$.

In this representation, we see that the 2^N spin components decouple and the system is really 2^N copies of the Neumann chain. Thus, the spectrum is $(1-\cos k)/2$, a cosine dispersion with bandwidth 1, ground state energy 0 at wave vector $k = 0$ and allowed $k = n\pi/(T+1)$.

---

[8] The observant reader will notice the addition ancillae qubits. These are necessary when computation is done by reversible gates, as in unitary circuit computation. We leave it as an exercise to figure out why the absence of ancillae would make the verification circuit unsound.

[9] In fact, Neumann boundary conditions (which stipulate the value of the derivative of the solution to a differential equation) apply to the problem obtained in the time continuum limit of the discrete hopping Hamiltonian under consideration.

The propagation Hamiltonian has (zero energy) ground states of the form (in the original basis):

$$|\psi\rangle = \frac{1}{\sqrt{T+1}} \sum_t |t\rangle \otimes U_t U_{t-1} \cdots U_1 |\xi\rangle \tag{2}$$

where $|\xi\rangle$ is an arbitrary 'input state' for the N qubit space. This state $|\psi\rangle$ is called the *history state* of the computation performed by the verification circuit given input $|\xi\rangle$. It is a sum over the state of the quantum computation at each step in the circuit evolution. Any correct computation corresponds to a zero energy history state – incorrect computations will have a non-zero overlap with higher energy hopping states.

Now, we have a Hamiltonian that encodes the correct operation of the verification circuit. We simply need to add terms that will penalize computations which do not correspond to appropriate input states and accepting output states. These terms affect the computational state at times $t = 0$ and T, so they are boundary fields from the point of view of the hopping problem. In general, they should split the 2^N degeneracy of the pure hopping in H_p, and since they are positive operators, lift the ground state energy.

The initialization term is simply a sum over the ancilla qubits of projectors penalizing $|\xi\rangle$ with incorrectly zeroed ancillae:

$$H_i = \sum_{j \in \text{Ancilla}} |0\rangle\langle 0|_t \otimes |1\rangle\langle 1|_j \tag{3}$$

Similarly, the output term penalizes output states which overlap $|1\rangle$ on the measured output bits:

$$H_o = \sum_{j \in \text{Accept}} |T\rangle\langle T|_t \otimes |1\rangle\langle 1|_j \tag{4}$$

Now we consider the full Hamiltonian

$$H = H_i + H_p + H_o \tag{5}$$

If $|\psi\rangle$ is a zero energy state of H, then it is a zero energy state of each of the three pieces. Hence, it will be a history state in which the input state $|\xi\rangle$ has appropriately zeroed ancillae and the output state has no overlap with $|1\rangle$ on the measured qubits – thus, $|\xi\rangle$ is a proof that the verifier accepts with probability 1 and the original instance A is a Yes-instance. Conversely, if such a proof state $|\xi\rangle$ exists then the history state built from it will have zero energy.

It is somewhat more work to show the soundness of the construction: A is a No-instance if and only if the Hamiltonian H has ground state energy bounded polynomially away from 0 [3]. The intuition is straightforward – the strength of the boundary fields in the basis transformed hopping problem for a given spin state corresponds to the acceptance probability of the associated input state. Since these repulsive fields lift Neumann conditions, they raise the ground state energy quadratically in 1/T – they effectively force the ground state wavefunction to bend on the scale of T. For a No-instance, since no spin sector is both valid and accepting, all states must gain this inverse quadratic energy.[10]

To be a bit more precise, we assume for contradiction that we have a state $|\psi\rangle$ with energy exponentially small in N (hence smaller than any polynomial in N or T):

$$\langle\psi|H|\psi\rangle = \langle\psi|H_i|\psi\rangle + \langle\psi|H_p|\psi\rangle + \langle\psi|H_o|\psi\rangle \leq O(e^{-N}) \tag{6}$$

Since each term is positive, each is bounded by the exponential. The hopping Hamiltonian H_p has a gap of order $1/T^2$ for chains of length T, thus if we decompose $|\psi\rangle$ into a zero energy piece (a history state) and an orthogonal complement,

$$|\psi\rangle = \frac{\sqrt{1-\alpha^2}}{\sqrt{T+1}} \sum_t |t\rangle \otimes U_t \cdots U_1 |\xi\rangle + \alpha |\text{Exc}\rangle \tag{7}$$

we must have exponentially small overlap onto the complement:

$$O(e^{-N}) > \langle\psi|H_p|\psi\rangle = \alpha^2 \langle\text{Exc}|H_p|\text{Exc}\rangle > \alpha^2 O(1/T^2) \tag{8}$$

---

[10] This is overly simplified: in the absence of the output term H_o, the gauge transformed problem can be thought of as 2^N decoupled hopping chains, some fraction of which have boundary fields at $t = 0$. The output term is not simply a field on these chains – it couples them and in principle allows hopping between them as a star of chains. The upshot is that the repulsive (diagonal) piece outweighs the off-diagonal mixing.

In other words,

$$|\psi\rangle = \frac{1}{\sqrt{T+1}} \sum_t |t\rangle U_t \cdots U_1 |\xi\rangle + O(e^{-N}) \tag{9}$$

The input term H_i has energy 0 for valid input states $|\xi^V\rangle$ and energy at least 1 for invalid states $|\xi^I\rangle$. Thus, decomposing $|\xi\rangle = \sqrt{1-\beta^2}|\xi^V\rangle + \beta|\xi^I\rangle$, we find (abusing notation and dropping explicit reference to the $|t=0\rangle$ sector on which H_i acts):

$$O(e^{-N}) > \langle\psi|H_i|\psi\rangle = \frac{\beta^2}{T+1}\langle\xi^I|H_i|\xi^I\rangle + O(e^{-N}) \geq \frac{\beta^2}{T+1} + O(e^{-N}) \tag{10}$$

In other words,

$$|\psi\rangle = \frac{1}{\sqrt{T+1}} \sum_t |t\rangle U_t \cdots U_1 |\xi^V\rangle + O(e^{-N}) \tag{11}$$

Finally, considering the output term we find:

$$\begin{aligned} \langle\psi|H_o|\psi\rangle &= \frac{1}{T+1}\langle\xi^V|U_1^\dagger \cdots U_t^\dagger H_o U_t \cdots U_1|\xi^V\rangle + O(e^{-N}) \\ &= \frac{p_r}{T+1} + O(e^{-N}) \end{aligned} \tag{12}$$

where p_r is the probability that the original QMA_1 verifier rejects the proposed proof $|\xi^V\rangle$ for the No-instance A. Since this rejection probability is bounded below by a constant, the state $|\Psi\rangle$ cannot possibly have exponentially small energy.

We have reduced the arbitrary QMA_1 instance A to asking about the zero energy states of a hopping Hamiltonian H constructed out of projectors. This is almost what we want. We have a Hamiltonian constructed out of a sum of projectors but they each act on three qubits tensored with a (large) particle hopping space rather than on a small collection of qubits.

The final step in the reduction to k-QSAT is to represent the single particle Hilbert space in terms of a single excitation space for a chain of clock qubits in such a way that we guarantee the single particle sector is described by H above and that it remains the low energy sector. We refer the interested reader to the literature for more details on these clock constructions. Each of the projectors of H becomes a joint projector on one or two of the computational (spin) qubits and some number of the clock qubits (two in [12]). The final 4-QSAT Hamiltonian will then be given by a sum of projectors involving at most 4 qubits

$$H = H_i + H_p + H_o + H_c \tag{13}$$

where H_c acts on the clock qubits to penalize states which have more than one clock particle. This concludes our brief overview of complexity theory. We next turn to a review of results obtained by applying ideas from (quantum) statistical mechanics to random ensembles of classical and quantum k-SAT.

III. PHYSICS FOR COMPLEXITY THEORY

There are two main ways for physicists to contribute to complexity theory. One is to bring to bear their methods to answer some of the questions posed by complexity theorists. Another is to introduce concepts from physics to ask new types of questions, thereby providing an alternative angle, permitting a broader view and new insights. This section is devoted to the illustration of this point, using the k-SAT problem introduced above as a case in point. In particular, we discuss both classical k-SAT and its quantum generalisation k-QSAT [12, 26].

