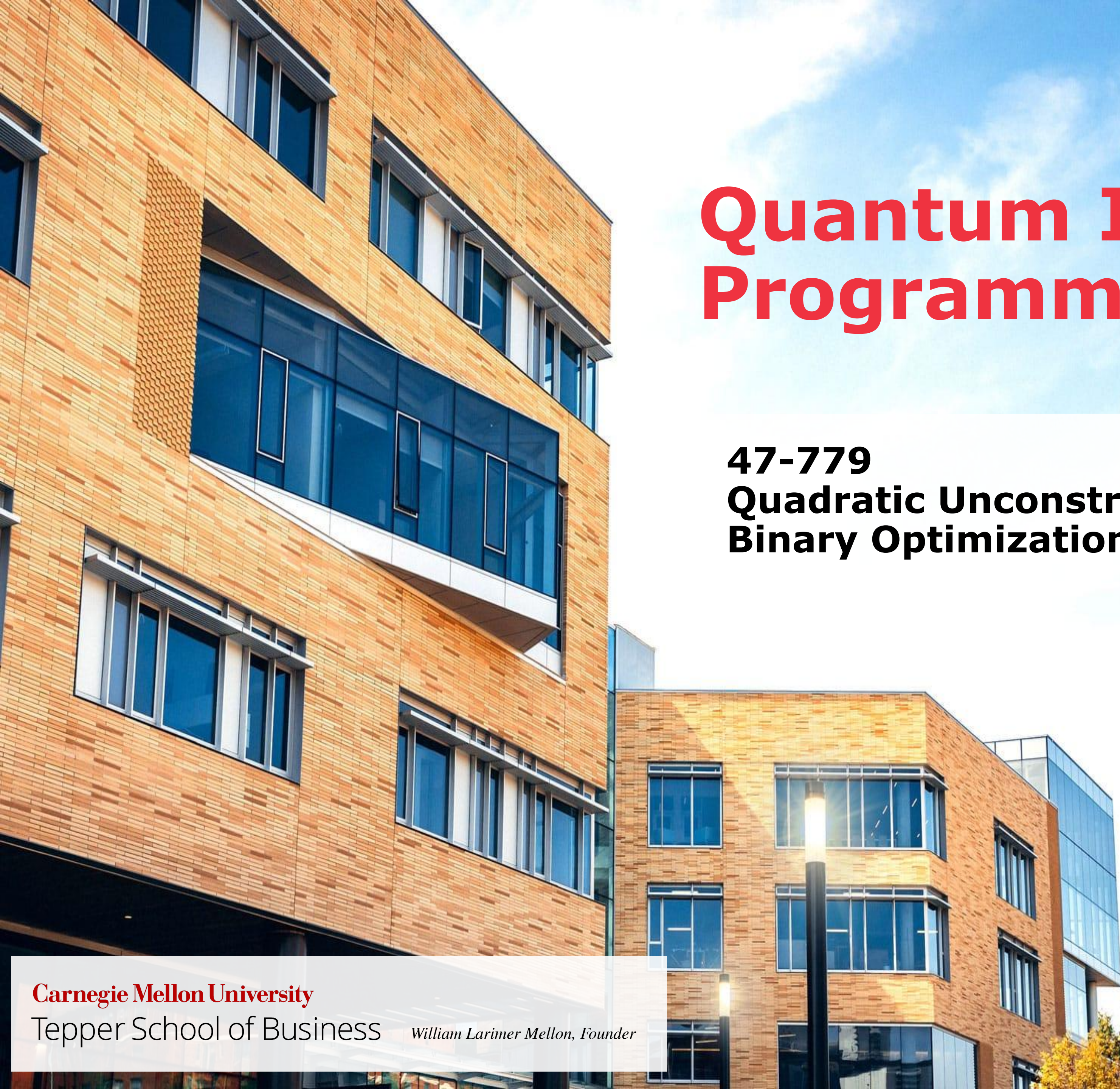


Quantum Integer Programming

47-779
Quadratic Unconstrained Binary Optimization (QUBO)

Carnegie Mellon University
Tepper School of Business
William Larimer Mellon, Founder

Quiz 3

Follow Lecture X and create your Amazon Braket Account!
If you already have an AWS account please send it over the chat, if not send us an email that you would like to use to register to it.

