Efficient Quadratic Ising Hamiltonian Generation with Qubit Reduction

Rahul Sarkar

Marco Pistoia

Patent idea summary

- We present a technique for automatic generation of Ising Hamiltonians supporting optimization problems with the following characteristics:
  - ***Polynomial*** objective function.
  - Equality and inequality ***polynomial*** constraints.
  - Integer variables which can take values in ***contiguous*** and ***non-contiguous finite sets***.

- In order to solve this class of problems we also introduce a combination of techniques to reduce the number of variables in the final Ising Hamiltonian. This is effectively a ***qubit reduction procedure***.
  - Casting the problem as a quadratic pseudo-boolean optimization problem.
  - Use of ***roof duality*** and ***extended roof duality*** techniques to automatically determine the optimal values of a subset of the variables – thus reducing the number of variables in the optimization problem.
  - This results in an ***automated, problem instance dependent qubit reduction procedure***.
  - This can be useful in the near term for fitting optimization problems on near term quantum computers.

Agenda

- Motivation
- Summary of invention
- Technical details
  - Representing integer variables
  - Transforming inequality constraints to equality constraints
  - Equivalent psuedo-boolean optimization problem
  - Quadratization to obtain quadratic pseudo-boolean optimization problem
  - Automatic determination of optimal values of some of the variables
- Comparison to prior state of the art

Agenda

- Motivation
- Summary of invention
- Technical details
  - Representing integer variables
  - Transforming inequality constraints to equality constraints
  - Equivalent psuedo-boolean optimization problem
  - Quadratization to obtain quadratic pseudo-boolean optimization problem
  - Automatic determination of optimal values of some of the variables
- Comparison to prior state of the art

Motivation: constraints and special kinds of integer variables

- Many real world integer programming problems involve equality and inequality constraints.

  Example

  - Traveling salesman problem: It contains both equality and inequality constraints when posed as a discrete optimization problem.
  - Job shop scheduling problem, knapsack problem, subset sum problem are other examples where the optimization problem involves equality / inequality constraints, and the variables are integers.

- In some problems, integer variables can take values in non-contiguous set of integers.

  Example

  - ***Semicontinuous integer variables***: These are variables that can take either the value of ***zero***, or integer values between some upper and lower bounds. Such variables are typically used to model situations, where a certain variable will either not be used, or when used must be in a specific range.
  - A typical situation: Suppose an airline company wants to decide whether to fly a plane on a route or not. They may decide that they will only fly the plane if the number of passengers is at least 10. If the maximum capacity of the plane is 100, then the revenue collected by the airline is either ***zero*** (if the plane does not fly), or ***ticket price * N*** , where N is an integer between 10-100.

Motivation: pseudo-boolean optimization

Computer Vision applications

- Many problems in computer vision involve optimization of **binary Markov random fields (MRFs)**.
- MRFs (also known as ***pseudo-boolean functions*** in the discrete optimization literature) are used to solve computer vision problems of the following kinds:
  - Image segmentation, Texture recognition, Super resolution / view synthesis
- In each case, one needs to minimize the MRF.
  - Historically a restricted class of MRFs were considered, for e.g. pairwise interactions only, which lead to quadratic pseudo-boolean optimization problems (with some exception where triple interactions were considered).
  - This restriction was imposed because of the difficulty of minimizing higher order pseudo-boolean functions (even the quadratic case is hard in general). Quadratic reductions of higher order problems were also in general avoided, because the reductions often led to non-submodular optimization.
  - But natural scenes are too rich to be well captured by nearest neighbor interactions.
  - ***Ability to optimize such higher order MRFs thus has potential applications in computer vision.***

Agenda

- Motivation
- Summary of invention
- Technical details
  - Representing integer variables
  - Transforming inequality constraints to equality constraints
  - Equivalent psuedo-boolean optimization problem
  - Quadratization to obtain quadratic pseudo-boolean optimization problem
  - Automatic determination of optimal values of some of the variables
- Comparison to prior state of the art

