max-SAT, multi-body frustration, and multi-body sampling on a two local Ising system

AQC 2016 Los Angeles

Nick Chancellor

June 29, 2016

Acknowledgements

Collaborators:

Stefan Zohren (Oxford)
Paul Warburton (UCL)
Simon Benjamin (Oxford)
Stephen Roberts (Oxford)

Supervisor:

Viv Kendon (Durham)

Funding:

NC: UK Engineering and Physical Sciences Research Council (EPSRC)
SZ: Lockeed Martin, Nokia, and EPSRC
PW: Lockheed Martin and EPSRC
SB: EPSRC

Outline

1. SAT, max-SAT, frustration, non-linear constraints, and sampling
2. Spectral Mapping with Ising Spins
   - Arbitrary max-SAT on the chimera graph
   - Aside: Another way to do parity checks
   - A better mapping
3. Applications
   - Classical communications: turbo code decoder
   - Particle simulation

Penalty formalism for SAT

Hamiltonian to implement clauses:

$$H(a) = \sum_i Pen(\{a_i^{(l)}\})$$

$$a_i^{(l)} \in \{a_1, a_2, a_2, ...\} \cup \{\neg a_1, \neg a_2, \neg a_3, ...\}$$

$$Pen(\{a^{(l)}\}) \begin{cases} \geq g & a_i^{(l)} = 0, \ \forall i \\ = 0 & \text{otherwise.} \end{cases}$$

- Violating any clause gives an energy of at least g
- zero energy state satisfies all clauses
- Lowest energy state not meaningful if cannot satisfy simultaneously

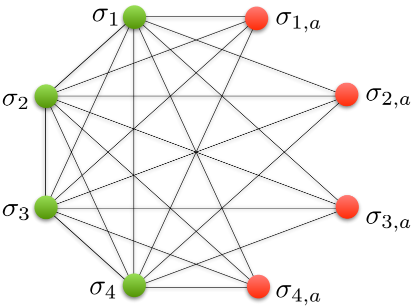
Alternate approach: penalize every clause violation equally

$$Spec(\{a^{(l)}\}) = \begin{cases} g & a_i^{(l)} = 0, \ \forall i \\ 0 & \text{otherwise.} \end{cases}$$

$$H(a) = \sum_i Spec(\{a_i^{(l)}\}) = N(a)\, g$$

- $N(a)$ is the number of clauses violated by bitstring a
- minimum energy bitstring satisfies the maximum number of clauses
- Boltzmann distribution can be used for maximum entropy inference $\rightarrow$ will come back to this

Making clauses using the Ising model (arXiv:1604.00651)



- couple all logical bits to each other and to all ancillas
- bias ancillas so that ancilla 1 will flip if 1 or more logical bits are up, 2 if 2 etc...
- ancilla couplings and couplings between bits cancel, fields on ancillas can assign arbitrary penalties
- single bit sector corresponds to clause on previous slide, can also create parity checks ($\oplus$ clauses) arXiv:1603.09521

simple example

Logical bit values	Ancilla values	E
1111	0000	0
0111, 1011, 1101, 1110	0001	0
0011, 0101, 0110, 1001, 1010, 1100	0011	0
1000, 0100, 0010, 0001	0111	0
0000	1111	g

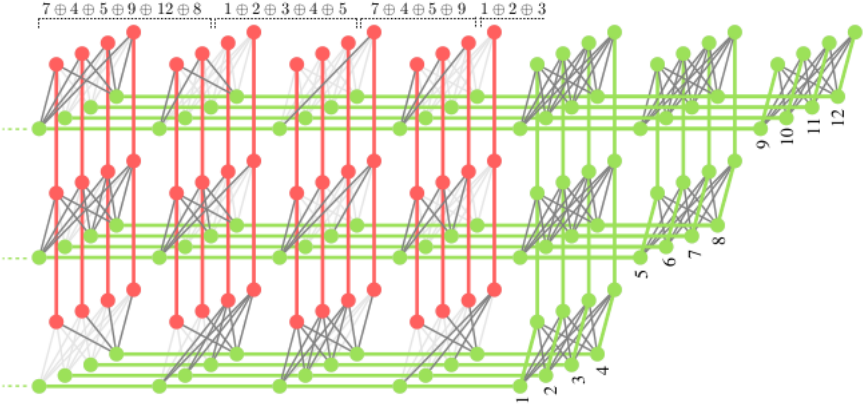
- Example in which applying a field on the last ancilla applies an OR clause $q_1 \vee q_2 \vee q_3 \vee q_4$
- Applying to first ancilla will instead apply AND $q_1 \wedge q_2 \wedge q_3 \wedge q_4$
- Alternating ancilla fields can apply XOR $q_1 \oplus q_2 \oplus q_3 \oplus q_4$
- see: arXiv:1604.00651 for Hamiltonian details

