Lecture 2: Problem mappings and a bit of (classical) complexity theory fun

September 30, 2022

1 Overview: what is in this lecture

- What it does (and more importantly doesn't) mean for a problem to be NP-Hard, with cautionary example
  - Funtime bonus! how to show Super Mario is (NP)-Hard
- The relationship between Ising and QUBO formalisms and why they can be used almost interchangeably
- Mapping more connected graphs to less connected: minor embedding and parity encoding
- Mapping higher order interactions
- Mapping higher than binary variables (one hot, binary encoding, and domain wall)

2 NP-hardness

- NP stands for non-deterministic polynomial (not "non-polynomial"!): this means that a "maximally lucky" Monte-Carlo type algorithm[1] could solve

[1] technically any algorithm run on a non-deterministic Turning machine which gets maximally lucky

the problem in polynomial time, even if every bit is wrong this only takes n bit flips

- P?=NP, can a non-cheating i.e. not maximally lucky algorithm solve these in polynomial time, strongly suspected but not proven that P $\neq$ NP
- Can be mapped into each other in polynomial time/space, showing that one can be solved in poly time shows that all can
- There are harder classes of problems for example #P hard $\rightarrow$ how many solutions does a problem have, even "maximally lucky" algorithms cannot solve these in polynomial time because there may be exponentially many solutions

3 Using NP-hard problems to benchmark solvers

- Finding ground state of Ising problem D-Wave chimera graph is NP-hard (easy to show actually, just max-cut mapping+how to map arbitrary graphs, which I will explain later)
- Take random Ising instances and use them for benchmarking (these are hard problems right? the problem class has "hard" right in the name) [1, 2]
- Actually no, [3] showed using spin glass theory that these problems are easy to solve using simulated annealing type algorithms (no finite T spin glass transition)
- What is going on? did the authors of [3] accidentally show that P=NP?
  - No, they showed that that these problems are *typically* easy, NP-hardness is about what can be mapped and is therefore a *worst case* statement[2]

4 More on NP-hardness

- Many other problems are actually NP-hard but typical cases are (at least computationally) easy, my favourite example, playing Super Mario (not joking, see [4] for a mapping of 3-SAT to Super Mario and other classic games)
- How it works (slightly simplified from the paper but same idea):
  1. For each variable a_i you can choose one pipe True or False, for true you get to a pipe for all clauses which involve a and if you chose false you get one for all involving $\neg a$

[2] I am not trying to belittle the authors of [1, 2], they do amazing work and are leaders in the field, I would have probably made the same mistake in their shoes

2. These pipes go underneath boxes where you can release invincibility stars
3. Then you go where the stars are, there is fire, if at least one star you can make it through, if not, too bad
4. Can only beat the level if you find a satisfying arrangement!

- In all seriousness, [4] is a really accessible way to learn the basic of how these hardness proofs work
- So what do you want for benchmarking? *uniform* hardness, much more difficult to prove, doesn't just require you to show mapping
- Final note: even though these problems all map to each other with polynomial overhead, not all polynomial overheads are equal (big difference in practice between n^2 and n^3 for example) → finding good mappings is important

5 Mapping problems to quantum annealers

- What you have (hardware) → Ising model with two body interactions and not all connections allowed

$$H_{\text{Ising}} = \sum_{ij \in \chi} J_{ij} Z_i Z_j + \sum_i h_i Z_i \tag{1}$$

- What real problems often have
  - Usually not expressed as Ising models
  - Unlikely to match the graph of your hardware (unless you design special hardware to match the structure)
  - Interactions involving more than two variables (think 3-SAT)
  - Variables may be higher than binary
- In principle all of these issues are solvable assuming your hardware is big enough and has enough dynamic range (range of values the h's and J's can take)

6 First mapping QUBO to Ising

- Physicists love Ising models, but no-one else cares
- Optimisation problems often represented as QUBO's (Quadratic Unconstrained Binary Optimisation)

$$E = \vec{x}^T * Q * \vec{x} \tag{2}$$

- $\vec{x}_i \in \{0,1\}$ the B in QUBO (U comes from the fact that any value is allowed)
- Q involves both diagonal and off diagonal elements 2=quadratic=the Q in QUBO
- Goal is to minimize E (the O in QUBO)

How to map this to an Ising model:

- $x_i x_i = x_i$ up to irrevelvant constant offset, can transform to $-\frac{Q_{ii}}{2} Z_i$[3]
- $x_i x_j$ for $i \neq j$, a bit trickier, because $x_i x_j$ is only non-zero if both variables are 1 need both single and two body terms
  1. $-Z_i - Z_j$ will give $|11\rangle$ a higher energy than $|10\rangle$ or $|01\rangle$ but will give $|00\rangle$ an even lower energy
  2. $Z_i Z_j$ can be used to offset $|00\rangle$ so that $|10\rangle$, $|01\rangle$, and $|00\rangle$ all have the same energy
  3. $-Z_i - Z_j + Z_i Z_j$ will give -1 for $|10\rangle$, $|01\rangle$, and $|00\rangle$ and $+3$ for $|11\rangle$ $\rightarrow$ normalize by dividing by 4
  4. End up with $\frac{Q_{ij}}{4}(-Z_i - Z_j + Z_i Z_j)$ (note this also works for diagonal elements if we recall that $Z_i Z_i$ gives the identity
- Could have also taken a shortcut $x_i \rightarrow \frac{1}{2}(1 - Z_i)$ and factored out and ignored constants

7 How to map to hardware graph

- Minor embedding: take a graph minor (a connected subcomponent of a graph), and couple together with strong ferromagnetic coupling

$$-\lambda \sum_{\text{chains}} \sum_{(i,j \in \text{chain})} Z_i Z_j \tag{3}$$

- As long as λ is "big enough" I can force all qubits in chain to take same value [5]
- Act like one variable
- Graph needs to be non-planar (needs some crossings), but can have highly local connectivity
- Quasi-planar graphs (local connectivity in small region) can embed fully connected graphs but require n^2 variables [6]

[3] Recall $\langle 1| Z_i |1\rangle = -1$

- Alternative approach: parity mapping, each physical qubit corresponds to a coupler in a fully connected graph, four body terms enforce logical consistency (all neighbouring qubits see same value)
- Parity schemes, such as the LHZ (Lechner-Hauke-Zoller) scheme [7, 8], flipping a single variable, need to keep flipping until all constraints are satisfied
- Not clear which is better, LHZ allows for more clever decoding, but numerics show minor embedding performs better in some practical cases [9]

8 More than 2 body terms

- Core idea: add constrained auxilliary qubits and penalize those as well
- Simple example: $Z_iZ_jZ_k$, add a single auxilliary qubit, a, and constrain with $-\lambda(Z_iZ_a + Z_jZ_a + Z_kZ_a)$, for large λ this constrains a to take a majority vote of the three qubits
- Problem: without adding more penalties, the $|000\rangle$ state gets a -3λ contribution to the energy, while $|011\rangle$ only gets $-\lambda$
- Solution: add $\frac{\lambda}{2}(Z_iZ_j + Z_jZ_k + Z_iZ_k)$, now we get $-\frac{3}{2}\lambda$ contribution from both, everything else works out by symmetry (try it if you don't believe me)
- Now let's actually make a $Z_iZ_jZ_k$, penalizing the majority vote $-Z_a$ almost does the right thing, except for it gives $|111\rangle$ and $|000\rangle$ the wrong energies, fix by adding single body terms after some algebra we find

$$Z_iZ_jZ_k \rightarrow$$
$$Z_i + Z_j + Z_k + 2Z_a + \lambda\left(\frac{1}{2}(Z_iZ_j + Z_jZ_k + Z_iZ_k) - Z_iZ_a - Z_jZ_a - Z_kZ_a\right) \tag{4}$$

- $Z_iZ_jZ_k$ terms can be chained together to make arbitrary multi-body terms $(Z_iZ_jZ_kZ_l \rightarrow Z_iZ_jZ_a + Z_aZ_kZ_l)$ [10] alternative trick based on symmetry in [11]
- Many other ways to do this I am not going to talk about, this is just to give you a flavour of how this works

9 Higher-than-binary variables

- Variables which represent more than two mutually exclusive possibilities, but still only interact pairwise discrete quadratic modes (DQMs)