Problem statement

We describe a technique to formulate efficient Ising Hamiltonians for the following class of integer optimization problems, followed by an automated qubit reduction procedure, which can then be solved using quantum optimization algorithms.

$$\begin{aligned} \text{minimize} \quad & f(x_1, \dots, x_p) & \\ \text{subject to} \quad & x_i \in I_i \subset \mathbb{Z}, & i = 1, \dots, p \\ & c_j(x_1, \dots, x_p) = 0, & j = 1, \dots, m \\ & d_k(x_1, \dots, x_p) \leq 0, & k = 1, \dots, n. \end{aligned}$$

- The functions f, c_j, d_k are ***polynomials over*** $\mathbb{Z}$ in the integer valued variables $x_1, \dots, x_p$.
- The sets I_i are finite subsets of the integers. These subsets ***may not be contiguous sets of integers.***
  - For example $x_1 \in \{0,1,2\} \cup \{5,6,7,8\}$.

Outline of the method (1/2)

The method comprises of the following steps, which we briefly describe below. They are described in more detail later.

Step 1: Represent each integer variable as linear sums of binary variables. Depending on the representation, one may end up introducing additional equality constraints.

Step 2: Introduce additional slack variables to change all inequality constraints to equality constraints. The slack variables are integer variables, that are represented using the same techniques used in Step1. Depending on the method used, the representation of the integer variables may introduce additional equality constraints.

Outline of the method (2/2)

Step 3: Create an unconstrained pseudo-boolean optimization problem, by squaring and adding all the equality constraints, with a large weight to the objective function.

Step 4: Quadratize the pseudo-boolean optimization problem, using the best known current techniques that minimize the number of additional variables used. This gives an equivalent quadratic pseudo-boolean function (QPBF) to optimize. ***This is a qubit reduction step***.

Step 5: Reduce the number of variables in the QPBF using ***roof duality*** and ***extended roof duality*** techniques (which have polynomial runtime). This determines the values of a subset of the boolean variables. We thus obtain a new QPBF with fewer number of boolean variables. ***This is a qubit reduction step***.

The output of Step 5 is the final QPBF that is solved using a quantum optimization algorithm.

****Note: Step 5 can be applied to any quadratic Ising Hamiltonian (that the user may wish to solve using quantum optimization algorithms, like QAOA[2], and VQE[1]).**

Agenda

- Motivation
- Summary of invention
- Technical details
  - Representing integer variables
  - Transforming inequality constraints to equality constraints
  - Equivalent psuedo-boolean optimization problem
  - Quadratization to obtain quadratic pseudo-boolean optimization problem
  - Automatic determination of optimal values of some of the variables
- Comparison to prior state of the art

Step 1: Representing integer variables

- Integer variables can be of two types:
  - **Contiguous set valued.**
    For e.g. $x \in \{5,6,7,8\}$.
  - **Non-contiguous set valued.**
    For e.g. $y \in \{0,1,2\} \cup \{5,6,7,8\}$.

  Sets are assumed to be finite.

- **Methods to encode integer variables**

Contiguous set valued	Non-contiguous set valued
Log encoding One hot encoding *Discrete slack method* *Greedy decomposition method*	One hot encoding *Efficient encoding scheme*

Existing methods in literature.
Methods introduced in patent application draft 95762203 (Matsuo, Imamichi, Worner, Pistoia, Sarkar), 2019.

Equivalent problem after Step 1

After Step 1, all integer variables have been replaced with binary variables, but potentially new equality constraints have been added.

$$
\begin{array}{lll}
\text{minimize} & f(x_1, \ldots, x_{p'}) & \\
\text{subject to} & x_i \in \{0,1\}, & i = 1, \ldots, p' \\
& c_j(x_1, \ldots, x_{p'}) = 0, & j = 1, \ldots, m' \\
& d_k(x_1, \ldots, x_{p'}) \leq 0, & k = 1, \ldots, n'.
\end{array}
$$

- The functions f, c_j, d_k are ***polynomials*** in the binary variables $x_1, \ldots, x_{p'}$.

