Quadratization in Pseudo-Boolean Optimization and Ground-State Quantum Computing

Richard Tanburn,[1,*] Nicholas Chancellor,[2,†] and Nikesh S. Dattani[1,3,4,‡]

[1]*Oxford University, Department of Mathematics*
[2]*Durham University, Department of Physics*
[3]*McMaster University, Department of Chemistry and Chemical Biology*
[4]*Harvard-Smithsonian Center for Astrophysics*

A collaborative, evolving, open review paper on k-local to 2-local transformations (quadratizations) in classical computing, quantum annealing, and universal adiabatic quantum computing.

When optimizing discrete functions, it is often easier when the function is qudaratic than if it is of higher degree. But notice that the cubic and quadratic funcitons:

$$b_1b_2 + b_2b_3 + b_3b_4 - 4b_1b_2b_3 \quad \text{(cubic)}, \tag{1}$$
$$b_1b_2 + b_2b_3 + b_3b_4 + 4b_1 - 4b_1b_2 - 4b_1b_3 \quad \text{(quadratic)}, \tag{2}$$

where each b_i can either be 0 or 1, both never go below the value of -1, and all minima have the form $(b_1, b_2, b_3, b_4) = (1, 1, 1, b_4)$. Therefore if we are interested in the ground state of a discrete function of degree k, we may optimize either function and get exactly the same result. **Part I** gives more than 10 different ways to do this.

The binary variables b_i can be either of the eigenvalues of the matrix b below, which is related to the Pauli z matrix by $z = 2b - \mathbb{1}$. Other Pauli matrices are listed below:

$$b \equiv \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \; z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \; x \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \; y \equiv \begin{pmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix}, \; \mathbb{1} \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{3}$$

Any Hermitian 2×2 matrix can be written as a linear combination of the Pauli matrices $z, x, y, \mathbb{1}$, so we can therefore describe the Hamiltonian of any number of spin-1/2 particles by a function of Pauli matrices acting on each particle, for instance:

$$s_1s_2 + s_2s_3 + s_3s_4 - 4s_1s_2s_3 \quad \text{(cubic)}, \tag{4}$$

where each spin s_i can be any of the Pauli matrices, and the coefficients tell us about the strengths of couplings between these particles. The Schrödinger equation tells us that the eigenvalues of the Hamiltonian are the allowed energy levels and their eigenvectors (wavefunctions) are the corresponding physical states. More generally these do not have to be spins but can be any type of qubits, and we can encode the solution to *any* problem in the ground state of a Hamiltonian, then solve the problem by finding the lowest energy state of the physical system (this is called adiabatic quantum computing).

Two-body physical interactions occur more naturally than many-body interactions so **Parts II-III** give more than 10 different ways to quadratize genreal Hamiltonians (some of these methods do not even require the s_i to be 2×2 matrices, meaning that we can have types of qudits that are not qubits).

The optimization problems of Eqs. (1)-(2) are specific cases of Eq. (4) where only z matrices are present.

Part I

Diagonal Hamiltonians (pseudo-Boolean functions)

I. METHODS THAT INTRODUCE ZERO AUXILIARY VARIABLES

A. Deduction Reduction (Deduc-reduc)

Summary

We look for *deductions* (e.g. $b_1 b_2 = 0$) that hold at the global minimum. These can be found by *a priori* knowledge of the given problem, or by enumerating solutions of a small subset of the variables. We can then substitute high-order terms using the low-order terms of the deduction, and add on a penalty term to preserve the ground states [1].

Cost

- 0 auxiliary variables needed.
- The computational cost of the search for deductions is difficult to estimate. The approximate worst-case complexity is $\mathcal{O}(n^{d+1}2^m)$ where m is the number of variables in a 'small' problem, n is the total number of qubits and d is the maximum degree of deductions we are searching for. We suggest $10 \leq m \leq 20$, so that a small problem involves checking roughly $1,000$ to $1,000,000$ states, and $d = 2$. See the appendix for more details.

Pros

- No auxiliary variables needed.

Cons

- When deductions cannot be determined naturally (as in the Ramsey number determination problem, see Example IX), deductions need to be found by 'brute force', which scales exponentially with respect to m. For highly connected systems (systems with a large number of non-zero α_{ij} coefficients), the value of m required to find even one deduction can be prohibitively large.

Example

Consider the Hamiltonian:

$$H_{4\text{–local}} = b_1 b_2 (10 + b_3 + b_3 b_4) + b_1 (b_3 - 3) + b_2 (b_3 - 2b_3 - b_4 - 1) + F(x_3, x_4, x_5, \ldots, x_N) \quad (5)$$

where F is any polynomial in b_i for $i \geq 3$.
Since the 10 coefficient of $b_1 b_2$ is greater than the sum of all of the other coefficients involving b_1 or b_2, it must be the case that $b_1 b_2 = 0$. Specifically, for the 4 assignments of (b_3, b_4), we see that $b_1 b_2 = 0$ at every minimum of $H_{4\text{–local}} - F$.

Using deduc-reduc we have:

$$H_{2\text{–local}} = 12 b_1 b_2 + b_1 (b_3 - 3) + b_2 (b_3 - 2b_3 - b_4 - 1) + F(x_3, x_4, x_5, \ldots, x_N) \quad (6)$$

which has the same global minima as $H_{4\text{–local}}$ but one fewer quartic and one fewer cubic term.

Bibliography

- Original paper, with application to integer factorization: [1].

B. ELC Reduction

Summary

An Excludable Local Configuration (ELC) is a partial assignment of variables that make it impossible to achieve the minimum. We can therefore add a term that corresponds to the energy of this ELC without changing the solution to the minimization problem. In practice we can eliminate all monomials with a variable in which a variable is set to 0, and reduce any variable set to 1. Given a general Hamiltonian we can try to find ELCs by enumerating solutions of a small subset of variables in the problem [2].

Cost

- 0 auxiliary variables needed.
- For n qubits there is $\binom{n}{m}$ ways to choose m of them and 2^m assignments of these m variables, therefore $\mathcal{O}\left(2^m\binom{n}{m}\right)$ operations required to enumerate all possible cases for all possible subsets of size $m \leq n$ variables.
- Approximate methods exist which have been shown to be much faster and give good approximations to the global minimum [2].

