PHASE TRANSITIONS IN THE POLYNOMIAL HIERARCHY

N. LARACUENTE[†]

Abstract. This essay describes a subset of the theories and computational experiments relating complexity theory to phase transitions surrounding the famous “P vs. NP” problem. Primary areas of focus are: 1) the meaning and physical significance of robust asymptotic complexity theory 2) observations and theories of phase transitions in complexity 3) past attempts to characterize the class of problems known collectively as “boolean satisfiability.”

Term Essay for Nigel Goldenfeld’s Physics 563: Phase Transitions and the Renormalization Group

Department of Physics, University of Illinois, Urbana, IL 61801, USA
[Nicholas LaRacuente]laracue2@illinois.edu

[†] This material is based upon work supported by NSF Graduate Research Fellowship Program DGE-1144245. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

1. Introduction: Physics and Computational Complexity

This paper concerns the fundamental hardness of solving particular computable problems. A good example of this kind of problem is the *Traveling Salesman Problem* (TSP): given a list of cities and the estimated time to go from each city to another, can one travel through all of the cities, returning to one's starting point, in less than a given maximum time? While the statement of this problem is fairly simple, there is a $1-million dollar bounty for a proof that a computer can solve arbitrarily large instances of this problem in an amount of time that is polynomially bounded in the size of the problem statement (abbreviated "PTIME" or just "P"), or that no such algorithm can exist [1]. There are at the time of this writing 116 recorded attempts on "The P-versus-NP page," one of which survived peer review and has not been debunked (and that one doesn't claim to actually resolve the question) [2].

A *computational complexity class* is a set of problems solvable under specified runtime or memory bounds. The Traveling Salesman Problem is a popular representative of the computational complexity class NP, short for *non-deterministic polynomial time.* NP it contains those problems for which answers can be verified by a computer program in polynomially bounded time. In the TSP example, if we are provided a suggested route, it's easy to check whether the total length adds up to less than the allowable maximum - if so, the suggested route *satisfies* the TSP instance. The hard part is deciding the best route(s) to check, as there are in general exponentially many possible routes. A theoretical "non-deterministic" computer would be able to try all of the possible solutions in a sort of enlightened superposition (more powerful than quantum superposition), and decide afterward which branch it was supposed to have taken. A regular, deterministic computer can simulate this process but requires an exponentially long time to try all of the possibilities. P vs. NP asks whether a more sophisticated algorithm might do this in P. For a recent guide to P vs. NP, see Scott Aaronson's survey [3].

Our paper is not particularly concerned with implementation of solver algorithms or differences in the hardware upon which they run. A foundational result in computational complexity theory is the *Church-Turing* thesis stating the equivalence in *computability* between different models of computation [4]. Informally, this is usually extended to the claim that any two classical computers, however constructed, asymptotically approach the same *robust* complexity classes as one scales up the size of the input problem [5]. For the purposes of this paper, we will take this to be the definition of a *robust complexity class*: a complexity class independent of the computer on which it runs. Usually this means defining complexity classes broadly enough that a constant or polynomial speedup of any algorithm would still land in the same class. There is for robust complexity purposes no asymptotic distinction between Blue Waters and the earliest vacuum tube computers. The practically relevant differences are absorbed into polynomial degrees and constants (that obviously differ by many orders of magnitude). Challenges have appeared to the Church-Turing thesis from quantum computing [6] and even some less-powerful models [7], but we mostly ignore quantum computing in this paper, focusing on classical examples. Also, rather than attack the TSP directly, we will focus on easier-to-analyze equivalent problems.

It is natural to wonder what computational complexity has to do with physics. The practical but unexciting answer is that many computational physics tasks, such as some models of protein folding [8], extracting dynamical system equations from data [9] and even some Ising models [10] [11] are in the harder regions of NP. Much of the literature on how phase transitions relate to computational complexity exploits this fact, transforming abstract computable problems into questions about the ground state(s) of Hamiltonians [11]. There are more fundamental reasons why physicists should pay attention to complexity theory. One argument surrounding the "black hole firewall paradox" claims that the physics of the event horizon might depend on whether certain experiments could be performed with the (presumed quantum) computing resources available within the observable universe [12]. Others claim (though to evaluate such claims is beyond the scope of this paper) that P vs. NP may affect foundational principles of quantum mechanics [13]. A lecture a the Perimeter Institute's 2016 "It from Qubit" summer school titled "Why Physicists Should Care About the Complexity Zoo" discusses how recent developments in quantum theory connect to computational complexity [14]. Computational complexity in this sense relates not to a particular computer or algorithm, but to the rules of what algorithms and computers could exist. We really ask, "what questions do the universe allow to be easy?"