Agenda

- Motivation
- Summary of invention
- Technical details
  - Representing integer variables
  - Transforming inequality constraints to equality constraints
  - Equivalent psuedo-boolean optimization problem
  - Quadratization to obtain quadratic pseudo-boolean optimization problem
  - Automatic determination of optimal values of some of the variables
- Comparison to prior state of the art

Step 2: Convert inequality constraints to equality constraints

- Since we can not directly write Ising Hamiltonians with inequality constraints, we need to transform inequality constraints into equality constraints. Assuming finite precision, we can always convert an inequality constraint to an equality constraint with an additional integer variable without loss of generality.
  - $f(x) \le c \rightarrow f(x) + s = c$
    - $f(x) = \sum_i a_i x_i + \sum_i b_i y_i$
      - $\underline{f} \le f(x) \le \overline{f}$
    - x_i:binary variable, y_i:integer variable, a_i, b_i:coefficients, c: constant
    - s: contiguous set valued integer variable with range $\left[0, c - \underline{f}\right]$
  - Similarly: $f(x) \ge c \rightarrow f(x) - s = c$
    - s: contiguous set valued integer variable with range $\left[0, \overline{f} - c\right]$
- By applying above methods to an additional integer variable s (as in the above expressions), we can transform an inequality constraint into an equality constraint with less additional binary variables compared to applying existing methods to the additional integer variable s.

* The technique in this slide is based on the patent P201809806 [4].

Equivalent problem after Step 2

After Step 2, all inequality constraints have been replaced with equality constraints, and new binary variables have been added to the problem.

$$
\begin{aligned}
\text{minimize} \quad & f(x_1, \dots, x_{p''}) & \\
\text{subject to} \quad & x_i \in \{0,1\}, & i = 1, \dots, p'' \\
& c_j(x_1, \dots, x_{p''}) = 0, & j = 1, \dots, m''.
\end{aligned}
$$

- The functions f, c_j are ***polynomials*** in the binary variables $x_1, \dots, x_{p''}$. Such polynomials are also known as ***pseudo-boolean functions (PBF),*** and we will use this terminology in the next slides.

Agenda

- Motivation
- Summary of invention
- Technical details
  - Representing integer variables
  - Transforming inequality constraints to equality constraints
  - Equivalent psuedo-boolean optimization problem
  - Quadratization to obtain quadratic pseudo-boolean optimization problem
  - Automatic determination of optimal values of some of the variables
- Comparison to prior state of the art

Step 3: Create an unconstrained optimization problem

- The next step is to convert the optimization problem into an equivalent unconstrained optimization problem.
  - Square the equality constraints.
  - Add the squared equality constraints to the objective function with a large weight parameter λ.
  - Such a parameter λ always exists.
  - No extra variables are added to the problem.

The new objective function is given by

$$g(x_1, \ldots, x_{p''}) = f(x_1, \ldots, x_{p''}) + \lambda \sum_{j=1}^{m''} \left(c_j(x_1, \ldots, x_{p''})\right)^2 .$$

Equivalent problem after Step 3

After Step 3, we have an unconstrained pseudo-boolean optimization problem.

$$\begin{array}{lll} \text{minimize} & g(x_1, \ldots, x_{p''}) & \\ \text{subject to} & x_i \in \{0,1\}, & i = 1, \ldots, p''. \end{array}$$

- The function g is a ***pseudo-boolean function.***

Agenda

- Motivation
- Summary of invention
- Technical details
  - Representing integer variables
  - Transforming inequality constraints to equality constraints
  - Equivalent psuedo-boolean optimization problem
  - Quadratization to obtain quadratic pseudo-boolean optimization problem
  - Automatic determination of optimal values of some of the variables
- Comparison to prior state of the art

Step 4: Quadratization of the pseudo-boolean function

- The next step involves ***quadratization*** of the pseudo-boolean function. At a high level it achieves the following (explained in detail in the next slides):
  - Creates an equivalent quadratic pseudo-boolean function (QPBF).
  - Introduces additional binary variables into the problem.