Pros

- No auxiliary variables needed.

Cons

- No known way to find ELCs except by 'brute force', which scales exponentially with respect to m.
- ELCs do not always exist.

Example

Consider the Hamiltonian:

$$H_{3\text{-local}} = b_1b_2 + b_2b_3 + b_3b_4 - 4b_1b_2b_3. \tag{7}$$

If $b_1b_2b_3 = 0$, no assignment of our variables will we be able to reach a lower energy than if $b_1b_2b_3 = 1$. Hence this gives us 12 ELCs, and one example is $(b_1, b_2, b_3) = (1, 0, 0)$ which we can use to form the polynomial:

$$H_{2\text{-local}} = H_{3\text{-local}} + 4b_1(1 - b_2)(1 - b_3) \tag{8}$$

$$= b_1b_2 + b_2b_3 + b_3b_4 + 4b_1 - 4b_1b_2 - 4b_1b_3. \tag{9}$$

In both cases Eqs. (7) and (9), the only global minima occur when $b_1b_2b_3 = 1$.

Bibliography

- Original paper and application to computerized image denoising: [2].

C. Groebner Bases

Summary

Given a set of polynomials, a Groebner basis is another set of polynomials that have exactly the same zeros. The advantage of a Groebner basis is it has nicer algebraic properties than the original equations, in particular they tend to have smaller degree polynomials. The algorithms for calculating Groebner bases are generalizations of Euclid's algorithm for the polynomial greatest common divisor.

Work has been done in the field of 'Boolean Groebner bases', but while the variables are Boolean the coefficients of the functions are in $\mathbb{F}_2$ rather than $\mathbb{Q}$.

Cost

- 0 auxiliary variables needed.
- $\mathcal{O}\left(2^{2^n}\right)$ in general, $\mathcal{O}(d^{n^2})$ if the zeros of the equations form a set of discrete points, where d is the degree of the polynomial and n is the number of variables [3].

Pros

- No auxiliary variables needed.
- General method, which can be used for other rings, fields or types of variables.

Cons

- Best algorithms for finding Groebner bases scale double exponentially in n.
- Only works for Hamiltonians whose minimization corresponds to solving systems of discrete equations (RICHARD, why is this the ONLY case?).

Example

Consider the following pair of equations:

$$b_1b_2b_3b_4 + b_1b_3 + b_2b_4 - b_3 = b_1 + b_1b_2 + b_3 - 2 = 0. \tag{10}$$

Feeding these to Mathematica's `GroebnerBasis` function, along with the binarizing $b_1(b_1 - 1) = \ldots = b_4(b_4 - 1) = 0$ constraints, gives a Groebner basis:

$$\{b_4b_3 - b_4, b_2 + b_3 - 1, b_1 - 1\}. \tag{11}$$

From this we can immediately read off the solutions $b_1 = 1$, $b_2 = 1 - b_3$ and reduce the problem to $b_3b_4 - b_4 = 0$. Solving this gives a final solution set of: $(b_1, b_2, b_3, b_4) = (1,0,1,0), (1,0,1,1), (1,1,0,0)$.

Bibliography

- Reduction and embedding of factorizations of all bi-primes up to $200,000$: [4].

D. Split Reduction

Summary

It has been shown in [5] that, if multiple runs of a minimization algorithm is permitted, it is possible to reduce a lot of the problem by conditioning on the most connected variables. We call each of these operations a *split*.

Cost

Exponential in the number of splits, as the number of problems to solve doubles with every split.

Pros

- This method can be applied to any problem and can be very effective on problems with a few very connected variables.

Cons

- Exponential cost in the worst case.

Example

Consider the simple objective function

$$H = 1 + b_1 b_2 b_5 + b_1 b_6 b_7 b_8 + b_3 b_4 b_8 - b_1 b_3 b_4. \tag{12}$$

In order to quadratize H, we first have to choose a variable to split over. In this case b_1 is the obvious choice since it is present in the most terms and contributes to the quartic term.

We then obtain two different problems:

$$H_0 = 1 + b_3 b_4 b_8 \tag{13}$$
$$H_1 = 1 + b_2 b_5 + b_6 b_7 b_8 + b_3 b_4 b_8 - b_3 b_4. \tag{14}$$

At this point, we could split H_0 again and solve it entirely, or use a qubit we saved in the previous split to quadratize our only problem.

To solve H_1, we can split again on b_8, resulting in problems:

$$H_{1,0} = 1 + b_2 b_5 + b_6 b_7 \tag{15}$$
$$H_{1,1} = 1 + b_2 b_5 + b_3 b_4. \tag{16}$$

Now both of these problems are quadratic. Hence we have reduced our original, hard problem into 3 easy problems, requiring only 2 extra runs of our minimization algorithm, and without needing any auxiliary variables.

Bibliography

- Original paper and application to Ramsey number calculation: [5].

II. METHODS THAT INTRODUCE AUXILIARY VARIABLES TO QUADRATIZE A SINGLE TERM

A. Negative Term Reduction

Summary

For a negative term $-b_1 b_2 ... b_k$, introduce a single auxiliary variable b_a and make the substitution:

$$-b_1 b_2 \dots b_k = \min_{b_a} \left((k - 1 - \sum_i b_i) b_a \right). \tag{17}$$

Cost

- 1 auxiliary variable for each k-local term.

Pros

- All resulting quadratic terms are submodular (have negative coefficients).
- Can reduce arbitrary order terms with only 1 auxiliary.

Cons

- Only works for negative terms.

Example

$$H_{6-\text{local}} = -2 b_1 b_2 b_3 b_4 b_5 b_6 + b_5 b_6, \tag{18}$$

has a unique minimum energy of -1 when all $b_i = 1$.

$$H_{2-\text{local}} = 2 \left(5 b_a - b_1 b_a - b_2 b_a - b_3 b_a - b_4 b_a - b_5 b_a - b_6 b_a \right) + b_5 b_6 \tag{19}$$

has the same unique minimum energy, and it occurs at the same place (all $b_i = 1$), with $b_a = 1$.

Bibliography

- Original paper: [6].
- Discussion: [7], [8].