1.1. **The Classes.**

The idea of "hierarchy" in computational complexity comes from a classic result known as the "Time Hierarchy Theorem," which shows that there is at least a partial ordering on complexity classes ([15] p145). Another key concept is *reduction*:

Definition 1.1. *Consider two classes of computational problems, A and B. A* ***reduces*** *B if $\exists$ a polynomial-time algorithm to convert any instance of B into an instance of A, such that a solution of this A instance implies a corresponding solution of the B instance.*

If A reduces B, then A is at least as hard as B. The Time Hierarchy Theorem and the use of reduction allows us to classify problems by their complexity: hence we refer to "complexity classes." We concern ourselves with a particular set of robust complexity classes. The polynomial hierarchy includes multitudes of mathematically distinguished classes involving constructions with oracles, provers and other abstract objects. We will not concern ourselves with these details, instead focusing on P, NP and a few interesting related classes. For a full treatment of complexity classes on which this short list is based, see [15], from which this list is derived.

- **P** - problems decidable in *polynomial time* (polynomial in the problem size), often regarded by computation theorists as the class of tractable problems (though this need not be the case, as there is no restriction on polynomial degree or coefficients). Example: matrix multiplication.
- **NP-Intermediate (NPI)** - problems solvable in non-deterministic polynomial time, but which don't reduce everything in NP.
  Example: factoring integers.
- **NP-Complete (NPC)** - problems that reduce everything in NP, and are contained within NP (the hardest problems in NP). Example: TSP.
- **coNP** - the complexity class of proving that an NP problem instance, such as TSP, has no satisfying solutions. Possibly harder than NPC.

It is widely suspected (again see [15]) that these classes are all distinct.

2. Connection to Phase Transitions

Much effort has gone into trying to intuitively express what makes NP problems hard. Somewhat surprisingly, NPC problems tend to be easy in practice, despite being the hardest problems within NP. A common theme among articles referenced here is that most instances of a given NPC problem are easy, either because they have so many solutions that a simple guess-and-check algorithm would probably hit one almost immediately, or because they are obviously unsatisfiable.

The main thrust of this line of study is that there appears to be a phase transition separating the usually satisfiable levels of constrainedness from the usually unsatisfiable levels. Decisions are tricky near the boundary. A solver that attempts to build up solutions to a problem near the boundary will find many promising starts that take much time and lead nowhere.

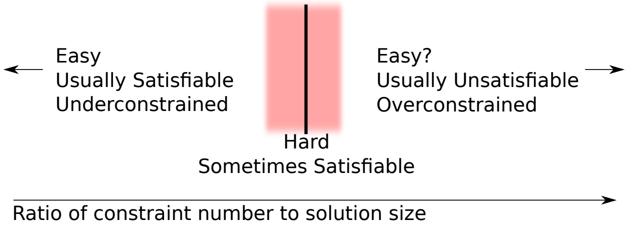

Figure 1. As we increase the constainedness of an NP-Complete problem, we expect a sharp, discontinuous transition from a bulk phase in which almost all instances are satisfiable to one in which almost all instances are unsatisfiable, represented by the thick, black line. Around it, there is a region that is thought to be most difficult for decider algorithms, repesented by the blurry pink region.

As discussed by Vardi [16], the overconstrained phase has not been definitively established as easy. This might be analogous to NP vs. coNP. In the underconstrained phase, a solver will usually find a satisfying certificate very quickly - at that point, the solver is finished. In contrast, a solver that has failed to return a certificate after running for some time has not necessarily proven that the instance is not satisfiable - it might just be looking in a wrong part of the space. Vardi's review concludes that the overconstrained region might be better labeled “less-hard” than easy.

2.1. **Generalities: Quenched Randomness & The Replica Trick.** Most studies on this topic involve instances of problems or corresponding physical models with a random coupling strength between each microscopic site or unit pair, known as *quenched disorder* [11]. This is the key attribute that distinguishes an *Ising spin glass* from a simple Ising model. A spin glass may hence have a random, complicated structure of local and global energy minima.