- **Quadratization**

  Given the pseudo-boolean function $g(x_1, \ldots, x_{p''})$, its quadratization is a new ***quadratic pseudo-boolean function*** $h(x_1, \ldots, x_{p''}, w_1, \ldots, w_s)$ which has the following property:

$$g(x_1, \ldots, x_{p''}) = \min_{w_1, \ldots, w_s} h(x_1, \ldots, x_{p''}, w_1, \ldots, w_s).$$

  Thus minimization of $g(x_1, \ldots, x_{p''})$ over $x_1, \ldots, x_{p''}$, is the same as the minimization of $h(x_1, \ldots, x_{p''}, w_1, \ldots, w_s)$ over all the variables $x_1, \ldots, x_{p''}, w_1, \ldots, w_s$.

Step 4: Quadratization procedure

- The quadratization strategy that we follow is ***term-by-term quadratization***.
  - Each ***monomial*** term of the pseudo-boolean function is ***quadratized separately***.
  - Use a ***different*** set of new binary variables to quadratize each monomial term.
  - Add all the monomial quadratizations to get the quadratization of the original pseudo-boolean function.
  - The total number of new binary variables introduced into the problem is the sum of the number of binary variables introduced to quadratize each monomial term.

- Example
  - Suppose we want to quadratize the function : $g(x_1, x_2, x_3, x_4) = x_1 + 2x_2x_3 - x_1x_2x_3 - 3x_1x_2x_3x_4$.
  - The first two terms x_1 and $2x_2x_3$ are already quadratized.
  - Let $u(x_1, x_2, x_3, w_1)$ be a quadratization of $-x_1x_2x_3$ (only 1 extra binary variable is needed).
  - Let $v(x_1, x_2, x_3, x_4, w_2)$ be a quadratization of $-x_1x_2x_3x_4$ (only 1 extra binary variable is needed).
  - Then the quadratization of $g(x_1, x_2, x_3, x_4)$ is given by the quadratic pseudo-boolean function $h(x_1, x_2, x_3, x_4, w_1, w_2) = x_1 + 2x_2x_3 + u(x_1, x_2, x_3, w_1) + 3v(x_1, x_2, x_3, x_4, w_2)$.

Step 4: Quadratization of negative monomials

- We use the ***currently best known quadratization schemes*** to quadratize each monomial.

- **Negative monomials**

  Suppose we have the negative monomial $-x_1 \ldots x_n$ which we want to quadratize. Let us define the set $S = \{x_1, \ldots, x_n\}$. Then a quadratization of the monomial is given in [6] by

$$-x_1 \ldots x_n = \min_{w \in \{0,1\}} \left(|S| - 1 - \sum_{x_i \in S} x_i \right) w \, .$$

- This quadratization scheme introduces 1 extra binary variable for each negative monomial term.

Step 4: Quadratization of positive monomials

- **Positive monomials**

Suppose we have the positive monomial $x_1 \dots x_n$ which we want to quadratize. First define the functions

$$G_1 = \sum_{i=1}^{n} x_i \;, G_2 = G_1(G_1 - 1)/2 = \sum_{i,j=1, j>i}^{n} x_i x_j \,.$$

Then the quadratization of the monomial is given in [7] by

$$x_1 \dots x_n = G_2 + \left(\min_{w_1,\dots,w_k \in \{0,1\}} \sum_{i=1}^{k} w_i \left(c_{i,n}(-G_1 + 2i) - 1 \right) \right),$$

where $k = floor\left(\frac{n-1}{2}\right)$, and $c_{i,n} = \begin{cases} 1 & if\ n\ is\ odd, and\ i = k \\ 2 & otherwise. \end{cases}$

- This quadratization scheme introduces new binary variables. The number of these extra variables is approximately ½ the number of original variables in the positive monomial.