B. Positive Term Reduction

Summary

By considering the negated literals $\bar{b}_i = 1 - b_i$, we recursively apply the previous method to $b_1 b_2 \dots b_k = -\bar{b}_1 b_2 \dots b_k + b_2 b_3 \dots b_k$. The final identity is:

$$b_1 b_2 \dots b_k = \min_{b_a} \left(\sum_{i=1}^{k-2} b_{a_i} \left(k - i - 1 + b_i - \sum_{j=i+1}^{k} b_j \right) \right) + b_{k-1} b_k \tag{20}$$

Cost

- $k - 2$ auxiliary variables for each k-local term.

Pros

- Works for positive monomials.

Cons

- $k - 1$ non-submodular quadratic terms.

Example

$$b_1 b_2 b_3 b_4 = \min_{b_a} b_{a_1} (2 + b_1 - b_2 - b_3 - b_4) + b_{a_2} (1 + b_2 - b_3 - b_4) + b_3 b_4 \tag{21}$$

Bibliography

- Summary: [9].

C. Ishikawa's Symmetric Reduction (Positive Term Reduction)

Summary

This method rewrites a positive monomial using symmetric polynomials, so all possible quadratic terms are produced and they are all non-submodular:

$$b_1 \dots b_k = \min_{b_{a_1},\dots,b_{a_{n_k}}} \left(\sum_{i=1}^{n_k} b_{a_i} \left(c_{i,d} \left(-\sum_{j=1}^{k} b_j + 2i \right) - 1 \right) + \sum_{i<j} b_i b_j \right) \tag{22}$$

where $n_k = \left\lfloor \frac{k-1}{2} \right\rfloor$ and $c_{i,k} = \begin{cases} 1, & i = n_d \text{ and } k \text{ is odd,} \\ 2, & \text{else.} \end{cases}$

Cost

- $\left\lfloor \frac{k-1}{2} \right\rfloor$ auxiliary variables for each k-order term
- $\mathcal{O}(kt)$ for a k-local Hamiltonian with t terms.

Pros

- Works for positive monomials. About half as many auxiliary variables for each k-order term as the previous method.

Cons

- $\mathcal{O}(k^2)$ quadratic terms are created, which may make chimerization more costly.
- $\frac{k(k-1)}{2}$ non-submodular terms.
- Worse than the previous method for quartics, with respect to submodularity.

Example

$$b_1 b_2 b_3 b_4 = \min_{b_a} (3 - 2b_1 - 2b_2 - 2b_3 - 2b_4) b_a + b_1 b_2 + b_1 b_3 + b_1 b_4 + b_2 b_3 + b_2 b_4 + b_3 b_4 \tag{23}$$

Bibliography

- Original paper and application to image denoising: [7].

D. Reduction by Minimum Selection

Summary

This method can also be used to rewrite positive cubic terms in terms of 6 quadratic terms.

The identity is given by:

$$b_1 b_2 b_3 = \min_{b_a} -\left(b_a + b_1 + b_2 + b_3\right) + b_a\left(b_1 + b_2 + b_3\right) + b_1 b_2 + b_2 b_3 + b_3 b_1 \tag{24}$$

Cost

1 auxiliary qubit per positive cubic term.

Pros

- Works on positive monomials.

Cons

- Introduces all 6 possible non-submodular terms.

Bibliography

- Original introduction by Kolmogorov and Zabih: [10].

E. Asymmetric Reduction

Summary

Similar to other methods of reducing one term, this method can reduce a positive cubic monomial into quadratic terms using only one auxiliary variable, while introducing fewer non-submodular terms than the symmetric version.

The identity is given by:

$$b_1 b_2 b_3 = \min_{b_a} \left(b_a - b_2 b_a - b_3 b_a + b_1 b_a + b_2 b_3 \right) \tag{25}$$

$$= \min_{b_a} \left(b_a - b_1 b_a - b_3 b_a + b_2 b_a + b_1 b_3 \right) \tag{26}$$

$$= \min_{b_a} \left(b_a - b_1 b_a - b_2 b_a + b_3 b_a + b_1 b_2 \right) \tag{27}$$

$$\tag{28}$$

Cost

1 auxiliary qubit per positive cubic term.

Pros

- Works on positive monomials.
- Fewer non-submodular terms than Ishikawa Reduction.

Cons

- Only been shown to work for cubics.

Bibliography

- Original paper and application to computer vision: [11].

F. Bit flipping

Summary

For any variable b, we can consider the negation $\bar{b} = 1 - b$. The process of exchanging b for $\bar{b}$ is called *flipping*. Using bit-flipping, an arbitrary function in n variables can be represented using at most $2^{(n-2)}(n-3)+1$ variables, though this is a gross overestimate.

Can be used in many different ways:

1. Flipping positive terms and using II A, recursively;

2. For $\alpha < 0$, we can reduce $\alpha\bar{b}_1\bar{b}_2...\bar{b}_k$ very efficiently to submodular form using II A. A generalized version exists for arbitrary combinations of flips in the monomial which makes reduction entirely submodular [7];

3. When we have quadratized we can minimize the number of non-submodular terms by flipping.

4. We can make use of both b_i and $\bar{b}_i$ in the same Hamiltonian by adding on a sufficiently large penalty term: $\lambda(b_i + \bar{b}_i - 1)^2 = \lambda(1 + 2b_i\bar{b}_i - b_i - \bar{b}_i)$. This is similar to the ideas in reduction by substitution or deduc-reduc. In this way, given a quadratic in n variables we can make sure it only has at most n nonsubmodular terms if we are willing to use the extra n negation variables as well (so we have $2n$ variables in total).

Cost

- None, as replacing b_i with it's negation $\bar{b}_i$ costs nothing except a trivial symbolic expansion.

Pros

- Cheap and effective way of improving submodularity.

- Can be used to combine terms in clever ways, making other methods more efficient.

Cons

- Unless the form of the Hamiltonian is known, spotting these 'factorizations' using negations is difficult.

- We need an auxiliary variable for each b_i for which we also want to use $\bar{b}_i$ in the same Hamiltonian.