Studies of quenched disorder commonly apply the *replica trick* to simplify their calculations. Following discussion in [11] and using the notation $\bar{Z}$ for the average of a function Z, we may write the partition function for the average of N copies of a quenched system,

$$\overline{Z(J)^N} = \overline{\left(\sum_{\mathcal{C}} \exp(-\mathcal{H}(\mathcal{C},J))\right)^N} = \sum_{\mathcal{C}_1 \ldots \mathcal{C}_N} \overline{\exp(-\sum_{i=1}^{N} \mathcal{H}(\mathcal{C}_i,J))} \qquad (1)$$

where $\mathcal{H}$ is a temperature-including Hamiltonian that depends on a coupling scale J and microscopic coupling configuration $\mathcal{C}$. The physical intuition is that by replacing an average over an ensemble with an ensemble of an average, we remove the complicated differences between random instances from the calculation. The "trick" is then to take an analytic continuation from integer n to $n = 0$, using the Taylor expansion of $Z^n = \exp(n \ln Z)$ to rewrite the free energy for a single copy of the system as

$$\overline{F(J)} = \overline{-k_B T \ln Z} = -k_B T \lim_{n \to 0} \Big(\frac{\overline{Z(J)^n} - 1}{n} \Big) \tag{2}$$

As further discussed in [11], the analytic continuation used in the replica trick is not mathematically rigorous in all cases, but its applications are well-studied and its results often usable as a guide for more rigorous techniques. Another aspect discussed in [11] is to look at where replica symmetry breaks, suggesting more complicated dynamics.

2.2. **Graphs as Spin Glass.**

In 1986, Fu and Anderson reduced the NPC Graph Partitioning problem to finding the ground state of an Ising spin glass Hamiltonian [17]. This result motivated subsequent studies, including the much greater attention paid to boolean satisfiability (as we discuss in section 3). While this is not the most prominent example, its chronological precedence and relative simplicity make it a good starting point.

We define a graph as a set of vertices $V = \{v_1 ... v_n\}$ and edges $E = \{(v_i, v_j) : i, j \in 1...n\}$. We define a coupling parameter α to be probability of a random vertex pair to be connected by an edge. Following the work of Fu and Anderson [17], we assume n to be even and define the NPC problem: what is the smallest ϵ for which we can split V into two equally sized subsets V_1 and V_2 such that fewer than ϵ edges connect vertices in V_1 to those in V_2? Fu and Anderson construct an Ising model Hamiltonian by associating a spin s_i to each vertex v_i, such that $s_i = +1$ if $v_i \in V_1$, and $s_i = -1$ if $v_i \in V_2$. They define a constant coupling scale J and Ising coupling parameter $J_{i,j} = J$ if $(v_i, v_j) \in E$, and $J_{i,j} = 0$ otherwise. The Hamiltonian (in our notation, which differs from theirs) is

$$\mathcal{H} = -\sum_{i,j} J_{i,j} s_i s_j = -J \Big(\frac{1}{2} N(N-1)\alpha - 2\epsilon \Big) \tag{3}$$

This implies that $\epsilon = \mathcal{H}/2J + N(N-1)\alpha/4$, and we have reduced the original problem to that of finding the ground state for $\mathcal{H}$. Fu and Anderson then note that there should be a phase transition between large α implying a highly-connected phase, and vanishing α implying a mostly disconnected phase. They show the existence of a phase transition using the replica trick. They state (about the local energy minima in an Ising spin glass):

> If there are too many local minima, sitting very close to each other, the transition between neighbouring minima would involve O(1) spins, there will be no rigidity of the low temperature phases and hence no phase transition (e.g. the infinite-range antiferromagnetic model), and the optimisation will be easy. Computationally non-trivial cases arise when local minima are numerous but not excessively numerous, the distances between them large but not of the order of N. These are features shared by the SK spin glass, and we expect the existence of a spin-glass-like transition in these systems to reflect the difficulty involved in optimisation ([17], p1613).