Quick Aside: Non-linear Constraints

Logical bit values	Ancilla values	E
1111	0000	E_4
0111, 1011, 1101, 1110	0001	E_3
0011, 0101, 0110, 1001, 1010, 1100	0011	E_2
1000, 0100, 0010, 0001	0111	E_1
0000	1111	E_0

- Energies can be assigned arbitrarily: can implement non-linear constraints on the number of ones

max-SAT on the chimera



- Use Choi complete graph minor embedding plus rows of ancillas to implement clauses arXiv:1604.00651

Drawbacks of this technique

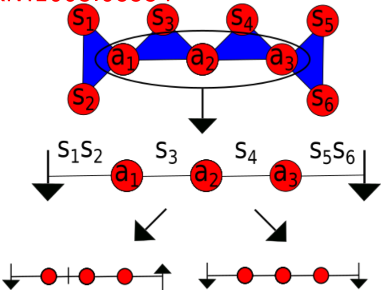
Conceptually nice to demonstrate direct mapping but....

- high connectivity makes mapping inefficient
- some clauses, ex. $(a_1 \vee a_2) \wedge (a_3 \vee a_4) \wedge \dots$ require an exponentially growing number of ancillas

I will show how both these problems can be addressed after a brief (1-slide) aside

one slide aside: another way of implementing parity checks

Proposed by other authors in arXiv:1604.02359 , alternate formulation in arXiv:1603.08554

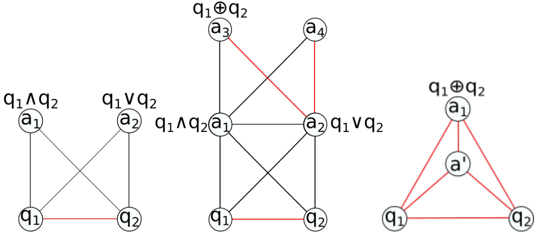


Triangles $\rightarrow$ effective 3 body coupling gadgets, s $\rightarrow$ logical bit,s a $\rightarrow$ ancillas

- parity of $\{s\}$ is even $\rightarrow$ no domain wall (unique), odd $\rightarrow$ domain wall (degenerate)
- degeneracy can be removed by weakening any of the 3 local gadgets, $\rightarrow$ necessary for thermal sampling

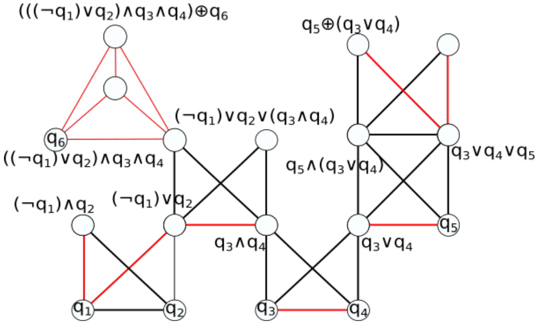
Ancilla representations for AND ($\wedge$), OR ($\vee$), and XOR ($\oplus$) on 2 bits

Using the methods of arXiv:1604.00651, single 'indicator' bit corresponds to result of any of these operations



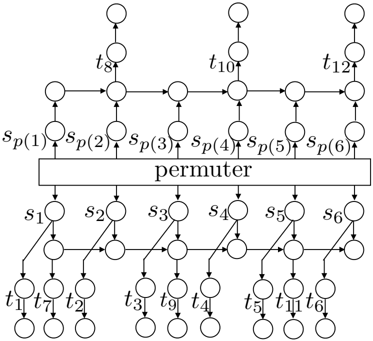
- $\wedge$ and $\vee$ 'mark' single states, so single bits act as 'indicators' for these clauses
- second copy of gadget Hamiltonian needed to create single bit corresponding to $\oplus$,
- alternatively 3 bit gadget acts as indicator for $\oplus$