A. Typical versus worst-case complexity

As explained above k-SAT is NP complete for $k \geq 3$. Thus, for any given algorithm, we expect that there are instances which will take an exponentially long time to solve. However, we ought not be too discouraged – some instances of k-SAT may be parametrically easier to solve than others, and these may be the ones of interest in a

given context. To make this more precise, it is useful to introduce the concept of typical, as opposed to worst-case, complexity.

In order to define typicality, one can consider an ensemble in which each problem instance is associated with a probability of actually occurring. Typical quantities are then given by stochastic statements, e.g. about a median time required for solving problem instances, which may differ substantially from the corresponding average, or indeed the worst-case, quantities when the latter have a sufficiently small weight in the ensemble. Precisely what quantities to calculate depends on the aspects of interest. For instance, a median run-time is not much affected by a small fraction of exponentially long runs, while these may dominate the expectation value of the run-time.

It is worth emphasizing again that the polynomial reductions discussed in Section II provide a characterization of the *worst case* difficulty of solving problems. The reductions and algorithms in this context must work for *all* instances of a problem. Reductions however may transform typical instances of A into rather 'atypical' instances of B. Whether a useful framework of reductions can be defined that preserve typicality is an open question (see Chapter 22 of Ref. [9]), but the study of typical instances of particular hard problems has itself been a fruitful activity, as we will discuss in the following.

B. Classical statistical mechanics of k-SAT

We now give an account of an analysis of such an ensemble for classical k-SAT. For completeness, let us begin with reviewing the original definition of classical k-SAT, expanding on the brief definition provided in Section II E. Indeed, the original computer science definition of satisfiability looks somewhat different from the Hamiltonian problem we introduced. Consider a set of N Boolean variables $\{x_i \mid i = 1 \dots N\}$, i.e. each variable x_i can take two values, true or false (in which cases the negation $\bar{\mathrm{x}}_i$ is false or true, respectively). Classical k-SAT asks the question, "Does the Boolean expression:

$$\mathcal{C} = \bigwedge_{m=1}^{\mathrm{M}} \mathrm{C}_m \tag{14}$$

evaluate to true for some assignment of the x_i?" Here, each clause is composed of a disjunction of k literals, e.g. for $k = 3$:

$$\mathrm{C}_m = \mathrm{x}_{m_1} \vee \bar{\mathrm{x}}_{m_2} \vee \mathrm{x}_{m_3} \tag{15}$$

where each variable occurs either affirmed (x_{m_1}) or negated ($\bar{\mathrm{x}}_{m_2}$). Hence, there are 2^k possible clauses for a given k-tuplet $\{\mathrm{x}_{m_j} \mid j = 1 \dots k\}$.[11]

This definition is equivalent to the definition in terms of the k-body interacting spin Hamiltonian of Eq. (1). To obtain a spin Hamiltonian from the collection of clauses, Eq. (14), we convert the Boolean variables x_i into Ising spins $\sigma_i = \pm 1$, with $\sigma_i = +1(-1)$ representing x_i being true (false). A clause then becomes a k-spin interaction designed such that the satisfying assignments evaluate to energy 0, and the forbidden assignment to energy 1. For instance, the clause given in Eq. (15) turns into:

$$H_m = 2^{-3} \left(1 - \sigma_{i_{m_1}}\right) \left(1 + \sigma_{i_{m_2}}\right) \left(1 - \sigma_{i_{m_3}}\right) \quad . \tag{16}$$

(It is this formulation of classical k-SAT that will lend itself naturally to a quantum generalisation, which we describe below.)

The k-SAT ensemble is now random in two ways:

(R1): each k-tuple occurs in H with probability $p = \alpha N \Big/ \binom{N}{k}$

(R2): each k-tuple occurring in H is randomly assigned one of the 2^k possible clauses.

Here, we have introduced a parameter α for the number of clauses, $\mathrm{M} = \alpha N$, which is proportional to the number of variables[12] because there are $\binom{N}{k}$ possible k-tuples. The 'interactions' can be pictorially represented by an interaction graph, Fig. 4. This is a bipartite graph, one sublattice of which has N nodes, denoted by circles representing the x_i,

---

[11] The symbols $\wedge$ and $\vee$ are the Boolean operators 'and' and 'or'.

[12] Actually, only the expectation value of M equals αN. The Poissonian distribution for M of course has vanishing relative fluctuations $(\langle \mathrm{M}^2 \rangle - \langle \mathrm{M} \rangle^2)/\langle M \rangle^2$ as $N \to \infty$.

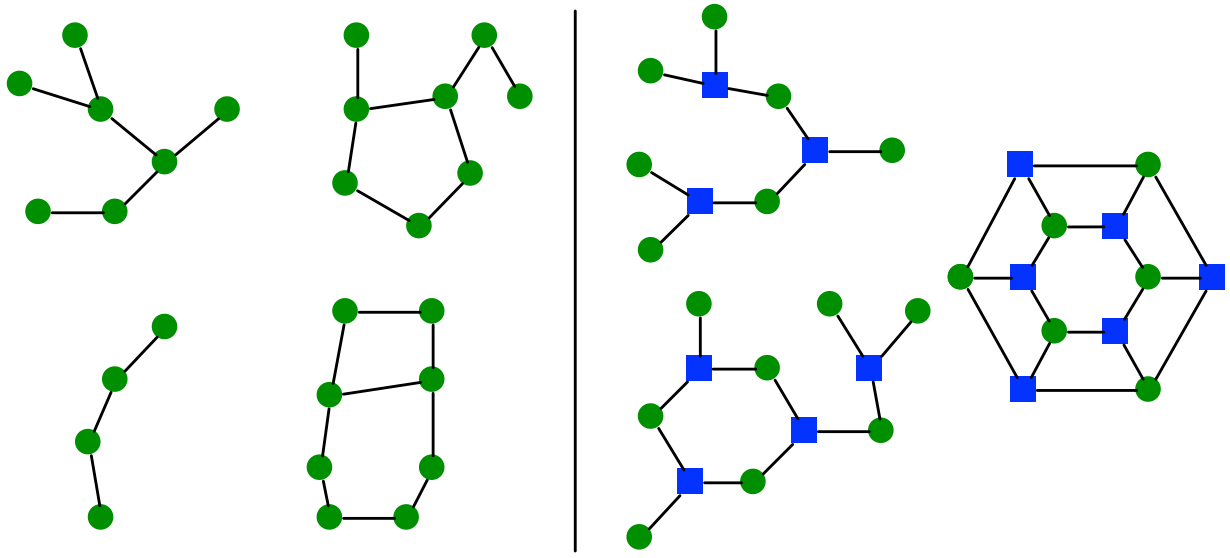


FIG. 4. Examples of random interaction graphs for (a) 2-SAT and (b) 3-SAT, respectively. The (green) circles represent qubits. (a) The clusters, clockwise from bottom left, are chain, tree, clusters with one and two closed loops ("figure eight"). The short closed loops, as well as the planarity of the graphs, are not representative of the large-N limit. (b) Each (blue) square represents a clause connected to 3 nodes. Clockwise from top left are a tree, a graph with nontrivial core and a graph with simple loops but no core.

and the M nodes of the other sublattice denoted by triangles represent the clauses C_m. Each triangle is connected to the k variables participating in the clause it represents, whereas each variable is connected to all clauses it participates in, which implies an average coordination of αk, with a Poissonian distribution. The random graph thus constructed contains all the information on a given problem instance if we label each triangle with which of the 2^k possible clauses it represents. This graph will be used for random quantum k-SAT as well, where the Boolean variables and clauses will be replaced by appropriate quantum generalisations.