Example

By bit-flipping b_2 and b_4, i.e. substituting $b_2 = 1 - \bar{b}_2$ and $b_4 = 1 - \bar{b}_4$, we see that:

$$H = 3b_1b_2 + b_2b_3 + 2b_1b_4 - 4b_2b_4 \tag{29}$$
$$= -3b_1\bar{b}_2 - \bar{b}_2b_3 - 2b_1\bar{b}_4 - \bar{b}_2\bar{b}_4 + 5b_1 + b_3 + 4\bar{b}_2 + 4\bar{b}_4 - 4. \tag{30}$$

The first expression is highly non-submodular while the second is entirely submodular.

Bibliography

- Original paper: [7].

III. METHODS THAT INTRODUCE AUXILIARY VARIABLES TO QUADRATIZE MULTIPLE TERMS WITH THE SAME AUXILIARIES

A. Reduction by Substitution

Summary

Pick a variable pair (b_i, b_j) and substitute $b_i b_j$ with a new auxiliary variable $b_{a_{ij}}$. Enforce equality in the ground states by adding some scalar multiple of the penalty $P = b_i b_j - 2b_i b_{a_{ij}} - 2b_i b_{a_{ij}} + 3b_{a_{ij}}$ or similar. Since P > 0 if and only if $b_{a_{ij}} \neq b_i b_j$, the minimum of the new $(k-1)$-local function will satisfy $b_{a_{ij}} = b_i b_j$), which means that at the minimum, we have precisely the original function. Repeat $(k-2)$ times for each k-local term and the resulting function will be 2-local.

Cost

- 1 auxiliary variable per reduction.
- At most $k\,t$ auxiliary variables for a k-local Hamiltonian of t terms, but usually fewer.

Pros

- Variable can be used across the entire Hamiltonian, reducing many terms at once.
- Very easy to implement.
- Reproduces not only the ground state, but the full spectrum.

Cons

- Inefficient for single terms as it introduces many auxiliary variables compared to Ishikawa reduction, for example.
- Introduces quadratic terms with large positive coefficients, making them highly non-submodular.
- Determining optimal substitutions can be expensive.

Example

We pick the pair (b_1, b_2) and combine.

$$b_1 b_2 b_3 + b_1 b_2 b_4 \mapsto b_3 b_a + b_4 b_a + b_1 b_2 - 2b_1 b_a - 2b_1 b_a + 3b_a \tag{31}$$

Bibliography

- Original paper: [12]
- Re-discovered in the context of diagonal quantum Hamiltonians: [14].
- Used in: [15, 16].

B. FGBZ Reduction (Fix-Gruber-Boros-Zabih)

Summary

Here we consider a set C of variables which occur in multiple monomials throughout the Hamiltonian. Each application 'rips out' this common component from each term [13][9].

Let $\mathcal{H}$ be a set of monomials, where $C \subseteq H$ for each $H \in \mathcal{H}$ and each monomial H has a weight α_H. The algorithm comes in 2 parts: when all $\alpha_H > 0$ and when all $\alpha_H < 0$. Combining the 2 gives the final method:

1. $\alpha_H > 0$

$$\sum_{H\in\mathcal{H}} \alpha_H \prod_{j\in H} b_j = \min_{b_a} \left(\sum_{H\in\mathcal{H}} \alpha_H \right) b_a \prod_{j\in C} b_j + \sum_{H\in\mathcal{H}} \alpha_H (1-b_a) \prod_{j\in H\setminus C} b_j \tag{32}$$

2. $\alpha_H < 0$

$$\sum_{H\in\mathcal{H}} \alpha_H \prod_{j\in H} b_j = \min_{b_a} \sum_{H\in\mathcal{H}} \alpha_H \left(1 - \prod_{j\in C} b_j - \prod_{j\in H\setminus C} b_j \right) b_a \tag{33}$$

Cost

- One auxiliary variable per application.
- In combination with II A, it can be used to make an algorithm which can reduce t positive monomials of degree d in n variables using $n + t(d-1)$ auxiliary variables in the worst case.

Pros

- Can reduce the connectivity of a Hamiltonian, as it breaks interactions between variables.

Cons

- $\alpha_H > 0$ method converts positive terms into negative ones of same order rather than reducing them, though these can then be reduced more easily.
- $\alpha_H < 0$ method only works for $|C| > 1$, and cannot quadratize cubic terms.

Example

First let $C = b_1$ and use the positive weight version:

$$\begin{aligned} b_1b_2b_3 + b_1b_2b_4 &\mapsto 2b_{a_1}b_1 + (1-b_{a_1})b_2b_3 + (1-b_{a_1})b_2b_4 && (34)\\ &= 2b_{a_1}b_1 + b_2b_3 + b_2b_4 - b_{a_1}b_2b_3 - b_{a_1}b_2b_4 && (35)\end{aligned}$$

now we can use II A:

$$\begin{aligned} -b_{a_1}b_2b_3 - b_{a_1}b_2b_4 &\mapsto 2b_{a_2} - b_{a_1}b_{a_2} - b_{a_2}b_2 - b_{a_2}b_3 + 2b_{a_2} - b_{a_1}b_{a_2} - b_{a_2}b_2 - b_{a_2}b_4 && (36)\\ &= 4b_{a_2} - 2b_{a_1}b_{a_2} - 2b_{a_2}b_2 - b_{a_2}b_3 - b_{a_2}b_4. && (37)\end{aligned}$$

Bibliography

- Original paper and application to image denoising: [13].

IV. METHODS THAT REPRODUCE THE FULL SPECTRUM

These techniques will transform k-local functions to 2-local functions that not only have the same ground state as the k-local function, but also the entire input/output spectrum of the k-local functions will be preserved in the low-lying energy space of the corresponding 2-local function (which also has higher energy states due to the auxiliary variables added).

A. Flag Based SAT Mapping

Summary

This method is very similar to recursive order reduction, but uses gadgets to produce separate 3-SAT clauses which allow variables which 'flag' the state of pairs of other variables

Cost

- Varies, but generally higher than other methods

Pros

- Highly general and therefore conducive to proofs.

Cons

- Designed for generality rather than efficiency.