Besides, go and create a free account for D-Wave
https://cloud.dwavesys.com/leap/

And for IBM Quantum Experience
https://quantum-computing.ibm.com/login

Agenda

- QUBO Definition
- QUBO - Ising Model Mapping
- QUBO - Integer Programming model
  - QU*B*O: Binarization
  - Q*U*BO: Unconstraining
  - *Q*UBO: Quadratization
- Integer Programming as QUBO
- Coloring Example
- Integer Factorization

QUBO model

Quadratic Unconstrained Binary Optimization

$$\min_{\mathbf{x}\in\{0,1\}^n} \sum_{(ij)\in E(G)} x_i Q_{ij} x_j + \sum_{i\in V(G)} Q_{ii} x_i + c$$
$$= \min_{\mathbf{x}\in\{0,1\}^n} \mathbf{x}^\top \mathbf{Q}\mathbf{x} + c$$

Quadratic coefficient matrix $\mathbf{Q}$

- Can be either upper triangular or complete $x_i x_j = x_j x_i$
- Elements on diagonal can be linearized $x_i^2 = x_i \text{ if } x_i \in \{0,1\}$
- Represents adjacency matrix of a problem

$$\begin{bmatrix} -46. & 0. & 0. & 48. & 48. & 48. & 0. & 48. & 48. & 48. & 48. \\ 0. & -44. & 0. & 48. & 0. & 48. & 48. & 0. & 48. & 48. & 48. \\ 0. & 0. & -44. & 0. & 48. & 0. & 48. & 48. & 48. & 48. & 48. \\ 48. & 48. & 0. & -92. & 48. & 96. & 48. & 48. & 96. & 96. & 96. \\ 48. & 0. & 48. & 48. & -92. & 48. & 48. & 96. & 96. & 96. & 96. \\ 48. & 48. & 0. & 96. & 48. & -92. & 48. & 48. & 96. & 96. & 96. \\ 0. & 48. & 48. & 48. & 48. & 48. & -91. & 48. & 96. & 96. & 96. \\ 48. & 0. & 48. & 48. & 96. & 48. & 48. & -92. & 96. & 96. & 96. \\ 48. & 48. & 48. & 96. & 96. & 96. & 96. & 96. & -139. & 144. & 144. \\ 48. & 48. & 48. & 96. & 96. & 96. & 96. & 96. & 144. & -138. & 144. \\ 48. & 48. & 48. & 96. & 96. & 96. & 96. & 96. & 144. & 144. & -139. \end{bmatrix}$$

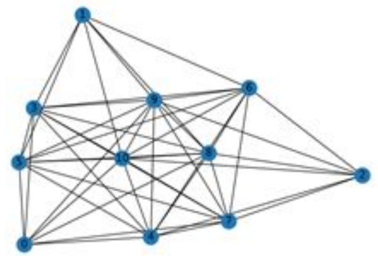
Offset c

- Irrelevant for optimization

In terms of IP it would be a

(possible non-convex if $\mathbf{Q} \not\succeq 0$)

Integer Nonlinear Program



QUBO - Ising Mapping

1-to-1 mapping between QUBO and Ising models using spin-binary bijection

$$\sigma_i = 2x_i - 1 \qquad \Longleftrightarrow \qquad x_i = (\sigma_i + 1)/2$$

$$\sigma_i \sigma_j = 4x_i x_j - 2x_i - x_j + 1 \qquad \qquad x_i x_j = (\sigma_i \sigma_j + \sigma_i + \sigma_j + 1)/4$$

$$\min_{\sigma \in \{-1,+1\}^n} \sum_{(ij) \in E(G)} J_{ij} \sigma_i \sigma_j + \sum_{i \in V(G)} h_i \sigma_i + c_I =$$

$$\min_{\mathbf{x} \in \{0,1\}^n} \sum_{(ij) \in E(G)} x_i Q_{ij} x_j + \sum_{i \in V(G)} Q_{ii} x_i + c_Q$$

$$Q_{ij} = 4J_{ij}, Q_{ii} = 2h_i - \sum_{j \in V(G)} (2J_{ij} + 2J_{ji}), c_I = c_Q + \sum_{i<j} J_{ij} - \sum_{i \in V(G)} h_i$$

$$\Updownarrow$$

$$\min_{\mathbf{x} \in \{0,1\}^n} \sum_{(ij) \in E(G)} x_i Q_{ij} x_j + \sum_{i \in V(G)} Q_{ii} x_i + c_Q =$$

$$\min_{\sigma \in \{-1,+1\}^n} \sum_{(ij) \in E(G)} J_{ij} \sigma_i \sigma_j + \sum_{i \in V(G)} h_i \sigma_i + c_I$$

$$J_{ij} = Q_{ij}/4, h_i = Q_{ii}/2 + \sum_{j \in V(G)} (Q_{ij}/4 + Q_{ji}/4), c_I = c_Q + \sum_{i<j} Q_{ij}/4 - \sum_{i \in V(G)} Q_{ii}/2$$