C. Schematic phase diagram of classical random k-SAT

In Fig. 5, we show a schematic phase diagram for random k-SAT. The first question one might ask is: is there a well-defined phase transition, at some value $\alpha = \alpha_s(k)$, such that instances for $\alpha < \alpha_s$ are satisfiable, and those for $\alpha > \alpha_s$ are not? It has been shown that there exists such a transition for the random ensemble. This does not mean that *all* instances with $\alpha < \alpha_s$ are satisfiable: given an UNSAT instance with N sites and αN clauses, one could simply add N disconnected sites to get a new UNSAT instance with $\alpha' = \alpha/2$. What is true instead is that the probability of having such an UNSAT graph with $\alpha' < \alpha_s$ is exponentially small in N, so that for $N \to \infty$, such graphs do not arise with a probability approaching 1.

It is easy to provide a very rough estimate for where this happens, by adapting an idea of Pauling's which, amusingly, was devised for estimating the configurational entropy of the protons in water ice. We consider the clauses as constraints, each of which 'decimates' the number of allowed configurations by a factor $(1-2^{-k})$: only 1 out of the 2^k possible configurations of variables of any given clause is ruled out. For $\mathrm{M} = \alpha N$ such constraints, one is left with $2^N(1-2^{-k})^{\alpha N}$ solutions. In the thermodynamic limit, this number vanishes for $\alpha > \alpha_{wb} = -1/\log_2(1-2^{-k}) \sim 2^k \log 2$. In the k-SAT literature, this is known as the 'first-moment bound', for which there is a straightforward rigorous derivation. To find rigorous *upper bounds* one should instead employ different techniques, with inequalities coming from the analysis of the *second moment* of the number of solutions (after appropriately restricting the ensemble to reduce the fluctuations) [15]. It is interesting here to note that for large k the upper bounds and lower bounds converge, becoming a prediction for the actual location of the threshold.

The SAT-UNSAT transition is not the only transition of this problem, though. As indicated in Fig. 5, statistical mechanical methods imported from the study of spin glasses have been used to establish finer structure in the SAT phase. This plot shows a set of cartoons of configuration space, indicating the location of satisfying assignments. For N variables, configuration space is an N-dimensional hypercube and this plot indicates, in a two-dimensional 'projection', how 'close' satisfying assignments are to each other. Roughly, two solutions belong to the same cluster

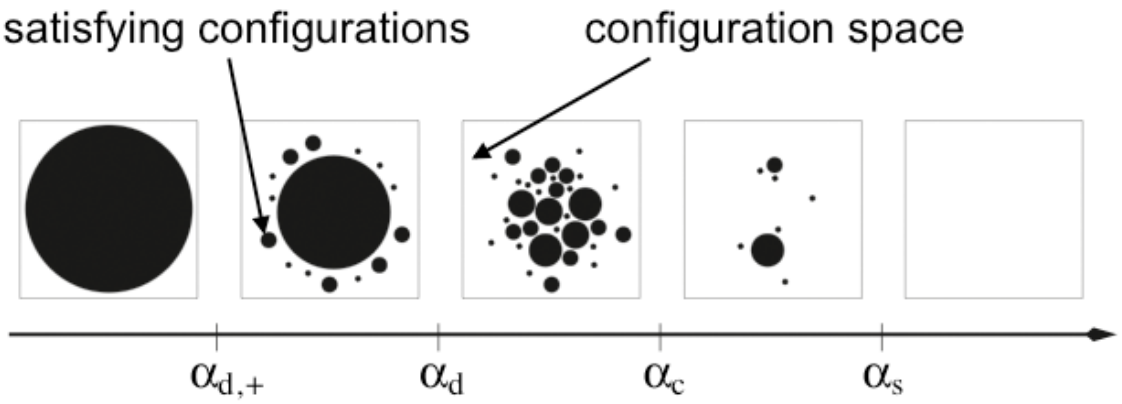


FIG. 5. Schematic phase diagram for random classical k-SAT ($k \geq 4$) [22]. Actually, configuration space is very high-dimensional (an N-dimensional hypercube), and the cartoons are only suggestive of the actual structure of the space of solutions, for the real complexity of which our everyday intuition from low dimensions may be quite inadequate.

if they can be reached via a sequence of satisfying configurations such that two consecutive ones differ by $O(N^\beta)$ variables with $\beta < 1$ [22].

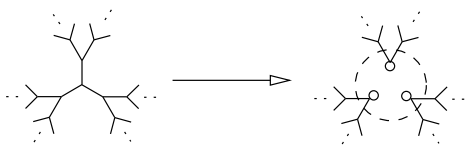
Figure 5 thus documents a set of transitions in the clustering of satisfying assignments. For the smallest α, all solutions belong to one single giant cluster – the full hypercube for $\alpha = 0$ – and then there is a successive break-up into smaller, and increasingly numerous clusters as α grows [22].

This structure of configuration space should have ramifications for how hard it is to solve the corresponding instances: small-scale clusters indicate a rugged energy landscape, with numerous local minima providing opportunities for search algorithms to get stuck. Indeed, all known algorithms slow down near α_s. That said, many simple approaches to random 3-SAT problems actually do quite well even in the clustered phases and the detailed relationship between clustering in configuration space and algorithmic difficulty is subtle, somewhat detail dependent and an ongoing research topic. It is particularly worth noting that even the simplest random greedy algorithms typically work across most of these transitions, at least for k-SAT. Indeed, while the identification of distinct phases has grown to give a phase diagram replete with fine structure, the portion of the phase diagram containing 'hard' instances – those for which deciding satisfiability takes exponentially long typically – has by now been pushed back to a tiny sliver at α_s of width $\delta\alpha < \alpha_s/100$. In the meantime, however, the action has started to shift to other problem ensembles which at the time of writing have proven more robustly difficult.

The derivation of this phase diagram was obtained using methods imported from the study of spin glasses, in particular the *cavity method* [20, 22]. The insights thus gained have lead to the development of an impressive arsenal of techniques for not only determining whether or not a k-SAT problem instance is soluble, but also for actually finding solutions in the form of satisfying assignments [11, 30, 32, 34]. In the following section, we provide a brief introduction to cavity analysis.

D. Cavity analysis

The cavity method is a cluster of techniques and heuristics for solving statistical models on sparse, tree-like interaction graphs G. In this approach, one determines the behavior of the model on G by first analyzing the restriction of the model to so-called *cavity graphs*. A cavity graph $G\backslash\{i\}$ is formed by carving site i out of G:



The neighbors of i, ∂i, which now sit at the boundary of the cavity in $G\backslash\{i\}$, are called *cavity spins*. The central assumption of the cavity method is that cavity spins are statistically independent in the absence of site i because they sit in disconnected components of $G\backslash\{i\}$. This assumptions massively simplifies the evaluation of observables in such models, ultimately leading to efficient procedures for finding ground states, evaluating correlation functions and determining thermodynamic free energies, phase diagrams, and clustering phenomena in models with quenched disorder.

The ensemble of interaction graphs G that arise from the rule (R1) have loops and thus do not fall into disconnected pieces when a cavity site i is removed. Nonetheless, for large N, any finite neighborhood of a randomly chosen point in G is a tree with high probability. That is to say, G does not contain short closed loops and we call it *locally tree-like*. For such G, we can hope that the cavity assumption will hold at least to a good approximation.

That neighborhoods in G are trees can be seen as follows: the subgraph consisting of site i and its neighbors has on average $n_1 = (1 + \alpha k)$ out of the N sites. The αk neighbors will in turn have αk further neighbors, so that the subgraph containing those as well has approximately $n_2 = 1 + (\alpha k) + (\alpha k)^2$ sites. Up to the γ-th nearest neighbors, the resulting subgraph grows exponentially, $n_\gamma \sim (\alpha k)^\gamma$. Obviously, $n_\gamma \leq N$, so that closed loops must appear at length $\gamma_c = (\ln N)/\ln(\alpha k)$. For $\gamma < \gamma_c$, $n_\gamma \ll N$ due to the exponential growth of n_γ, so that the randomly chosen interaction partners are overwhelmingly likely to be drawn from the sites not yet included in the neighborhood. Thus, the length of loops on G deverges with N, although excruciatingly (logarithmically) slowly.