Equivalent problem after Step 4

After Step 4, we have an unconstrained pseudo-boolean optimization problem.

$$\begin{array}{lll} \text{minimize} & h(x_1, \ldots, x_{p''}, w_1, \ldots, w_s) & \\ \text{subject to} & x_i \in \{0,1\}, & i = 1, \ldots, p'' \\ & w_j \in \{0,1\}, & j = 1, \ldots, s. \end{array}$$

- The function h is a ***quadratic pseudo-boolean function (QPBF).***

Agenda

- Motivation
- Summary of invention
- Technical details
  - Representing integer variables
  - Transforming inequality constraints to equality constraints
  - Equivalent psuedo-boolean optimization problem
  - Quadratization to obtain quadratic pseudo-boolean optimization problem
  - Automatic determination of optimal values of some of the variables
- Comparison to prior state of the art

Step 5: Determination of optimal values for subset of variables

- The last step involves running a polynomial time algorithm to find the optimal values of a subset of the variables.
  - The tools used are techniques known as ***roof duality*** and ***extended roof duality***.
  - The process does not add any new binary variables to the problem.

- The following steps are carried out under Step5:
  - **Step 5a**: Finding optimal values of a subset of the variables using roof duality.
    - Run the ***quadratic pseudo-boolean optimization (QPBO) algorithm***.
    - This involves solving a flow optimization problem on a graph.
  - **Step 5b**: Find optimal values of some more variables that could not be determined by Step 5a / reduce the number of variables by identifying which variables take the same value in an optimal solution.
    - Probing.

Step 5a: Using roof duality to determine optimal values of a subset of variables

- Roof duality technique[8]
  - The quadratic pseudo-boolean optimization problem can be put in one-to-one correspondence with a capacitative network, where the capacities of the edges are related to the coefficients of the QPBF.
  - A max flow problem is solved using this capacitative network (see [8] for details)
  - The solution of this max flow problem gives rise to a cut, or partition of the nodes of the capacitative network.
  - This cut is used to infer which variables have been fixed to their optimal values using the roof duality technique.

- Roof duality gives the first subset of variables which have been fixed to their optimal values.

- These variables can be eliminated from the problem.

Equivalent problem after Step 5

After Step 5, we have an unconstrained pseudo-boolean optimization problem, with less number of binary variables than after Step 4.

$$\begin{aligned} &\text{minimize} && \tilde{h}(y_1, \dots, y_q) \\ &\text{subject to} && y_i \in \{0,1\}, \qquad i = 1, \dots, q. \end{aligned}$$

- The function $\tilde{h}$ is also a ***quadratic pseudo-boolean function (QPBF).***

Step 5b: Using extended roof duality to eliminate more variables from the problem

- Extended roof duality technique[9]
  - In this work we only use the technique of probing, which is part of the extended roof duality techniques.
  - A variable that was unlabeled in Step 5a is selected.
  - Two instances of the optimization problem is created, one by fixing this variable to 0 and another by fixing this variable to 1.
  - In both cases roof duality (or Step 5a) is run.
  - The variables that get assigned their optimal values in both cases (i.e. intersection of the variables) can be reliably eliminated from the problem.
  - This process is repeated till no more variables are eliminated (see [9] for details).

- After Step 5b, we have the final optimization problem.

An example of qubit reduction

Suppose that after Step 4, we have the following unconstrained quadratic pseudo-boolean optimization problem, or suppose that the user obtained this problem (by some other means) which he/she wants to solve using some quantum optimization algorithm.

$$\begin{aligned} &\text{minimize} \quad a\,x_1 + b\,x_2x_3 + c\,x_1x_2 + d\,x_3x_4 \\ &\text{subject to} \quad x_i \in \{0,1\}, \qquad i = 1,2,3,4 \end{aligned}$$

a, b, c, d are integers

We show how roof duality (Step 5a) will work for example, and reduce qubit requirements.