Example

To create a system which maps $(z_1 = -1) \vee (z_2 = -1) \vee (z_3 = -1)$ (up to an energy shift), we use the following gadget:

$$\begin{aligned} H_{3\text{–SAT}} &= (1 - \sum_{i=1}^{3} z_{a_i})(1 - \frac{1}{2}\sum_{i=1}^{3} z_i) + \frac{1}{2}\sum_{i<j}^{3} z_{a_i} z_{a_j} - 1 \\ &= -\sum_{i=1}^{3} z_{a_i} - \frac{1}{2}\sum_{i=1}^{3} z_i + \frac{1}{2}\sum_{i<j}^{3} z_{a_i} z_{a_j} + \frac{1}{2}\sum_{i=1}^{3} z_i z_{a_i}. \end{aligned} \tag{38}$$

Implementing $\lambda H_{3\text{–SAT}}$, creates a situation where z_3 is a 'flag' for $z_1 = -1$ and $z_2 = -1$ in other words z_3 is constrained to be -1 in the low energy manifold iff $z_1 = -1$ and $z_2 = -1$ and $+1$ otherwise. It follows from the universality of $3 - \mathrm{SAT}$ that these 'flag' clauses can be combined to map any spin Hamiltonian.

Bibliography

- Paper showing the universality of the Ising spin models: [17].

B. Chained Three Body Parity Operators

Summary

Goal: Guarantee that $z_{a_k} = z_i z_j$ can be chained together to make large product terms consisting of z. The penalty term $P(z_{a_k}, z_i, z_j) = \mp z_{a_k} z_i z_j$ guarantees that $z_{a_k} = \pm z_i z_j$. The 3-local term can be made from gadgets.

This method was originally only used to reproduce the ground state of high locality terms, but states of the "wrong" parity ($q_k = \mp z_i z_j$) will all have the same energy as well, so it reproduces the full spectrum.

Cost

- The best known gadget for a 3-local Ising term uses one auxilliary qubit. Based on this gadget an $n \geq 4$ body Ising term can be made using $3 + 2(n-4)$ auxilliary qubits.

Pros

- Natural transmon implementation [18].
- Chain like structure means that long range connectivity not required.

Cons

- Does not preserve degeneracy, ground state will retain orginal degeneracy, but excited states will have degeneracy multipled by $n-3$
- Not very symmetric.

Example

$$H_{\text{5-local}} = z_1 z_2 z_3 z_4 z_5 - 2 z_1 z_4 z_5 + 3 z_1 z_3 - z_5 + 9 \tag{39}$$

The full spectrum of $\pm z_1 z_2 z_3$ is reproduced by:

$$P_{\pm}(z_1, z_2, z_3; \lambda) = \lambda \left(z_1 z_2 + z_2 z_3 + z_3 z_1 + 2 z_a (z_1 + z_2 + z_3) \right) \mp (z_1 + z_2 + z_3 + 2 z_a). \tag{40}$$

Using three copies of this 3-local gadget as a building block, and using two additional auxilliary variables, the spectrum of the 5-local term $z_1 z_2 z_3 z_4 z_5$ can be reproduced by the following Hamiltonian

$$H_{\text{2-local}} = P_+(z_1, z_2, z_{a_1}; \lambda) + P_+(z_{a_1}, z_3, z_{a_2}; \lambda) + P_+(z_{a_2}, z_4, z_5; \lambda). \tag{41}$$

Bibliography

- Original proposal with transmon implementation: [18].

C. Symmetry Based Mappings

Summary

Auxilliary qubits can be made to "count" the number of logical qubits in the 1 configuration. By applying single qubit terms to the auxilliary qubits, the spectrum of *any* permutation symmetric Hamiltonian can be reproduced.

Cost

- For a k local coupler requires k auxilliary qubits.

Pros

- Natural flux qubit implementation [19].
- Single gadget can reproduce any permutation symmetric spectrum.
- High degree of symmetry means this method is natural for some kinds of quantum simulations [20].

Cons

- Requires coupling between all logical qubits and from all logical qubits to all auxilliary qubits.
- Requires single body terms of increasing strength as k is increased.

Example

A 4 qubit gadget guarantees that the number of auxillary bits in the 0 state is equal to the number of logical bits in the 1 state

$$H_{4-\text{count}} = \sum_{i=2}^{4}\sum_{j=1}^{i-1} z_i z_j - \frac{1}{2}\sum_{i=1}^{4} z_i + \sum_{i=1}^{4}\sum_{j=1}^{4} z_i z_{a_j} + \frac{1}{2}\left(5z_{a_1} + z_{a_2} - 3\,z_{a_3} - 7z_{a_4}\right). \tag{42}$$

To replicate the spectrum of $z_1 z_2 z_3 z_4$, we add

$$H_{4-\text{local}} = z_{a_1} - z_{a_2} + z_{a_3} - z_{a_4} + \lambda H_{4-\text{count}}, \tag{43}$$

where λ is a large number.

For the spectrum of $b_1 b_2 b_3 b_4$, we implement,

$$H_{4-\text{local}} = -\frac{1}{2} z_{a_4} + \lambda H_{4-\text{count}}. \tag{44}$$

Bibliography

- Paper on flux qubit implementation: [19]
- Paper on max-k-sat mapping: [21]
- Talk including use in quantum simulation: [20]

Part II

Hamiltonians quadratic in z and linear in x (Transverse Field Ising Hamiltonians)

The Ising Hamiltonian with a transverse field in the x direction is possible to implement in hardware:

$$H = \sum_i \left(\alpha_i^{(z)} z_i + \alpha_i^{(x)} x_i\right) + \sum_{ij} \left(\alpha_{ij}^{(zz)} z_i z_j\right). \tag{45}$$