Let us make these considerations more precise. Consider carving a cavity into a large regular random graph G with N spins and M edges representing two body interactions, as in Fig. 6. We will not explicitly consider the straightforward generalization to k-body interactions with $k > 2$; it needlessly complicates the notation. The statistical connection at temperature $1/\beta$[13] between the removed spin σ_0 and the rest of the graph is entirely mediated by the *joint* cavity distribution

$$\psi_{G\backslash\{0\}}(\sigma_1, \sigma_2, \sigma_3) = \frac{1}{Z_{G\backslash\{0\}}} \prod_{j \notin \{0,1,2,3\}} \sum_{\sigma_j} e^{-\beta H_{G\backslash\{0\}}} \tag{17}$$

That is, the thermal distribution for σ_0 in the original model is given by:

$$\psi_0(\sigma_0) = \frac{1}{Z_0} \sum_{\sigma_1,\sigma_2,\sigma_3} e^{-\beta(H_{01}+H_{02}+H_{03})} \psi_{G\backslash\{0\}}(\sigma_1, \sigma_2, \sigma_3) \tag{18}$$

If, on carving out the cavity, the neighboring spins become independent then the joint cavity distribution factors:

$$\psi_{G\backslash\{0\}}(\sigma_1, \sigma_2, \sigma_3) = \psi_{1\to 0}(\sigma_1)\psi_{2\to 0}(\sigma_2)\psi_{3\to 0}(\sigma_3) \tag{19}$$

On trees, the independence is exact because the cavity spins sit in disconnected clusters; on locally tree-like graphs, the cavity spins are only connected through long (divergent in system size) paths and thus we might expect Eq. (19) to hold approximately.

Thus the objects of interest are the $2M$ cavity distributions $\psi_{i\to j}$, see Fig. 6. These are also known as messages or beliefs: $\psi_{i\to j}(\sigma_i)$ is a message passed from site i to site j which indicates site i's beliefs about what it should do in the absence of site j, and thus also its beliefs about what site j should do to optimize the free energy. Crucially, when the cavity distributions are independent, they also satisfy the iteration relation:

$$\psi_{i\to j}(\sigma_i) = \frac{1}{Z_{i\to j}} \prod_{k \in \partial i\backslash\{j\}} \sum_{\sigma_k} e^{-\beta H_{ik}(\sigma_i,\sigma_k)} \psi_{k\to j}(\sigma_k) \tag{20}$$

These are the Belief Propagation (BP) equations and they are really just the Bethe-Peierls self-consistency equations in more formal, generalized, notation. Indeed, if we parameterize the functions $\psi_{i\to j}$ by cavity fields $h_{i\to j}$

$$\psi_{i\to j}(\sigma_i) = \frac{1}{2\cosh(\beta h_{i\to j})} e^{-\beta h_{i\to j}\sigma_i}, \tag{21}$$

and specialize to an Ising Hamiltonian $H = \sum_{\langle ij\rangle} J_{ij}\sigma_i\sigma_j$, then the BP equation becomes:

$$h_{i\to j} = \frac{1}{\beta} \sum_{k \in \partial i\backslash\{j\}} \tanh^{-1}\left[\tanh(\beta J_{ki}) \tanh(\beta h_{k\to i})\right] \tag{22}$$

---

[13] We may eventually take the temperature to 0 ($\beta \to \infty$) in order to find actual ground state solutions of the optimization problem.

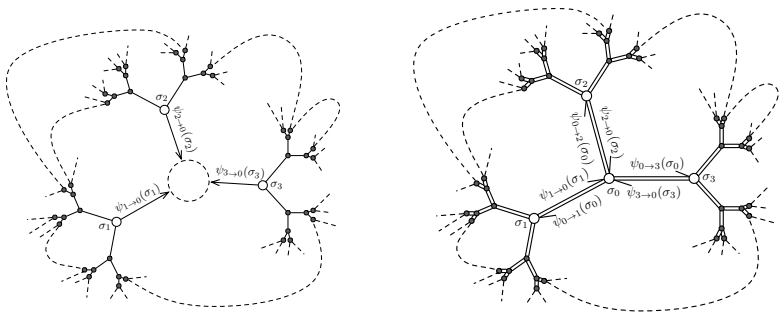


FIG. 6. *(Left)* Schematic of $q = 3$ regular random graph with a cavity carved out at spin σ_0. *(Right)* The belief propagation equations for a graph G involve $2M$ cavity distributions, or *beliefs*, $\psi_{i\to j}$, one for each of the two directions of a link in the graph.

which should be familiar from Ising mean field theory.

There are linearly many BP equations, each of which is a simple relation involving a finite summing out procedure on the right to define a cavity distribution on the left. We may now consider two approaches to solving them: a) take a thermodynamic limit and find the statistics of their solutions; and b) iteratively solve them for finite N using a computer. The former approach leads to the so-called cavity equations and the estimates regarding the thermodynamic phase diagram of Fig. 5. The latter leads to the belief propagation algorithm for solving particular instances of optimization problems. Indeed, once the solution of the BP equations is known for a particular instance, one can obtain a solution of the SAT formula by using a decimation heuristic which fixes the variables with positive (resp. negative) total of incoming messages to 1 (resp. 0) and re-running BP on the remaining formula if necessary [11].

There are as many BP equations as unknown cavity distributions and thus we generically expect to find discrete solutions. However, there can be more than one solution. For instance in the low temperature phase of the Ising ferromagnet, there are three: the (unstable) paramagnet and the two (symmetry related) magnetized solutions. For spin glass models with quenched disorder, there may be exponentially many solutions, each corresponding to a macroscopically distinct magnetization pattern in the system. When this occurs, the belief propagation algorithm may fail to converge. In this case, one needs to take into account the presence of multiple solutions of the BP equations, which can be done statistically using an algorithm known as Survey Propagation and in the thermodynamic limit using the 'replica symmetry breaking' cavity equations, which arise as a hierarchy of distributional equations. These equations describe the statistics of solutions of the BP equations. Although these analyses are important for a correct understanding of many types of glassy optimization problems on tree-like graphs [31], we will not consider these technical generalizations further. We note that the jargon of 'replica symmetry breaking' arises in a completely different approach to solving mean field glasses based on the so-called replica trick and Parisi ansatz. The terms have no intrinsic meaning in the context of the cavity approach.

For quantum stoquastic (or Frobenius-Perron) Hamiltonians – those for which a basis is known for which all off-diagonal matrix elements are negative, which guarantees that there exists a ground state wavefunction for which all components can be chosen to have a positive amplitude – BP has been generalized in some recent works [21, 28, 29] and has since been taken up in the study of a number of quantum models on trees [14, 23, 27, 37]. Spin-glasses with a transverse field are a case in point. This might be relevant for the study of the performance of some quantum adiabatic algorithms for solving classical instances of k-SAT, which we turn to next.

E. Adiabatic quantum algorithm for classical k-SAT

Before we move on to the problem of quantum satisfiability k-QSAT, let us first ask whether we could use a quantum algorithm to solve classical k-SAT efficiently. One possible strategy for this is based on a protocol known as adiabatic quantum computing [17], which in its full generality is equivalent to computation based on circuits. Here we will discuss a particularly simple member of this class of algorithms.

Consider a time-dependent quantum Hamiltonian, with real time t parametried by $s(t)$ (with $0 \le s \le 1$):

$$H(s) = (1-s)H_\Gamma + sH_0 \quad , \tag{23}$$

where H_Γ is a transverse-field term and H_0 is obtained from Eq. 1

$$\begin{aligned} H_\Gamma &= -\Gamma \sum \sigma_i^{\mathrm{x}} \\ H_0 &= \sum_m E_m \left(\{\sigma_i^z\}\right) \quad , \end{aligned} \tag{24}$$

by replacing the σ_i's by Pauli operators σ_i^z.

The ground state space of $H(s)$ at time $s = 1$ is spanned by (one of) lowest-energy classical configurations (ground states) of the k-SAT problem; it is these which we are after, but which can be hard to find. By contrast, at time $s = 0$, the quantum ground state has all spins polarized in the x-direction. This is both easy to describe and to prepare. If we start the system in its ground state at $s = 0$, and change $H(s)$ sufficiently slowly, the state of the system will evolve adiabatically, and reach the desired state at time $s = 1$.