Related to Fu and Anderson's work, [11] discusses the use of the Potts model to understand percolation. A *connected component* or *cluster* in a graph is a subset of the vertices $U \subset V$ such that each vertex pair $v_i, v_j \in U$ are joined by some path through edges between vertices in U. The percolation phenomenon is a phase transition between a mostly disconnected phase in which there are many small clusters, and a mostly-connected phase in which graph is dominated by a single large cluster. An analytic continuation used in [11] reproduces known results from graph theory regarding cluster size and number. We take this example as a sort of appetizer to the more prominent question of boolean satisfiability.

3. Boolean Satisfiability

To set up a general boolean satisfiability problem, we start with a set of n boolean (binary) variables $V = \{x_i : i \in 1...n, x_i \in \{0,1\}\}$. We may construct formulas by combining these variables with boolean operations. For brevity, we will restrict our attention primarily to **K-SAT**: Let $\vee$ be the logical OR operation, and $\wedge$ be the logical AND. Let $\bar{x}_i$ be the negation of x_i (1 if $x_i = 0$, 0 if $x_i = 1$). We construct K-SAT formulas of the form:

$$(v_{1,1} \vee ... \vee v_{1,K}) \wedge ... \wedge (v_{M,1} \vee ... \vee v_{M,K})$$

for some $M \in \mathbb{N}$, where each $v_{j,k}$ is an arbitrary x_i or $\bar{x}_i$. In other words, K-SAT is a conjunction of disjunctions of variables and their negations. We call each of the M disjunctions as a *clause* and define a coupling parameter $\alpha = M/n$ to be the ratio of clause number to variable number. K-SAT is NPC when $K \geq 3$. Reductions in NPC imply that any K-SAT with $K \geq 3$

(or any NPC problem, including the TSP) is for robust complexity purposes equivalent to 3-SAT.

3.1. **2-SAT's Continuous Phase Transition.** The salient feature of 2-SAT is that it is equivalent to an implication graph ([15], p184). Given a clause $(x_i \vee x_j)$ with any $i \neq j \in 1...n$, we define 2 directed edges representing the implications $\bar{x}_i \implies x_j$ and $\bar{x}_j \implies x_i$. Restricting the clause length to 2 allows us to identify nodes of this graph with variables and their negations. This is not true for higher values of K. In 3-SAT, for example, we may take the clause $(x_i \vee x_j \vee x_k)$. From this clause, we get the implications $\bar{x}_a \implies (x_b \vee x_c)$ and $\overline{(x_b \vee x_c)} \implies x_a$, where a, b, c are any permutation of $i \neq j \neq k \in 1...N$. A K-SAT implication graph with $K > 1$ naturally contains implications to and from clauses of length $K - 1$. This allows simple graph theory methods to solve and analyze 2-SAT, but not higher K values.

Two original proofs that 2-SAT has a continuous phase transition at critical ratio $\alpha_c = 1$ from the usually-satisfiable to usually-unsatisfiable phase were published simultaneously in 1992 ([18], [19]), and reference [19] claims that there existed another simultaneous (unpublished) manuscript titled "On random 2-SAT" by W. Fernandez de la Vega. Fernandez de la Vega's later review article [20] provides a centralized narrative of this and related discoveries as of 2001. These proofs relate the unsatisfiability of a 2-SAT instance to the presence of particular path structures containing a variable to its contradiction. The number of variables that are fully constrained is continuous across the critical threshold α_c [21], which does not hold for higher K.

3.2. **Phases and Clustering in Solutions of Random K-SAT.** Much of the interest in phase transitions as a paradigm for complexity flows from the paper "Critical Behavior in the Satisfiability of Random Boolean Expressions" by Scott Kirkpatrick and Bart Selman (1994) [22]. Kirkpatrick and Selman use finite size scaling to determine the point of the critical transition in K-SAT for several values of K above 3. One may understand the background of this technique from the derivation of equation 9.261 in Goldenfeld's textbook ([23] p282). In a physical system such as an Ising model, let L be the (finite) system size, T be the temperature, T_c be the critical temperature in the asymptotic infinite-size limit, $J(T)$ be the temperature-dependent scale of coupling between microscopic units (Goldenfeld calls this number "K," but we want to avoid confusion with the K in K-SAT), and $\xi(J(T), L^{-1})$ be the (finite system) correlation length. Goldenfeld expands around $T = T_c$ as

$$\frac{L}{\xi(J(T), L^{-1})} = A + B\left(\frac{T - T_c}{T_c}\right)L^{1/\nu} + h.o.t. \tag{4}$$

where ν is the commonly-defined critical exponent, and A and B are constants. At $T = T_c$ (equivalently $J(T) = J(T_c)$), this ratio always attains the same value. Kirkpatrick and Selman use an analogous technique to derive their figure 3.A, using α as the analogous variable to $J(T)$ and the fraction of unsatisfiable instances as analogous to $L/\xi(J(T), L^{-1})$. They estimate the critical ratio of clauses to variables as $\alpha_c = 4.17^{+}_{-}0.05$.