D. ZZZ-TI-CBBK: Transvese Ising from ZZZ, by Cao, Babbush, Biamonte, and Kais (2015)

There is only one reduction in the literature for reducing a Hamiltonian term to the transverse Ising Hamiltonian, and it works on 3-local zzz terms, by introducing an auxiliary qubit with label a:

$$\alpha z_i z_j z_k \to \alpha^I + \alpha_i^z z_i + \alpha_j^z z_j + \alpha_k^z z_k + \alpha_a^z z_a + \alpha_a^x x_a + \alpha_{ia}^{zz} z_i z_a + \alpha_{ja}^{zz} z_j z_a + \alpha_{ka}^{zz} z_k z_a \tag{46}$$

$$\begin{aligned}
\alpha^I &= \tfrac{1}{2}\left(\Delta + \left(\tfrac{\alpha}{6}\right)^{2/5} \Delta^{3/5}\right) \\
\alpha_i^z &= -\tfrac{1}{2}\left(\left(\tfrac{7\alpha}{6} + \left(\tfrac{\alpha}{6}\right)^{3/5} \Delta^{2/5}\right) - \left(\tfrac{\alpha\Delta^4}{6}\right)^{1/5}\right) \\
\alpha_j^z &= \alpha_i^{(z)} \\
\alpha_k^z &= \alpha_i^{(z)} \\
\alpha_a^z &= \tfrac{1}{2}\left(\Delta - \left(\tfrac{\alpha}{6}\right)^{2/5} \Delta^{3/5}\right) \\
\alpha_a^x &= \left(\tfrac{\alpha\Delta^4}{6}\right)^{1/5} \\
\alpha_{ia}^{zz} &= -\tfrac{1}{2}\left(\left(\tfrac{7\alpha}{6} + \left(\tfrac{\alpha}{6}\right)^{3/5} \Delta^{2/5}\right) + \left(\tfrac{\alpha\Delta^4}{6}\right)^{1/5}\right) \\
\alpha_{ja}^{zz} &= \alpha_{ia}^{(zz)} \\
\alpha_{ka}^{zz} &= \alpha_{ja}^{(zz)}
\end{aligned}$$

Including all coefficients and factorizing, we get:

$$\alpha z_i z_j z_k \to \left(\Delta + \frac{\alpha\Delta^4}{6}^{1/5} (z_i + z_j + z_k)\right)\left(\frac{1 - z_a}{2}\right) + \frac{\alpha\Delta^4}{6}^{1/5} x_a \tag{47}$$

$$+ \left(\left(\frac{\alpha}{6}\right)^{2/5} \Delta^{3/5} - \left(\frac{7\alpha}{6} + \left(\frac{\alpha}{6}\right)^{3/5} \Delta^{2/5}\right)(z_i + z_j + z_k)\right)\left(\frac{1 + z_a}{2}\right) \tag{48}$$

The low-lying spectrum (eigenvalues ***and*** eigenvectors) of the right side of Eq. (46) will match those of the left side to within a spectral error of ϵ as long as $\Delta = \mathcal{O}\left(\epsilon^{-5}\right)$.

Cost

- 1 auxiliary qubit
- 8 auxiliary terms not proportional to $\mathbb{1}$.

Example

E. 1B1-TI: 1-by-1 gadget for $ZZ\ldots Z \rightarrow$ Transverse Ising (Present Work)

Cost

- $k-2$ auxiliary qubits
- auxiliary terms not proportional to $\mathbb{1}$.

Example

F. SD-TI: Sub-division gadget for $ZZ\ldots Z \rightarrow$ Transverse Ising (Present Work)

Cost

- $k/2$

Example

Part III

General Quantum Hamiltonians

The most general time-independent Hamiltonian for n qubits acted on only by Pauli operators is:

$$H = \alpha I + \sum_{i}^{n} \left(\alpha_i^{(z)} z_i + \alpha_i^{(x)} x_i + \alpha_i^{(y)} y_i \right) + \tag{49}$$

$$\sum_{ij}^{n} \left(a_{ij}^{(zz)} z_i z_j + a_{ij}^{(zx)} z_i x_j + a_{ij}^{(zy)} z_i y_j + a_{ij}^{(xz)} x_i z_j + a_{ij}^{(xx)} x_i x_j + a_{ij}^{(xy)} x_i y_j + a_{ij}^{(yz)} y_i z_j + a_{ij}^{(yx)} y_i x_j + a_{ij}^{(yy)} y_i y_j \right) + \tag{50}$$

$$\sum_{ijk}^{n} \left(a_{ijk}^{(zzz)} z_i z_j z_k + a_{ijk}^{(zzx)} z_i z_j x_k + a_{ijk}^{(zzy)} z_i z_j y_k + a_{ijk}^{(zxz)} z_i x_j z_k + \cdots + a_{ijk}^{(yyy)} y_i y_j y_k \right) + \tag{51}$$

$$\sum_{ijkl\cdots n}^{n} \left(a_{ijkl\cdots n}^{(zzzz\cdots z)} z_i z_j z_k z_l \cdots z_n + a_{ijkl\cdots n}^{(zzz\cdots x)} z_i z_j z_k \cdots x_n + a_{ijkl\cdots n}^{(zzz\cdots y)} z_i z_j z_k \cdots y_n + a_{ijkl\cdots n}^{(zz\cdots xz)} z_i z_j \cdots x_{n-1} z_n + \cdots + a_{ijkl\cdots n}^{(yyy\cdots y)} y_i y_j y_k \cdots y_n \right) \tag{52}$$

We wish to find a 2-local Hamiltonian that is equivalent to Eq. (49) in terms of reproducing certain desired properties (e.g. same ground state(s), same ground eigenvalue(s), same full eigenspectrum, etc.) to within the desired precision.

V. 1-BY-1 GADGETS

A 1B1 gadget allows k-local terms to be quadratized one step at a time, where at each step the term's order is reduced by at most one. In each step, a k-local term is reduced to $(k-1)$-local, contrary to SD (sub-division) gadgets which can reduce k-local terms to $(1/2)$-local in one step.

A. 1B1-KKR (Kempe, Kitaev, Regev, 2006)

Summary

Group all k-local terms together and express their sum as a sum of products of commuting matrices s_{ij}:

$$H_{k-\text{local}} = \sum_i \prod_j^k s_{ij} + H_{(k-1)-\text{local}}. \tag{53}$$

Define auxiliary qubits labeled by a_{ij} and make the transformation:

$$\sum_i \prod_j^k s_{ij} \mapsto \frac{\Delta}{4} \left(9 - \sum_i \left(\sum_j z_{a_{ij}} \right)^2 - \sum_j z_{a_{ij}}^2 + \frac{1}{\sqrt[3]{6\Delta}} x_{a_{ij}} \right) + \Lambda. \tag{54}$$

The result will be a $(k-1)$-local Hamiltonian with the same low-lying spectrum as $H_{k-\text{local}}$ to within ϵ as long as $\Delta = \Theta(\epsilon^{-3})$.