Chaining 2 bit Hamiltonians to make more complex clauses



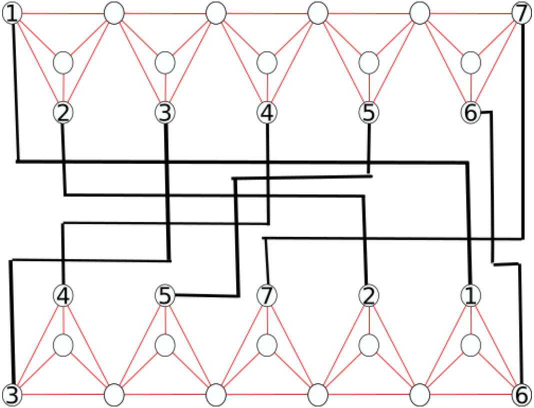
- Apply gadget to 'indicator' ancillas to create more complex clauses
- Fields on indicators enforce clauses
- Any clause which can be efficiently written using $\wedge$, $\vee$, $\oplus$, $\neg$, and parenthesis can be implemented efficiently
- Does not require full connectivity

Example: Classical Decoding, Turbo code implementation



- Interleaved convolution code: apply strings of $\oplus$ of even length on bits $(s_1 \oplus s_2)$, $(s_1 \oplus s_2 \oplus s_3 \oplus s_4)$ ...
- odd length strings of $\oplus$ applied to random permutation $(s_{p(1)})$, $(s_{p(1)} \oplus s_{p(2)} \oplus s_{p(3)})$ ...
- approaches Shanon limit for large block length

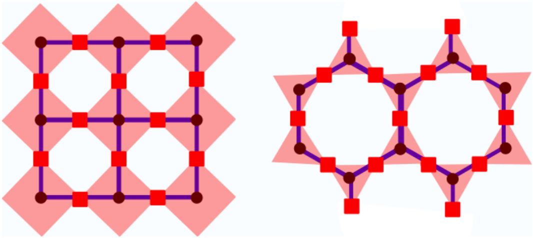
Turbo code decoder using minor embedding



- Black lines → embedding bonds
- red lines → ancilla couplings
- note similarity to ideas presented in arXiv:1603.08554v2

Application: Particle Simulation

Higher dimensional analogue of 1-D domain walls using high locality operators, red squares $\rightarrow$ spins, pink polyhedra $\rightarrow$ couplers, circles $\rightarrow$ particle sites, lines $\rightarrow$ allowed hops



$$H_{particle} = -\frac{m}{2}\sum_i \prod_{j\in d_i} \sigma_j^z - \tau \sum_i \sigma_i^z - \Delta \sum_i \sigma_i^x$$

Can this Hamiltonian be realized perturbatively using our gadget?

Realizing particle simulation Hamiltonian perturbatively

Using our gadget for each coupler

What works:

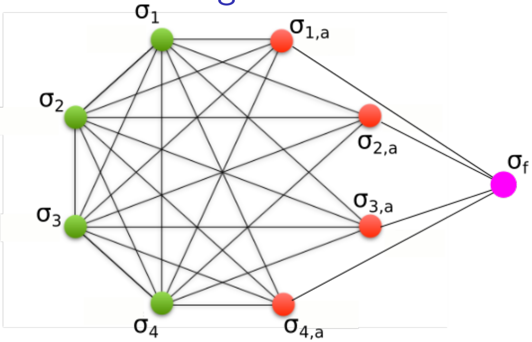
- All logical bit flips occur at 3rd order in perturbation theory (flip 2 ancillas plus bit)
- All transitions go through intermediate states with the same energy
- $\therefore$ bit flips (σ_i^x) realized perturbatively with no modifications

What Doesn't:

- Second order fluctuations are not the same for every logical state $\rightarrow$ some energetically favored over others

I will show how to rectify this on next slide

Matching fluctuation strengths with additional ancillas



'fluctuation control' ancillas (f), coupled to ancillas

$$H_{fc} = \sum_i (J_{i,f}(-\sigma^z_{i,a} + \sigma^z_{i,a}\sigma^z_f) + |J_{i,f}|\sigma^z_f) + h^0_f \sigma^z_f$$

- $h^0_f > 0$ no effect on Ising coupling
- $\{J_{i,f}\}$ tuned to control fluctuations
- more than 1 such ancilla can be added

Conclusions

- Alternate method for mapping problems based on max-SAT formalism
- Individual elements claimed together to realize any clause which can be written efficiently
- Example applications:
  - Classical message decoding: Turbo Code
  - Particle simulation: realize Hamiltonian perturabtively

Quick Plug for Some of my Other Work

Modernizing Quantum annealing using Local Search:
arXiv:1606.06833