However, we do not want to change $H(s)$ arbitrarily slowly, as this would be no gain over an exponentially long classical run-time.

What is it that limits the sweep-rate? A non-zero sweep rate can induce transitions to excited states. Careful derivation of the adiabatic theorem reveals that the probability of nonadiabatic transitions tends to zero so long as the sweep rate is slower than the minimal adiabatic gap Δ squared[14]. That is, the run time T must be greater than or of order $O(1/\Delta^2)$ in order to ensure adiabaticity.

Heuristically, we need to be concerned about avoided level crossings in the course of the evolution and in particular, the avoided crossing at the location of the minimal gap Δ.[15] As two levels approach closely, we get an effective two-level problem:

$$H_2 = \begin{pmatrix} \alpha(t - t_0) & \Delta/2 \\ \Delta/2 & -\alpha(t - t_0) \end{pmatrix} \quad . \tag{25}$$

At the closest approach, at $t = t_0$, the ground state is separated from the excited state by a gap Δ. For $|\alpha(t-t_0)| \gg \Delta$, variation of t has very little effect on the adiabatic eigenstates and the Schrödinger evolution remains adiabatic even for fast sweeping. It is only the time spent in the interaction region $|\alpha(t - t_0)| < \Delta$ where the adiabatic states rotate significantly and nonadiabatic transitions may arise. Thus, the interaction time is $t_I \sim \Delta/\alpha$ and the dimensionless figure of merit for adiabatic behavior should be $\Delta \cdot t_I \sim \Delta^2/\alpha$. In particular, for low sweep rates $\alpha \ll \Delta^2$ or long run times $T \geq O(1/\Delta^2)$, we expect to have purely adiabatic evolution. We note that the nonadiabatic transition probability P for this two-level model was calculated some eighty years ago by Landau [24] and Zener[41] whose exact result:

$$P = 1 - e^{-\pi\Delta^2/4\hbar\alpha} \tag{26}$$

quantifies the physical intuition in this case.

The quantum adiabatic algorithm has been studied extensively since its introduction ten years ago on a number of hard random constraint satisfaction problems closely related to 3-SAT [17, 38]. The critical question is simple: how does the typical minimal gap encountered during the procedure scale with increasing instance size N? Analytic work on simple (classically easy) problem ensembles found several polynomial time quantum adiabatic algorithms. Moreover, early numerical studies for very small instances of harder problems held promise that the gap would scale only polynomially [17, 39]. Unfortunately, subsequent numerical studies on larger systems indicate that the gap eventually becomes exponentially small due to a first order transition between a quantum paramagnet and a glassy state [40]. Even worse, recent perturbative work argues that many-body localization leads to a plethora of exponentially narrow avoided crossings throughout the glassy phase [5, 6] and thus the algorithm discussed here does not produce an efficient solution of random 3-SAT or related random constraint satisfaction problems. We note that this is consistent with other evidence that quantum computers will not enable the efficient solution of NP-complete problems.

IV. STATISTICAL MECHANICS OF RANDOM k-QSAT

Let us now turn to the study of random instances of quantum satisfiability k-QSAT. As discussed in Section II F, k-QSAT is QMA_1-complete and thus should be generally intractable. As in the classical case, one might hope to gain

---

[14] In fact, there is some controversy in the rigorous literature about whether the asymptotic sweep rate must be slower than $1/\Delta^2$ or $1/\Delta^{2+\delta}$ for some arbitrarily small constant δ. See [4].

[15] In the absence of any symmetries and fine-tuning, all level crossings are avoided as a function of a single adiabatic parameter s.

some insight into the nature of the difficulty of the quantum problem by studying a random ensemble of its instances. Moreover, the richness of phenomena exhibited by the classical random satisfiability problem – and the many important spin-off techniques that have been developed in their study – encourages us to seek analogous behaviors hiding in the quantum system.

A. Random k-QSAT ensemble

Let us recap the definition of k-QSAT from Section II F:

Input:: A quantum Hamiltonian $H = \sum_m \Pi_m$ composed of M projectors, each acting on at most k qubits of an N qubit Hilbert space.

Promise:: Either H has a zero energy state or all states have energy above a promise gap $\Delta > 1/\text{poly}(N)$.

Question:: Does H have a zero energy ground state?

Quantum satisfiability is a natural generalization of the classical satisfiability problem: bits become qubits and k-body clauses become k-body projectors. In key contrast to the classical case, where the binary variables and clauses take on discrete values, their quantum generalizations are continuous: the states of a qubit live in Hilbert space, which allows for linear combinations of $|0\rangle$ and $|1\rangle$.

Thinking of a Boolean clause as forbidding one out of 2^k configurations leads to its quantum generalization as a projector $\Pi_\phi^I \equiv |\phi\rangle\langle\phi|$, which penalizes any overlap of a state $|\psi\rangle$ of the k qubits in set I with a state $|\phi\rangle$ in their 2^k dimensional Hilbert space. Indeed, if we restrict the Π_m to project onto computational basis states, k-QSAT reduces back to k-SAT – all energy terms can be written as discrete 0 or 1 functions of the basis state labels and the promise gap is automatically satisfied since all energies are integers.

As in the classical problem, we make two random choices in order to specify an instance:

(R1): each k-tuple occurs in H with probability $p = \alpha N \Big/ \binom{N}{k}$

(R2): each k-tuple occurring in H is assigned a projector $\Pi_m = |\phi\rangle\langle\phi|$, uniformly chosen from the space of projectors of rank r. For these notes, we will mostly consider the case $r = 1$, although higher rank ensembles can be studied [36].

The first rule is identical to that of the classical random ensemble and thus the geometry of the interaction graphs (Figure 4) is the same – locally tree-like with long loops for sufficiently high clause density α.

The second rule, however, rather dramatically changes the nature of our random ensemble – the measure on instances is now continuous rather than discrete. This turns out to be a major simplification for much of the analysis: *Generic* choices of projectors reduce quantum satisfiability to a graph, rather than Hamiltonian, property. This "geometrization" property allows us to make strong statements about the quantum satisfiability of Hamiltonians associated with both random and non-random graphs and even non-generic choices of projector, both analytically and numerically. For the remainder of these notes, we use the term *generic* to refer to the continuous choice of projectors and *random* to refer to the choice of the graph. See Section IV C below for a more detailed discussion of geometrization.

B. Phase diagram

The first step in understanding the random ensemble is to compute the statistics of this decision problem as a function of α. Specifically we would like to know if there are phase transitions in the satisfying manifold as α is varied: these include both the basic SAT-UNSAT transition as well as any transitions reflecting changes in the structure of the satisfying state manifold. Additionally, we would like to check that the statistics in the large N limit are dominated by instances that automatically satisfy the promise gap.

The current state-of-the-art QSAT phase diagram is shown in Figure 7. Let us walk through a few of the features indicated. First, we have separated the $k = 2$ case from the higher connectivity cases because it is significantly simpler. For $k = 2$, we can solve the satisfiability phase diagram rigorously and even estimate energy exponents for the (non-zero) ground state energy above the satisfiability transition. Our ability to do so is consistent with the fact that 2-QSAT is in P – instances of 2-QSAT can be efficiently decided by *classical* computers! In particular, it can be shown that the zero energy subspace can be spanned, if it is nontrivial, by *product* states. Since these are much simpler to specify classically (a product state needs only $2N$ complex numbers instead of 2^N), it is perhaps

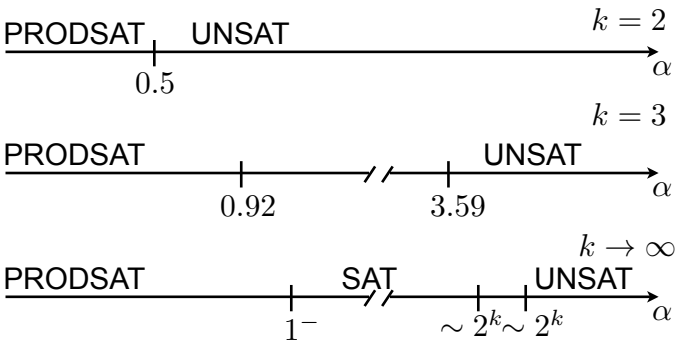


FIG. 7. Phase diagram of k-QSAT.

not surprising that we can decide whether or not such states exist that satisfy a given instance H. In any event, the only significant feature of the phase diagram is that for $\alpha < \alpha_s = 1/2$, we have a PRODSAT phase – that is a phase which is satisfiable by unentangled product states – and for $\alpha > \alpha_s$, we have an UNSAT phase with finite ground state energy density. The transition coincides with a geometric transition: $\alpha_s = 1/2$ corresponds to the emergence of a giant component in the underlying interaction graph.