Cost

- The number of auxiliary qubits needed is the number of s_{ij} matrices in Eq. (55).
- $\Delta = \Theta(\epsilon^{-3})$

B. 1B1-OT (Oliviera & Terhal, 2008)

Summary

We wish to reduce the k-local term:

$$H_{k-\text{local}} = \alpha \prod_j^k s_j. \tag{55}$$

Define one auxiliary qubit labeled by a and make the transformation:

$$H_{k-\text{local}} \to -\left(\frac{\alpha}{2}\right)^{1/3} \Delta^{2(1-r)} s_k \left(\frac{1-z_a}{2}\right) + \left(\frac{\alpha}{2}\right)^{1/3} \frac{\Delta^r}{\sqrt{2}} \left(s_{k-1} - s_{k-2}\right) x_a \tag{56}$$

$$+ \frac{1}{2}\left(\frac{\alpha}{2}\right)^{2/3} \left(\Delta^{r-1} s_{k-1} + \text{sgn}(\alpha)\sqrt{2}\Delta^{-1/4} \prod_j^{k-2} s_j\right)^2 + \frac{\alpha}{4}\left(1 + 2\text{sgn}^2\alpha\Delta^{3/2-2r}\right) s_k. \tag{57}$$

The result will be a $(k-1)$-local Hamiltonian with the same low-lying spectrum as $H_{k-\text{local}}$ to within ϵ as long as $\Delta = \Theta\left(\epsilon^{-3}\right)$.

Cost

- Only 1 auxiliary qubit.
- $\Delta = \Theta\left(\epsilon^{-3}\right)$

Example

Bibliography

C. 1B1-CBBK (Cao, Babbush, Biamonte, Kais, 2015)

Define one auxiliary qubit labeled by a and make the transformation:

$$H_{k-\text{local}} \to \left(\Delta + \left(\frac{\alpha}{2}\right)^{3/2} \Delta^{1/2} s_k\right)\left(\frac{1-z_a}{2}\right) \tag{58}$$

$$- \frac{\alpha^{2/3}}{2}\left(1+\text{sgn}^2\alpha\right)\left((2\alpha)^{2/3}\,\text{sgn}^2\alpha + \alpha^{1/3} s_k - \sqrt[3]{2}\Delta^{1/2}\right)\left(\frac{1+z_a}{2}\right) \tag{59}$$

$$+ \left(\frac{\alpha}{2}\right)^{1/3} \Delta^{3/4}\left(\prod_j^{k-2} s_j - \text{sgn}(\alpha) s_{k-1}\right) x_a + \text{sgn}(\alpha)\sqrt[3]{2}\alpha^{2/3}\left(\Delta^{1/2}+\Delta^{3/2}\right)\prod_j^{k-1} s_j. \tag{60}$$

The result is $(k-1)$-local and its low-lying spectrum is the same as that of $H_{k-\text{local}}$ when Δ is large enough.

VI. SUBDIVISION GADGETS

Instead of recursively reducing k-local to $(k-1)$-local one reduction at a time, we can reduce k-local terms to $(k/2)$-local terms directly for even k, or to $(k+1)/2$-local terms directly for odd k. Since when k is odd we can add an identity operator to the k-local term to make it even, we will assume in the following that k is even, in order to avoid having to write floor and ceiling functions.

A. SD-OT (Oliviera & Terhal, 2008)

Summary

For any k-local term, we can subdivide it into a product of two $(k/2)$ –local terms, with an auxiliary qubit labeled a:

$$H_{k-\text{local}} \to \Delta \frac{1-z_a}{2} + \sqrt{\frac{\Delta}{2}} \left(s_k - \prod_j^{k-1} s_j \right) x_a \tag{61}$$

Cost

- $\Delta = \frac{\|H_{(k-1)-\text{local}+}\|\epsilon^2}{2}$.
- $\Delta = \Theta\left(\epsilon^{-2}\right)$.

Pros

- only one qubit to reduce k to $\lceil k/2 \rceil + 1$

Cons

- Only beneficial for $k \geq 5$.

Example

Bibliography

B. SD-CBBK (Cao, Babbush, Biamonte, Kais 2015)

Summary

For any k-local term, we can subdivide it into a product of two $(k/2)$ –local terms:

$$H_{k-\text{local}} = \alpha H_{1,(k/2)-\text{local}} H_{2,(k/2)-\text{local}} + H_{(k-1)-\text{local}}. \tag{62}$$

Define one qubit a and make the following Hamiltonian is $(k/2)$-local:

$$\Delta \frac{1-z_a}{2} + |\alpha| \frac{1+z_a}{2} + \sqrt{|\alpha|\Delta/2}\left(\text{sgn}(\alpha) H_{1,(k/2-\text{local})} - H_{2,(k/2-\text{local})}\right) x_a \tag{63}$$

The result is a $(k/2)$-local Hamiltonian with the same low-lying spectrum as $H_{k-\text{local}}$ for large enough Δ. The disadvantage is that Δ has to be larger.

Cost

- $\Delta \geq \left(\frac{2|\alpha|}{\epsilon} + 1\right)\left(|\alpha| + \epsilon + 22\|H_{(k-1)-\text{local}}\right)$

Pros

- only one qubit to reduce k to $\lceil k/2 \rceil + 1$

Cons

- Only beneficial for $k \geq 5$.

Example

Bibliography

C. Duan et al (2011)

Summary

Cost

Example

Bibliography

D. Cao and Nagaj (2015)

Summary

Cost

Pros

- Smaller control precision requiremments.

Cons

- More auxiliary qubits.
- More terms.

Example

Bibliography

E. BOA (Babbush, O'Gorman, Aspuru-Guzik, 2013)

Summary

Cost

Example

Bibliography

VII. DIRECT REDUCTION

Here we do not reduce k by one order at a time (1B1 reduction) or by $k/2$ at a time (SD reduction), but we directly reduce k-local terms to 2-local terms.