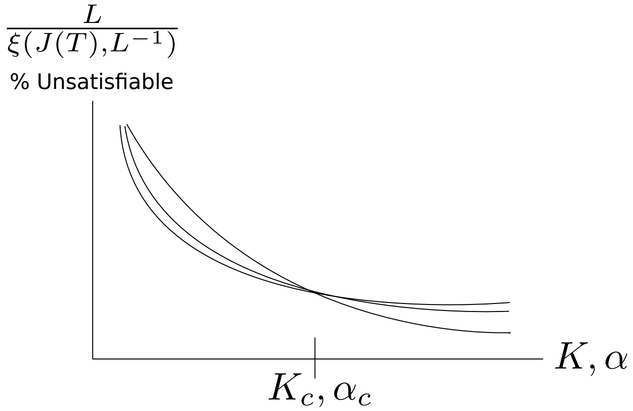

FIGURE 2. Qualitative schematic of finite size scaling, similar to figure 9.11 in [23]. Each curve is a physical system or problem of differing size L. At the critical point, the ratio of size to correlation length ξ or % unsatisfiable instances becomes size-independent.

Martin, Monasson and Zecchina describe in their 2001 review article [11] the use of the replica trick to analyze K-SAT. Conceptually, their picture follows that of Monasson, Zecchina and others in [21] that distinguishes K-SAT for $K \geq 3$ from the $K = 1, 2$ cases by the emergence of a discontinuity in the average fraction of variables that are fully constrained at α_c. They also note that an analysis of where replica symmetry breaks shows a clustering phase in the space of solutions to K-SAT (which we discuss at the end of this subsection).

A later paper by Achlioptas, Naor and Peres titled "Rigorous location of phase transitions in hard optimization problems" [24] analytically proves a bound for the critical ratio for K-SAT as $\alpha_c(K) \in (2^K \ln 2 - K, 2^K \ln 2)$. Their technique is based on the fact that for a random variable X,

$$\text{Prob}(X > 0) \geq \langle X \rangle^2 / \langle X^2 \rangle \tag{5}$$

In their case, X is a measure of satisfying assignments for a K-SAT instance F weighted by a function $w(\sigma, F)$, where σ is an assignment of the variables in F. They choose a weight function such that: 1) $w(\sigma, F) = 0$ if σ does not satisfy F 2) w penalizes assignments that satisfy more variables/negations in an average clause than would a random assignment, which prevents equation 5 from becoming trivial as it would under a naive weighting.

Mèzard, Mora and Zecchina explain the clustering idea more deeply in another article [25]. Let S be the set of solutions (satisfying assignments) in a K-SAT instance. Let w and z be 2 distinct solutions. We define a *step* through solution space as inverting a constant number of variable assignments in w to produce some w' that is also a solution. We define a *path* from w to z as a sequence of steps that starts with w as the 1st input and produces z as the last output. Well below the phase transition ($\alpha << \alpha_c$), we expect that with high probability, $\exists$ a path through solution space from w to z $\forall w, z$. Just below the phase transition we enter the clustered phase, in which we may write the solution space as a disjoint union $S = S_1 \cup S_2 \cup ...$ such that $S_m \cap S_l = \emptyset \ \forall m, l$, $\exists$ a path with high probability from w to z if $w, z \in S_m$ for some m, and there probably $\nexists$ a path from w to z if $w \in S_m, z \in S_l$ when $m \neq l$. In other words, as we approach α_c from below, we enter a sort of mostly disconnected small-cluster regime, but where solutions are still numerous. As we pass α_c and enter the overconstrained phase, we expect solutions to become rare. The article acknowledges that a clustering phase also appears in XOR-SAT, a known P problem, so it alone is not capable of distinguishing P from NP. Still, it is reminiscent of the percolation phenomenon in section 2.2 and interesting enough to mention.