QUBO as Integer Programs

$$\min_{\mathbf{x}\in\{0,1\}^n} \sum_{(ij)\in E(G)} x_i Q_{ij} x_j + \sum_{i\in V(G)} Q_{ii} x_i + c$$
$$= \min_{\mathbf{x}\in\{0,1\}^n} \mathbf{x}^\top \mathbf{Q}\mathbf{x} + c$$

Although this is already solvable using INLP programming tools, we can reformulate it as a ILP by adding a variable $x_{ij} = x_i x_j$ whose nonlinearity can be posed a linear inequalities.

Experimental results show this is the most efficient ILP formulation of the Ising problem

$$\min_{\mathbf{x}\in\{0,1\}^n} \sum_{(ij)\in E(G)} Q_{ij} x_{ij} + \sum_{i\in V(G)} Q_{ii} x_i + c$$
$$\text{s.t. } x_{ij} \geq x_i + x_j - 1, x_{ij} \leq x_i, x_{ij} \leq x_j \quad \forall (ij) \in E(G)$$
$$x_i \in \{0,1\} \quad \forall i \in V(G), x_{ij} \in \{0,1\} \quad \forall (ij) \in E(G)$$

[1] Billionnet, A., Elloumi, S.: Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. Mathematical Programming 109(1) (2007) 55–68

QUBO – Binarization

How to transform general Integer Programs into Binary programs?

$$y \in \{0, \cdots, \bar{y}\} \subseteq \mathbb{Z}$$

In general with a width of the integer encoding d

$$y = \sum_{j=1}^{d} k_j x_j = \mathbf{k}^\top \mathbf{x}, k_j \in \mathbb{Z}_+, x_j \in \{0,1\}$$

- Unary encoding $k_j = 1, d = \bar{y}$
- Binary encoding $k_j = 2^{j-1}, d = \lfloor \log_2(\bar{y}) \rfloor$
- Bounded encoding

We can find an encoding with an upper bound for the coefficients $\mu \ll \bar{y}$

$$\text{if } \bar{y} < 2^{\lfloor \log(\mu) \rfloor} + 1$$

$$\mathbf{k} = \left[2^0, 2^1, \dots, 2^{\lfloor \log(\bar{y}) \rfloor - 1}, \bar{y} - \sum_{i=1}^{\lfloor \log(\bar{y}) \rfloor} 2^{i-1}\right]$$

$$\text{if } \bar{y} > 2^{\lfloor \log(\mu) \rfloor} + 1$$

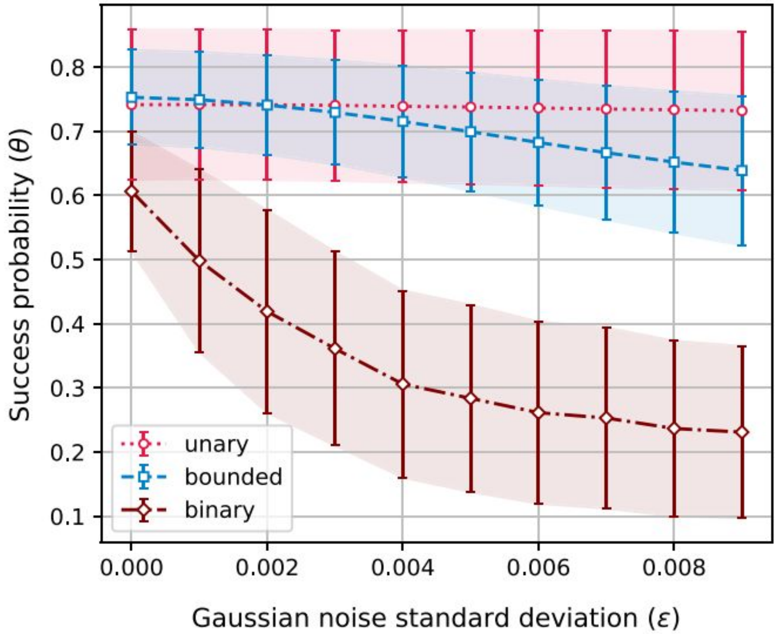
$$\rho = \lfloor \log \mu \rfloor + 1, v = \bar{y} - \sum_{i=1}^{\rho} 2^{i-1}, \text{ and } \eta = \left\lfloor \frac{v}{\mu} \right\rfloor$$

$$k_i = \begin{cases} 2^{i-1} & \text{for } i = 1, \dots, \rho \\ \mu & \text{for } i = \rho + 1, \dots, \rho + \eta \\ v - \eta\mu & \text{for } i = \rho + \eta + 1 \text{ if } v - \eta\mu \neq 0 \end{cases}$$