A. DR-JF (Jordan & Farhi)

Summary

Express a sum of k-local terms as a sum of products of Pauli matrices s_{ij}, and define k auxiliary qubits laelled by a_{ij} for each term i, and make the transformmation:

$$\sum_i \alpha_i \prod_j^k s_{ij} \to \frac{-k(-\epsilon)^k}{(k-1)!} \sum_i \frac{1}{2} \left(k^2 - \sum_{jl}^k z_{a_{ij}} z_{a_{il}} \right) + \epsilon \left(\alpha_i s_{i1} x_{i1} + \sum_j^k s_{ij} x_{ij} \right) - f(\epsilon)\Pi, \tag{64}$$

for some polynomial $f(\lambda)$. The result is a 2-local Hamiltonian with the same low-lying spectrum to within ϵ^{k+1} for sufficiently smmall ϵ.

Cost

- Number of auxiliary qubits is tk for t terms.
- Unknown requirement for ϵ.

Pros

- All done in one step, so easier to implement than 1B1 and SD gadgets.

Cons

- Requires 2 more auxiliary qubits per term than 1B1-KKR.
- Unknown polynomial $f(\lambda)$

Example

Bibliography

- Original paper [22].

VIII. NON-PERTURBATIVE EMBEDDINGS

A. NP-N (Nagaj, 2010)

Summary

Cost

Example

Bibliography

- Original paper: [**?**].

B. NP-OY (Ocko & Yoshida, 2011)

Summary

Cost

Example

Bibliography

- Original paper: [**?**].

C. NP-SJ (Subasi & Jarzynski, 2016)

Summary

The authors describe the method as a "transfer" of non-locality from the original Hamiltonian, into a unitary transforation operator.

Cost

Pros

-
-

Cons

- Can introduce new 3-local terms.
-

Example

Bibliography

- Original paper: [23].

Part IV

Appendix

IX. FURTHER EXAMPLES

Example Here we show how deductions can arise naturally from the Ramsey number problem. Consider $\mathcal{R}(4,3)$ with $N = 4$ nodes. Consider a Hamiltonian:

$$H = (1 - z_{12})(1 - z_{13})(1 - z_{23}) + \ldots + (1 - z_{23})(1 - z_{24})(1 - z_{34}) + z_{12}z_{13}z_{14}z_{23}z_{24}z_{34}. \quad (65)$$

See [5] for full details of how we arrive at this Hamiltonian.

Since we are assuming we have no 3-independent sets, we know that $(1 - z_{12})(1 - z_{13})(1 - z_{23}) = 0$, so $z_{12}z_{13}z_{23} = z_{12}z_{13} + z_{12}z_{23} + z_{13}z_{23} - z_{12} - z_{13} - z_{23} + 1$. This will be our deduction.

Using deduc-reduc we can substitute this into our 6-local term to get:

$$H = 2(1 - z_{12})(1 - z_{13})(1 - z_{23}) + \ldots + (1 - z_{23})(1 - z_{24})(1 - z_{34}) + \quad (66)$$

$$z_{14}z_{24}z_{34}(z_{12}z_{13} + z_{12}z_{23} + z_{13}z_{23} - z_{12} - z_{13} - z_{23} + 1). \quad (67)$$

We could repeat this process to remove all 5- and 4-local terms without adding any auxiliary qubits. Note in this case the error terms added by deduc-reduc already appear in our Hamiltonian.

ACKNOWLEDGMENTS

* richard.tanburn@gmail.com
† nicholas.chancellor@gmail.com
‡ nik.dattani@gmail.com

[1] R. Tanburn, E. Okada, and N. Dattani, (2015), arXiv:1508.04816.
[2] H. Ishikawa, in *2014 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2014) pp. 1362–1369.
[3] M. Bardet, Algorithms Seminar , 85 (2002).
[4] R. Dridi and H. Alghassi, (2016), arXiv:1604.05796.
[5] E. Okada, R. Tanburn, and N. S. Dattani, , 5 (2015), arXiv:1508.07190.
[6] D. Freedman and P. Drineas, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 2 (IEEE, 2005) pp. 939–946.
[7] H. Ishikawa, IEEE Transactions on Pattern Analysis and Machine Intelligence **33**, 1234 (2011).
[8] M. Anthony, E. Boros, Y. Crama, and A. Gruber, (2015).
[9] E. Boros and A. Gruber, (2014), arXiv:1404.6538.
[10] V. Kolmogorov and R. Zabih, IEEE Transactions on Pattern Analysis and Machine Intelligence **26**, 147 (2004).
[11] A. C. Gallagher, D. Batra, and D. Parikh, in *CVPR 2011* (IEEE, 2011) pp. 1857–1864.
[12] I. G. Rosenberg, Cahiers du Centre d'Etudes de Recherche Operationnelle **17**, 71 (1975).
[13] A. Fix, A. Gruber, E. Boros, and R. Zabih, in *2011 International Conference on Computer Vision* (IEEE, 2011) pp. 1020–1027.
[14] J. D. Biamonte, Phys. Rev. A **77**, 52331 (2008).
[15] A. Perdomo, C. Truncik, I. Tubert-Brohman, G. Rose, and A. Aspuru-Guzik, Phys. Rev. A **78**, 12320 (2008).
[16] Z. Bian, F. Chudak, W. G. Macready, L. Clark, and F. Gaitan, Physical Review Letters **111**, 130505 (2013).
[17] G. De las Cuevas and T. S. Cubitt, Science **351**, 1180 (2016).
[18] M. Leib, P. Zoller, and W. Lechner, Quantum Science and Technology **1**, 15008 (2016).
[19] N. Chancellor, S. Zohren, and P. A. Warburton, npj Quantum Information **3**, 21 (2017).
[20] N. Chancellor, "Max-k-SAT, Multi-Body Frustration, & Multi-Body Sampling on a Two Local Ising System," AQC 2016 https://www.youtube.com/watch?v=aC-6hg_h3EA (2016).
[21] N. Chancellor, S. Zohren, P. Warburton, S. Benjamin, and S. Roberts, Scientific Reports **6** (2016), 10.1038/srep37107.
[22] S. P. Jordan and E. Farhi, Physical Review A **77**, 062329 (2008).
[23] Y. Subasi and C. Jarzynski, Physical Review A **94**, 012342 (2016).