3.3. **Coppersmith's Decimation for Boolean Classification.** So far, we have not discussed approaches that explicitly mention the renormalization group (RG) in analyzing complexity theory. We now turn our attention to a 2008 attempt by Susan Coppersmith to construct an RG-based classification of boolean functions [26]. Let $F(x_1, ..., x_n)$ be a boolean function of the variables $x_1, ..., x_n$. F could be for example a K-SAT instance, but we need not restrict to this case. Coppersmith defines the following decimation procedure:

$$\begin{aligned} &F'(x_1, ...x_{i-1}, x_{i+1}, ..., x_n) = \\ &F(x_1, ...x_{i-1}, 0, x_{i+1}, ..., x_n) \oplus F(x_1, ...x_{i-1}, 1, x_{i+1}, ..., x_n) \end{aligned} \tag{6}$$

where $\oplus$ denotes the binary exclusive OR (XOR) operation that is 1 if its inputs are different, 0 if they are the same. F' is a function of 1 fewer variable than F. Coppersmith notes that one can write a boolean function in polynomial form:

$$F(x_1, ..., x_n) = A_{00...00}\bar{x}_1...\bar{x}_{n-1}\bar{x} \oplus A_{00...01}\bar{x}_1...\bar{x}_{n-1}x_n \oplus ... \oplus A_{11...11}x_1...x_n \tag{7}$$

using the standard multiplication operation. The coefficients $A_{x_1...x_n} = F(x_1, ..., x_n)$ for a random boolean function are essentially independent random variables.

For many general classes of boolean formulas, the probability of a randomly chosen such coefficient being 1 approaches 1/2 exponentially fast with the number of decimations. Since this is a fixed point of the RG transform, Coppersmith names it the *generic phase.* Many restricted classes of boolean function either approach a different fixed point or approach the 1/2 fixed point more slowly. Based on these observations, Coppersmith conjectures that easily computable boolean functions might be written as a non-generic-phase function $\oplus$ a generic-phase but restricted “remainder” term. At the time of this writing, Coppersmith’s conjecture remains unproven.

4. Conclusions and Further Reading

The link between phase transitions and computational complexity is not proven. Moshe Y. Vardi’s skeptical presentation at the 2012 Workshop on Finite and Algorithmic Model Theory [16] summarizes two common doubts: 1) experiments requiring exponential time that prevents use of large instances 2) problems in P showing structures otherwise associated with hardness. We might also consider: 3) reliance on non-rigorous methods like the replica trick. None of these criticisms falsify the conjecture that hardness relates to phase transitions. They might be a good starting point for future work on the topic, knowing what has been done and where it may appear deficient.

We have focused on the most classic examples of NP and phase transitions, leaving out very interesting discussions on the TSP and other problems [11] [27] [28], as well as newer studies of modern algorithms for random satisfiability [29] that problem’s relationship to chaos [30], or an analogy with Bose-Einstein condensation [31]. While it seems unlikely that the phase transition analogy will provide a quick resolution to P vs. NP, it succeeds in suggesting an intuitive reason for the subtle complexity of the polynomial hierarchy.

References

[1] P vs NP Problem | Clay Mathematics Institute. http://www.claymath.org/millennium-problems/p-vs-np-problem.

[2] P-versus-NP page. https://www.win.tue.nl/ gwoegi/P-versus-NP.htm.

[3] Scott Aaronson. Shtetl-Optimized Blog Archive My 116-page survey article on P vs. NP, 2017.

[4] Todd Rowland. Church-Turing Thesis. Wolfram MathWorld.

[5] Lance Fortnow. The Efficient Church-Turing Thesis. http://blog.computationalcomplexity.org/2006/12/efficient-church-turing-thesis.html.

[6] S. Kundu, R. Kundu, S. Kundu, A. Bhattachaijee, S. Gupta, S. Ghosh, and I. Basu. Quantum computation: From Church-Turing thesis to Qubits. In *2016 IEEE 7th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, pages 1–5, October 2016.

[7] Scott Aaronson and Alex Arkhipov. The Computational Complexity of Linear Optics. *Theory of Computing*, 9(1):143–252, 2013.

[8] Bonnie Berger and Tom Leighton. Protein Folding in the Hydrophobic-Hydrophilic (HP) Model is NP-Complete. *Journal of Computational Biology*, 5(1):27–40, January 1998.