For $k \geq 3$, the phase diagram is somewhat more interesting. Again, at low $\alpha < \alpha_{ps}(k) \sim 1$ there exists a PRODSAT regime in which satisfying product states are guaranteed to exist. Above α_{ps}, there are no satisfying product states, but the system remains SAT – thus, there is an "entanglement" transition in the ground state space as a function of α. Finally, above some $\alpha_c \sim 2^k$, there is an UNSAT phase in which it can be shown that there are no zero energy satisfying states. We note that the emergence of a giant component in the underlying interaction graph happens at $\alpha_{gc} = \frac{1}{k(k-1)} \ll \alpha_{ps} \ll \alpha_c$ – the various relevant transitions are well-separated at large k. A variety of different techniques go in to showing the existence of these transitions and phases – we sketch a few of these arguments in Section IV D.

C. Geometrization theorem

One of the most useful tools for studying random quantum satisfiability is geometrization. That is, the satisfiability of a generic instance of QSAT is a purely geometric property of the underlying interaction graph. This point of view extends to many properties of generic instances of QSAT – such as whether they are product satisfiable or not. Results about generic QSAT thus follow from identifying the right geometric properties in the interaction graph ensemble. We discuss a few of these examples in the sections to follow. Here, we provide an elementary proof of geometrization and a few immediate corollaries.

Geometrization Theorem 1. *Given an instance H of random k-QSAT with interaction graph G, the degeneracy of zero energy states $R(H) = \dim(\ker(H))$ takes a particular value R_G for almost all choices of clause projectors. R_G is minimal with respect to the choice of projectors.*

Proof. For a fixed interaction graph G with M clauses, $H = H_\phi = \sum_{i=1}^M \Pi_i = \sum_{i=1}^M |\phi_i\rangle\langle\phi_i|$ is a matrix valued function of the $2^k M$ components of the set of M vectors $|\phi_i\rangle$. In particular, its entries are polynomials in those components. Choose $|\phi\rangle$ such that H has maximal rank D. Then there exists an $D \times D$ submatrix of H such that $\det(H|_{D\times D})$ is nonzero. But this submatrix determinant is a polynomial in the components of $|\phi\rangle$ and therefore is only zero on a submanifold of the $|\phi\rangle$ of codimension at least 1. Hence, generically H has rank D and the degeneracy $R_G = \dim(\ker(H)) = 2^N - D$. □

The theorem holds for general rank r problems as well by a simple modification of the argument to allow extra ϕ's to be associated to each edge.

A nice corollary of this result is an upper bound on the size of the SAT phase at any k. Consider any assignment of classical clauses on a given interaction graph: this is a special instance of k-QSAT where the projectors are all diagonal in the computational basis. As this is a non-generic choice of projectors, the dimension of its satisfying manifold is an

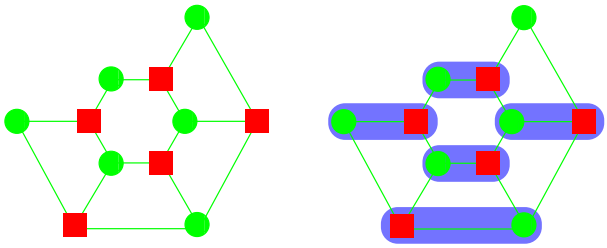


FIG. 8. Example of a $k = 3$ interaction graph with $M < N$, circles (green) indicate qubits and squares (red) indicate clause projectors that act on adjacent qubits (left); a dimer (blue shaded) covering that covers all clauses (right).

upper bound on the dimension for generic choices. We conclude then that the classical UNSAT threshold is an upper bound on the quantum threshold. Indeed, if we can identify the most frustrated assignment of classical clauses on a given interaction graph, i.e. the assignment that minimizes the number of satisfying assignments, we could derive an even tighter bound.

Corollary 1. *The generic zero state degeneracy is bounded above by the number of satisfying assignments of the most constrained classical k-SAT problem on the same graph.*

It is easy to construct example instances in which the quantum problem has fewer ground states than the most frustrated classical problem on the same interaction graph. Thus, the bound of corollary 1 is not tight.

D. A few details of phases and transitions

There are three flavors of arguments that have been used to pin down the phase diagram of $k \geq 3$-QSAT: construction of satisfying product states [25, 26]; combinatorial upper bounds on the zero state degeneracy[13, 26]; and, a non-constructive invocation of the quantum Lovasz local lemma to establish the entangled SAT phase[7]. All three ultimately rely on establishing a correspondence between some geometric feature of the interaction graph G (or its subgraphs) and the properties of zero energy states for generic instances through geometrization. We sketch each of the three kinds of results below and refer the motivated reader to the relevant literature for details.

1. Product states

Perhaps the most direct approach to establishing a SAT phase in the phase diagram is to attempt explicitly to construct satisfying product states [26]. This is sufficient completely to determine the $k = 2$ SAT phase, but only proves the existence of the PRODSAT phase of the $k \geq 3$ phase diagram.

In this approach, one constructs product states by a "transfer matrix"-like procedure: if there is a product state on some interaction graph G, then we can extend it to a product state on a graph G' which is G plus one additional clause C, so long as the clause C has at least one qubit not already in G. Thus, if a given graph G can be built up one clause at a time in an order such that there is always a previously unconstrained qubit brought in by each additional clause, G is PRODSAT.

Moving beyond our explicit construction, a complete characterization of product satisfiability can be found by analyzing the equations which a satisfying product state must obey [25]. This is a system of M algebraic equations in N complex unknowns and naive constraint counting suggests that $M \leq N$ should have solutions while $M > N$ should not. Since the system is sparse, however, a somewhat more detailed analysis is required to show:

Theorem 1. *G is PRODSAT for generic choices of projectors Π_m if and only if its interaction graph has a dimer covering of its clauses.*

Here a "a dimer covering of its clauses" is a pairing between qubits and clauses such that every clause appears paired with exactly one qubit and no qubit or clause appears more than once. The proof relies on 'product state perturbation theory', or in other words, the smoothness of the complex manifolds defining the projector space and the product state space.

If we apply the dimer covering characterization to the random interaction graph ensemble for G, we find the α_{ps} indicated in the phase diagram of Figure 7. In particular, for $\alpha < \alpha_{ps}$, such dimer coverings exist w.p. 1 in the thermodynamic limit while for $\alpha > \alpha_{ps}$ they do not. The location of the geometric transition for the existence of dimer coverings in known in the literature [33]. Thus, for $\alpha > \alpha_{ps}$ there are no satisfying product states, although there may still be satisfying entangled states.

We note that the dimer covering characterization of product states provides an explicit mapping between dimer coverings and generic product states. In the case $M = N$, this mapping is one-to-one and provides a handle on counting the number of product states and, with more work, the ability to estimate their linear dependence. There are many avenues to explore here.

2. *Bounding the degeneracy*

The existence of an UNSAT regime for large α follows immediately from the geometrization theorem and the existence of an UNSAT regime for classical SAT. That is, for $\alpha > \alpha_s^{\text{Classical}}$, typical graphs G are classically UNSAT and therefore, since the generic dimension R_G of the zero energy state space is minimal, they are also generically quantum UNSAT. In other words, the quantum SAT-UNSAT transition $\alpha_s \leq \alpha_s^{\text{Classical}}$.