- Examples: scheduling (where individual events may use the same resource and therefore conflict), colouring (where adjacent nodes can't be the same colour)
- Most efficient way to do this in principle: encode each value in a binary string (binary encoding)
- *In principle* we know how to make $ZZZ...Z$ terms, and these can encode any interaction
- In practice:
  - If we have two variables of size m than it will take m^2 Ising terms to express each interaction
  - Anything ZZZ or higher will require at least one auxilliary qubit
  - Not very practical for general interactions
  - Some specific interactions (i.e. multiplication of numbers) can be expressed efficiently in this way [12]
- A different approach: unary encodings, number of qubits scales with m rather than $log(m)$, but interactions easier
- One-hot encoding, enforce constraint that exactly one of a set of m qubits takes a $|1\rangle$ value, easier to imagine as a QUBO, set constraint $\lambda(\sum_i x_i - 1)^2$
- Since each qubit corresponds to a value the variable can take, interactions are of the form $x_{i,\alpha}x_{j,\beta}$, where (α, β) index variables and (i, j) index values
- A different way to do this, domain-wall encoding [13], use $m-1$ Ising Qubits in a frustrated chain $-\lambda(-Z_1 + \sum_{i=1}^{m-2} Z_i Z_{i+1} + Z_{m-1})$
- m fold degenerate ground state of domain wall states (where exactly one term is frustrated), can be addressed by terms of the form $Z_{i+1} - Z_i$ which give zero if there is no domain wall between i and $i+1$, but 1 if there is, products of these terms are quadratic
- Domain-wall encoding is a new idea, but recent work shows it performs better [14], and it can be shown to be maximally optimal if generic interactions are desired [15]

10 Key Points

- NP-hard does not mean "non-polynomial", instead non-deterministic polynomial relating to a (fictional) maximally lucky device
- It is a "worst case" statement random instances may not be hard (Super Mario levels are not typically (computationally) hard)

- Uniform hardness is the case where random instances are hard, much more difficult to prove
- Real problems do not look like Ising models, and need to be mapped to hardware graphs
- Most mapping involves constraining to an optimal subspace and potentially adding extra qubits
- Better problem mapping for annealing is an area of active research

References

[1] Sergio Boixo, Troels F. Rønnow, Sergei V. Isakov, Zhihui Wang, David Wecker, Daniel A. Lidar, John M. Martinis, and Matthias Troyer. Evidence for quantum annealing with more than one hundred qubits. *Nature Physics*, 10(3):218–224, Mar 2014.

[2] Troels F. Rønnow, Zhihui Wang, Joshua Job, Sergio Boixo, Sergei V. Isakov, David Wecker, John M. Martinis, Daniel A. Lidar, and Matthias Troyer. Defining and detecting quantum speedup. *Science*, 345(6195):420–424, 2014.

[3] Helmut G. Katzgraber, Firas Hamze, and Ruben S. Andrist. Glassy chimeras could be blind to quantum speedup: Designing better benchmarks for quantum annealing machines. *Phys. Rev. X*, 4:021008, Apr 2014.

[4] Greg Aloupis, Erik D. Demaine, Alan Guo, and Giovanni Viglietta. Classic Nintendo Games are (Computationally) Hard, 2012. arXiv:1203.1895.

[5] Vicky Choi. Minor-embedding in adiabatic quantum computation: I. the parameter setting problem. *Quantum Information Processing*, 7(5):193–209, Oct 2008.

[6] Vicky Choi. Minor-embedding in adiabatic quantum computation: Ii. minor-universal graph design. *Quantum Information Processing*, 10(3):343–353, Jun 2011.

[7] Wolfgang Lechner, Philipp Hauke, and Peter Zoller. A quantum annealing architecture with all-to-all connectivity from local interactions. *Science Advances*, 1(9), 2015.

[8] Andrea Rocchetto, Simon C. Benjamin, and Ying Li. Stabilisers as a design tool for new forms of Lechner-Hauke-Zoller annealer. *Science Advances*, 2(10), 2016.

[9] Tameem Albash, Walter Vinci, and Daniel A. Lidar. Simulated quantum annealing with two all-to-all connectivity schemes. *Phys. Rev. A*, (94):022327, 2016.

[10] Martin Leib, Peter Zoller, and Wolfgang Lechner. A transmon quantum annealer: decomposing many-body Ising constraints into pair interactions. *Quantum Science and Technology*, 1(1):015008, 2016.

[11] N. Chancellor, S. Zohren, P. A. Warburton, S. C. Benjamin, and S. Roberts. A direct mapping of max k-sat and high order parity checks to a chimera graph. *Scientific Reports*, 6(1):37107, 2016.

[12] David Joseph, Adam Callison, Cong Ling, and Florian Mintert. Two quantum ising algorithms for the shortest vector problem: one for now and one for later, 2020. arXiv:2006.14057.

[13] Nicholas Chancellor. Domain wall encoding of discrete variables for quantum annealing and QAOA. *Quantum Science and Technology*, 4(4):045004, aug 2019.

[14] Jie Chen, Tobias Stollenwerk, and Nicholas Chancellor. Performance of domain-wall encoding for quantum annealing, 2021. arXiv:2102.12224.

[15] Raouf Dridi Jesse Berwald, Nicholas Chancellor. Understanding domain-wall encoding theoretically and experimentally, 2021. arXiv preprint 2108.12004.