[9] Jens Eisert Toby S. Cubitt and Michael M. Wolf. Extracting Dynamical Equations from Experimental Data is NP Hard. *Physical Review Letters*, 108(12), 2012.

[10] Barry A. Cipra. SIAM: The Ising Model Is NP-Complete, 2000.

[11] Olivier C. Martin, Rémi Monasson, and Riccardo Zecchina. Statistical mechanics methods and phase transitions in optimization problems. *Theoretical Computer Science*, 265(1):3–67, August 2001.

[12] Daniel Harlow and Patrick Hayden. Quantum computation vs. firewalls. *Journal of High Energy Physics*, 2013(6):85, June 2013.

[13] Arkady Bolotin. Computational solution to quantum foundational problems. *arXiv:1403.7686 [quant-ph]*, March 2014. arXiv: 1403.7686.

[14] Why physicists should care about the complexity zoo. https://www.perimeterinstitute.ca/videos/why-physicists-should-care-about-complexity-zoo, 2016.

[15] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994. Google-Books-ID: 6Z6_QgAACAAJ.

[16] M. Y. Vardi. Phase transitions and computational complexity. http://www.lsv.fr/Events/fmt2012/program.php.

[17] Yaotian Fu and P. W. Anderson. Application of statistical mechanics to NP-complete problems in combinatorial optimisation. *Journal of Physics A: Mathematical and General*, 19(9):1605, 1986.

[18] Andreas Goerdt. A threshold for unsatisfiability. In *Mathematical Foundations of Computer Science 1992*, pages 264–274. Springer, Berlin, Heidelberg, August 1992.

[19] V. Chvatal and B. Reed. Mick gets some (the odds are on his side) [satisfiability]. In *Proceedings., 33rd Annual Symposium on Foundations of Computer Science*, pages 620–627, October 1992.

[20] W. Fernandez de la Vega. Random 2-SAT: results and problems. *Theoretical Computer Science*, 265(1):131–146, August 2001.

[21] Rémi Monasson, Riccardo Zecchina, Scott Kirkpatrick, Bart Selman, and Lidror Troyansky. Determining computational complexity from characteristic phase transitions. *Nature*, 400(6740):133–137, July 1999.

[22] Scott Kirkpatrick and Bart Selman. Critical Behavior in the Satisfiability of Random Boolean Expressions. *Science*, 264(5163):1297–1301, May 1994.

[23] Nigel Goldenfeld. *Lectures on Phase Transitions and the Renormalization Group*. Westview Press, 1992. Google-Books-ID: DdB1__nl7CYC.

[24] Dimitris Achlioptas, Assaf Naor, and Yuval Peres. Rigorous location of phase transitions in hard optimization problems. *Nature*, 435(7043):759–764, June 2005.

[25] M. Mézard, T. Mora, and R. Zecchina. Clustering of Solutions in the Random Satisfiability Problem. *Physical Review Letters*, 94(19), 2005.

[26] S. N. Coppersmith. Using the Renormalization Group to Classify Boolean Functions. *Journal of Statistical Physics*, 130(6):1063–1085, March 2008.

[27] J. L. Jacobsen, N. Read, and H. Saleur. Traveling Salesman Problem, Conformal Invariance, and Dense Polymers. *Physical Review Letters*, 93(3), 2004.

[28] Peter F. Stadler, Wim Hordijk, and José F. Fontanari. Phase transition and landscape statistics of the number partitioning problem. *Physical Review E*, 67(5), 2003.

[29] Hendrik Schawe, Roman Bleim, and Alexander K. Hartmann. Phase Transitions of the Typical Algorithmic Complexity of the Random Satisfiability Problem Studied with Linear Programming. *arXiv:1702.02821 [cond-mat]*, February 2017. arXiv: 1702.02821.

[30] Róbert Sumi, Melinda Varga, Zoltán Toroczkai, and Mária Ercsey-Ravasz. Order-to-chaos transition in the hardness of random Boolean satisfiability problems. *arXiv:1602.05152 [cond-mat]*, February 2016. arXiv: 1602.05152.

[31] Claudio Angione, Annalisa Occhipinti, Giovanni Stracquadanio, and Giuseppe Nicosia. Bose-Einstein condensation in satisfiability problems. *European Journal of Operational Research*, 227(1):44–54, May 2013.