This estimate of the SAT-UNSAT transition is not tight: the quantum UNSAT phase begins at a lower α than the classical UNSAT phase. This can be seen using another approach to bounding the ground state degeneracy. In this approach, one builds up a given graph G out of small clusters, each of which decimates the satisfying eigenspace by some known fraction. Indeed, if we consider two interaction graphs K and H on N qubits with that respective generic zero energy dimensions R_K and R_H, it is straightforward to show [13]

$$R_{K\cup H} \leq R_K \frac{R_H}{2^N} \tag{27}$$

for generic choices of projectors on K and H. As an example, let us build a graph G with M clauses, one clause at a time. Each individual clause H has $R_H = (1-1/2^k)2^N$ because it penalizes 1 out of the 2^k states in the local k-qubit space and leaves the other 2^{N-k} alone. Thus, adding each additional clause H_m decimates the satisfying subspace by at least a factor $(1-1/2^k)$ and:

$$R_G \leq 2^N (1-1/2^k)^M \tag{28}$$

Plugging in $M = \alpha N$ and taking N to infinity, we find that for $\alpha > -1/\log_2(1-1/2^k)$, R_G must go to zero. This simply reproduces the Pauling bound mentioned for classical k-SAT in Section III B.

However, one can do better by taking somewhat larger clusters H, calculating R_H exactly for these small clusters and then working out how many such clusters appear in the random graph G. This leads to much tighter bounds on α_s from above and in particular, as shown in Ref. 13, $\alpha_s(k) < \alpha_s^{\text{Classical}}(k)$ for all connectivities k.

3. *Quantum Lovasz local lemma*

The final technique that has been used to fill in the phase diagram of Figure 7 is the development of a quantum version of the Lovasz local lemma [7]. This lemma provides a nonconstructive proof that satisfying states must exist for k-QSAT instances built out of interaction graphs with sufficiently low connectivity – that is, graphs in which the degree of every qubit is bounded by $2^k/(ek)$. The QSAT ensemble which we study in fact has average degree αk but the degree distribution has an unbounded tail. By cutting the graph into low and high connectivity subgraphs, using the product state characterization on the high connectivity part and the Lovasz lemma on the low connectivity part, and carefully glueing these results back together, it is then possible to show that satisfying states exist for $\alpha < 2^k/(12ek^2)$.

For sufficiently large k, this result proves that $\alpha_s \geq O(2^k/k^2) \gg \alpha_{ps}$, establishing the entangled SAT regime indicated in Figure 7.

We now sketch the idea behind the classical and quantum Lovasz local lemmas.

Suppose we have some classical probability space and a collection of M events B_m, each with a probability of occurring $P(B_m) \leq p < 1$. We think of these as low probability 'bad' events, such as "the m'th clause of a k-SAT instance is not satisfied by σ" given a uniformly chosen configuration σ. In this particular case, $P(B_m) = p = 1/2^k$ for all clauses m. If there is a positive probability that no bad event comes to pass, then there is clearly an overall assignment of σ which satisfies all of the clauses. Thus, we would like to show this probability is positive.

If the events B_m are independent, this is clearly possible:

$$Pr(\bigwedge_m \neg B_m) = \prod_{m=1}^{M} (1 - Pr(B_m)) \geq (1-p)^M > 0 \tag{29}$$

In the k-SAT example, clauses are independent if they do not share any bits – thus this argument provides us the rather obvious result that k-SAT instances composed of only completely disconnected clauses are satisfiable. On the other hand, if the events B_m are dependent, it is clear that we can make

$$Pr(\bigwedge_m \neg B_m) = 0. \tag{30}$$

For instance, simply take a 3-SAT instance with 3 qubits and 8 clauses, each of which penalizes a different configuration. These clauses still have individually low probability ($p = 1/2^k$) but at least one of them is violated by any configuration.

The classical Lovasz local lemma [16, 35] provides an elementary method for relaxing the independence requirement a little bit. In particular, if each event B_m depends on no more than d other events $B_{m'}$ where

$$p\, e\, d \leq 1 \tag{31}$$

(Euler's constant $e \approx 2.7182\ldots$ being the basis of the natural logarithm) then the local lemma tells us that there is indeed a positive probability that no bad event happens:

$$Pr(\bigwedge_m \neg B_m) > 0 \tag{32}$$

This means that for connected k-SAT instances of sufficiently low degree, Lovasz proves the existence of satisfying configurations.

In the quantum generalization of the Lovasz lemma, probability is replaced by the relative dimension of satisfying subspaces. That is, for a QSAT projector Π of rank 1, the "probability of the clause being satisfied" is

$$\frac{\mathrm{Dim}(SAT)}{\mathrm{Dim}(\mathcal{H})} = \frac{2^k - 1}{2^k} = 1 - \frac{1}{2^k} \quad . \tag{33}$$

With the right definitions in hand, the generalization is also elementary and the result looks nearly identical to the classical case. However, now a positive probability that all projectors are satisfied tells us that there exists a (potentially quite entangled) quantum state of an N-qubit Hilbert space which satisfies the instance of k-QSAT.

The Lovasz local lemma is nonconstructive because it works by bounding (from below) the satisfying space degeneracy as the graph G is built up, so long as each additional clause does not overlap too many other clauses. In some sense this is dual to the arguments used to prove the UNSAT phase exists by bounding this degeneracy from above, but the technical details are somewhat more subtle since they require a more careful consideration of the interaction between additional clauses and the existing constraints.

In the last few years, computer scientists have developed a *constructive* version of the classical Lovasz local lemma. That is, there are now proofs that certain probabilistic algorithms will actually efficiently construct the Lovasz satisfying states [35]. Recent work suggests that a quantum generalization of this constructive approach may also be possible [8].

E. Satisfying the promise

Ideally, we would like to study an ensemble of k-QSAT instances which always satisfy the promise. Such an ensemble would only contain Yes-instances with strictly zero energy and No-instances with energy bounded away from zero energy by a polynomially small promise gap. Such an ensemble is hard to construct as one does not know *a priori* which instances have zero energy or not, let alone whether their energy might be exponentially small. The best we can hope to do is choose a random ensemble in which the promise is satisfied statistically – perhaps with probability 1 in the thermodynamic limit.

Physical arguments suggest that the k-QSAT ensemble that we study here satisfies the promise in this statistical sense and for $k = 2$ it can be proven. On the SAT side of the phase diagram, all of the arguments that have been constructed to date show the existence of strict zero-energy states in the thermodynamic limit. These arguments all rely on geometrization: the existence of generic zero energy states is a graph property and such properties are

either present or not in the thermodynamic limit of the random graph at a given α. Hence the zero energy phase as determined by such arguments is a strictly zero energy phase.

As statistical physicists, we expect that the UNSAT phase of k-QSAT has extensive ground state energy with relatively vanishing fluctuations for any k. If this is true, the promise that $E \geq O(N^{-a})$ fails to be satisfied only with exponentially small probability by Chebyshev's inequality. More generally, so long as the average ground state energy is bounded below by a polynomially small scale $E \geq O(N^{-b})$ with relatively vanishing fluctuations, the promise will be violated with only exponentially small probability for $a > b$.

For $k = 2$, it can shown rigorously that the expected ground state energy for $\alpha > \alpha_s = 1/2$ is bounded below by a nearly extensive quantity (*i.e.* $E \geq O(N^{1-\epsilon})$ for any $\epsilon > 0$). Also, we know that the SAT phase extends to $\alpha = 1/2$ because we can show that satisfying zero energy product states exist up to this clause density. Thus, the ensemble satisfies the promise with high probability in both phases. At the critical point, things are not quite so clear, but one might expect fluctuations around $E = 0$ at the scale $O(\sqrt{N})$ so that if the promise gap is chosen to be $O(N^{-1})$, the weight of the ensemble below the gap scale goes to zero.