- We construct the capacitative network consisting of the nodes $x_1, x_2, x_3, x_4, \overline{x_1}, \overline{x_2}, \overline{x_3}, \overline{x_4}$, $x_0, \overline{x_0}$, where $x_0, \overline{x_0}$ are the ***source*** and ***sink*** nodes (see [8] for details).
- Suppose after solving the max-flow problem on this capacitative network, we extract the corresponding min-cut.
- Suppose the ***source side cut*** is $\{x_0, x_1, \overline{x_2}, x_3, \overline{x_3}\}$, and the ***sink side cut*** is $\{\overline{x_0}, \overline{x_1}, x_2, x_4, \overline{x_4}\}$.
- This gives the following partial assignment:
  - $x_1 = 0, x_2 = 1$. They are assigned as complements appear in different cuts.
  - x_3, x_4 are unassigned as variable and its complement appear in the same cut.
- The Ising Hamiltonian now simplifies to

$$\begin{aligned} &\text{minimize} \quad b\,x_3 + d\,x_3x_4 \\ &\text{subject to} \quad x_i \in \{0,1\}, \qquad i = 3,4 \end{aligned}$$

b, d are integers

- ***The number of qubits reduced is thus 2.***
- Exact number of variables that can be reduced depends on the problem instance.

Agenda

- Motivation
- Summary of invention
- Technical details
  - Representing integer variables
  - Transforming inequality constraints to equality constraints
  - Equivalent psuedo-boolean optimization problem
  - Quadratization to obtain quadratic pseudo-boolean optimization problem
  - Automatic determination of optimal values of some of the variables
- Comparison to prior state of the art

Comparisons with prior art

- We have referenced several existing patents applications (P201901382, P201809806, patent application draft 95762203), all of which combined address the following points:
  - How to handle binary and integer variables.
  - How to transform problems with ***linear*** equality and inequality constraints, to Ising Hamiltonians.
  - Deals with ***only quadratic*** objective functions.

- None of these existing techniques address more general polynomial constraints or polynomial objective functions of integer variables.

- Novel features of this patent application
  - How to incorporate ***polynomial constraints*** and ***polynomial objective function*** of the integer variables.
  - ***Qubit reduction procedure*** to get the final Ising Hamiltonian for quantum optimization.

Individual contributions

Rahul Sarkar and Marco Pistoia jointly contributed to the following:

- Procedure to formulate optimization problems with polynomial constraints and polynomial objective function as Ising Hamiltonians.
- The idea that quadratization techniques can be applied to reduce pseudo-boolean functions to QPBFs.
- Idea to reduce the number of variables in the optimization problem using roof duality and extended roof duality techniques, before solving the problem using quantum optimization algorithms, ***which lowers the qubit requirement costs***.

References

[1] Daochen Wang, et al., Accelerated Variational Quantum Eigensolver, Phys. Rev. Lett. 122, 140504

[2] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, A Quantum Approximate Optimization Algorithm, arxiv:1411.4028

[3] Atsushi Matsuo, Takashi Imamichi, Marco Pistoia, Automatic generation of Ising Hamiltonian for solving optimization problems in quantum computing, P201901382

[4] Stefan Wörner, Giacomo Nannicini, Panagiotis Barkoutsos, Ivano Tavernelli, System & Method to solve Mixed Integer Problems on a quantum computer, P201809806

[5] pyqubo -Integer, https://pyqubo.readthedocs.io/en/latest/reference/integer.html, Accessed on 17th June 2019

[6] Ishikawa, H., 2010. Transformation of general binary MRF minimization to the first-order case. *IEEE transactions on pattern analysis and machine intelligence*, *33*(6), pp.1234-1249.

[7] Freedman, D. and Drineas, P., 2005, June. Energy minimization via graph cuts: Settling what is possible. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (Vol. 2, pp. 939-946). IEEE.

[8] Boros, E., Hammer, P.L. and Tavares, G., 2006. Preprocessing of unconstrained quadratic binary optimization.

[9] Rother, C., Kolmogorov, V., Lempitsky, V. and Szummer, M., 2007, June. Optimizing binary MRFs via extended roof duality. In *2007 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-8). IEEE.