F. Open questions

In closing let us take stock of where we are at in the analysis of QSAT with the set of results on SAT as our template. First, the phase diagram clearly needs more work starting with more precise estimates for the SAT-UNSAT boundary. Within the SAT phase we have identified one phase transition where the satisfying states go from being products to being entangled and the key question is whether there are any others and whether they involve a clustering of quantum states in some meaningful fashion. Second, we have not said anything about the performance of algorithms for QSAT or about the relationship between phase structure and algorithm performance. Apart from some preliminary work on the adiabatic algorithm for 2-SAT [19], this direction is wide open for exploration.

V. CONCLUSION

In this review, we have tried to provide a reasonably self-contained introduction to the statistical mechanics of classical and quantum computational complexity, starting at the venerable subject of classical complexity theory, and ending at an active current research frontier at the intersection in quantum computing, quantum complexity theory and quantum statistical mechanics. We hope that this review will not only encourage some of its readers to contribute to these fields of study, but that it will also have provided than with some of the background necessary for getting started.

Acknowledgements

We very gratefully acknowledge collaborations with Andreas Läuchli, in particular on the work reported in Ref. [25]. Chris Laumann was partially supported by a travel award of ICAM-I2CAM under NSF grant DMR-0844115.

[1] S. Aaronson. Guest column: NP-complete problems and physical reality. *SIGACT News*, 36(1), Mar 2005.
[2] D. Aharonov, D. Gottesman, and J. Kempe. The power of quantum systems on a line. May 2007, e-print arXiv:0705.4077.
[3] D. Aharonov and T. Naveh. Quantum NP - a survey. Oct 2002, e-print arXiv:quant-ph/0210077v1.
[4] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *Siam Rev*, 50(4):755–787, Jan 2008.
[5] B. Altshuler, H. Krovi, and J. Roland. Adiabatic quantum optimization fails for random instances of NP-complete problems. Aug 2009, e-print arXiv:0908.2782v2.
[6] B. Altshuler, H. Krovi, and J. Roland. Anderson localization casts clouds over adiabatic quantum optimization. Jan 2009, e-print arXiv:0912.0746.
[7] A. Ambainis, J. Kempe, and O. Sattath. A quantum Lovász local lemma. In *42nd Annual ACM Symposium on Theory of Computing*, 2009, e-print arXiv:0911.1696v1.
[8] I. Arad, T. Cubitt, J. Kempe, O. Sattath, M. Schwarz, and F. Verstraete. Private communication, June 2010.
[9] S. Arora and B. Barak. *Complexity Theory: A Modern Approach.* Cambridge University Press, 2009.
[10] F. Barahona. On the computational complexity of Ising spin glass models. *J. Phys. A: Math. Gen*, 15:3241–3253, Jan 1982.

[11] A. Braunstein, M. Mezard, and R. Zecchina. Survey propagation: an algorithm for satisfiability. *Rand. Struct. Alg.*, pages 201–226, Jan 2005.

[12] S. Bravyi. Efficient algorithm for a quantum analogue of 2-SAT. Feb 2006, e-print arXiv:quant-ph/0602108v1.

[13] S. Bravyi, C. Moore, and A. Russell. Bounds on the quantum satisfibility threshold. Jan 2009, e-print arXiv:0907.1297v2.

[14] G. Carleo, M. Tarzia, and F. Zamponi. Bose-einstein condensation in quantum glasses. *Phys. Rev. Lett.*, 103(21):215302, Nov 2009.

[15] O. Dubois. Upper bounds on the satisfiability threshold. *Theor. Comput. Sci.*, 265(1-2):187–197, 2001.

[16] P. Erdös and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. *Infinite and finite sets*, 2:609–627, 1975.

[17] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda. A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science*, 292(5516):472–475, Jan 2001.

[18] M. R. Garey and D. S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. Series of Books in the Mathematical Sciences. W. H. Freeman & Co Ltd, January 1979.

[19] J. Govenius. Junior paper: Running time scaling of a 2-QSAT adiabatic evolution algorithm. *Princeton Junior Paper*, 2008.

[20] A. K. Hartmann and M. Weigt. *Phase transitions in combinatorial optimization problems: basics, algorithms and statistical mechanics*. Wiley-VCH, Weinheim, Germany, 2005.

[21] M. B. Hastings. Quantum belief propagation: An algorithm for thermal quantum systems. *Phys. Rev. B*, 76(20), Nov 2007.

[22] F. Krzakala, A. Montanari, F. Ricci-Tersenghi, G. Semerjian, and L. Zdeborova. Gibbs states and the set of solutions of random constraint satisfaction problems. *P Natl Acad Sci Usa*, 104(25):10318–10323, Jan 2007.

[23] F. Krzakała, A. Rosso, G. Semerjian, and F. Zamponi. Path-integral representation for quantum spin models: Application to the quantum cavity method and monte carlo simulations. *Phys. Rev. B*, 78(13):134428, 2008.

[24] L. Landau. Zur Theorie der Energieubertragung. ii. *Physics of the Soviet Union*, 2:46–51, 1932.

[25] C. R. Laumann, A. M. Läuchli, R. Moessner, A. Scardicchio, and S. L. Sondhi. Product, generic, and random generic quantum satisfiability. *Phys. Rev. A*, 81(6):062345, Jun 2010.

[26] C. R. Laumann, R. Moessner, A. Scardicchio, and S. L. Sondhi. Phase transitions and random quantum satisfiability. *Quant. Inf. and Comp.*, 10(1):0001, Mar 2010.

[27] C. R. Laumann, S. A. Parameswaran, S. L. Sondhi, and F. Zamponi. AKLT models with quantum spin glass ground states. *Phys. Rev. B*, 81(17):174204, May 2010.

[28] C. R. Laumann, A. Scardicchio, and S. L. Sondhi. Cavity method for quantum spin glasses on the Bethe lattice. *Phys. Rev. B*, 78(13):134424, 2008.

[29] M. Leifer and D. Poulin. Quantum graphical models and belief propagation. *Ann. Phys.*, 323:1899–1946, 2008.

[30] M. Mézard. Physics/computer science: Passing messages between disciplines. *Science*, 301(5640):1685–1686, 2003.

[31] M. Mézard and A. Montanari. *Information, physics, and computation*. Oxford University Press Inc., New York, NY, USA, 2009.

[32] M. Mézard, G. Parisi, and R. Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297:812, Jan 2002.

[33] M. Mézard, F. Ricci-Tersenghi, and R. Zecchina. Two solutions to diluted p-spin models and XORSAT problems. *J. Stat. Phys.*, 111(3):505–533, May 2003.

[34] M. Mézard and R. Zecchina. Random k-satisfiability problem: From an analytic solution to an efficient algorithm. *Phys. Rev. E*, 66(5):056126, Jan 2002.

[35] R. A. Moser and G. Tardos. A constructive proof of the general Lovász local lemma. *J. ACM*, 57(2):1–15, 2010.

[36] R. Movassagh, E. Farhi, J. Goldstone, D. Nagaj, T. J. Osborne, and P. W. Shor. Unfrustrated qudit chains and their ground states. *Phys. Rev. A*, 82(1):012318, Jul 2010.

[37] G. Semerjian, M. Tarzia, and F. Zamponi. Exact solution of the Bose-Hubbard model on the Bethe lattice. *Phys. Rev. B*, 80(1):014524, Jul 2009.

[38] V. Smelyanskiy, S. Knysh, and R. Morris. Quantum adiabatic optimization and combinatorial landscapes. *Phys. Rev. E*, 70(3):036702, Jan 2004.

[39] A. P. Young, S. Knysh, and V. N. Smelyanskiy. Size dependence of the minimum excitation gap in the quantum adiabatic algorithm. *Phys. Rev. Lett.*, 101(17):170503, Jan 2008.

[40] A. P. Young, S. Knysh, and V. N. Smelyanskiy. First-order phase transition in the quantum adiabatic algorithm. *Phys. Rev. Lett.*, 104(2):020502, Jan 2010.

[41] C. Zener. Non-adiabatic crossing of energy levels. *Proc. Roy. Soc. London: Series A*, 137(833):696–702, 1932.