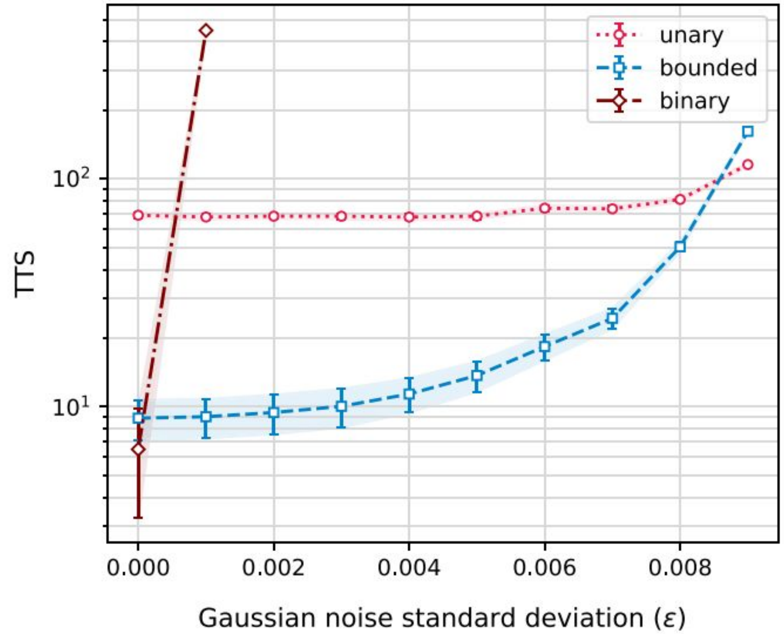
[1] Karimi, Sahar, and Pooya Ronagh. "Practical integer-to-binary mapping for quantum annealers." Quantum Information Processing 18.4 (2019): 94.

QUBO – Binarization

$\mu \ll \bar{y}$ upper bound can be computed from the problem matrix and the resolution of the quadratic and linear terms in the machine



(a) The scaling of success probability as a function of the standard deviation of noise. The success probability θ is reported averaged over the 128 UIQP instances of each noise parameter. The shaded stripe indicates the 50th percentile.



(b) The scaling of TTS as a function of the standard deviation of noise. Instances for which the optimal solution is never observed are ignored and the curves are discontinued if more than 15% of instances are never solved.

[1] Karimi, Sahar, and Pooya Ronagh. "Practical integer-to-binary mapping for quantum annealers." Quantum Information Processing 18.4 (2019): 94.

QU*B*O - Pseudo-Boolean functions and Quadratization

Any pseudo-Boolean function defined as

$$f : \{0,1\}^n \mapsto \mathbb{R}$$

Can be written uniquely as a sum of multilinear functions

$$f(x) = a_0 + \sum_i a_i x_i + \sum_{ij} a_{ij} x_i x_j + \sum_{ijk} a_{ijk} x_i x_j x_k + \ldots$$

We can transform any binary polynomial into a quadratic polynomial by introducing new variables as we saw before

Naive example: $x_{ij} = x_i x_j$

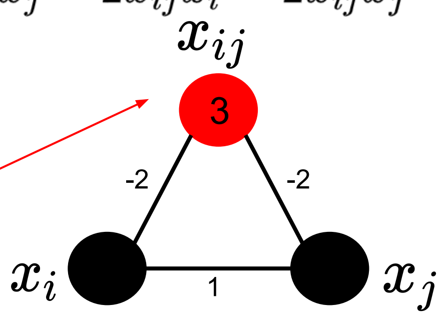
- Using linear inequalities

  $x_{ij} \geq x_i + x_j - 1, x_{ij} \leq x_i, x_{ij} \leq x_j$

- Using a polynomial expression

  $H(\mathbf{x}) = 3x_{ij} + x_i x_j - 2x_{ij} x_i - 2x_{ij} x_j$

  - Graph:



Usually called ancillary variables

x_i	x_j	x_{ij}	$x_i x_j$	$H(\mathbf{x})$
1	**1**	**1**	**1**	**0**
1	1	0	1	1
1	0	1	0	1
1	**0**	**0**	**0**	**0**
0	1	1	0	1
0	**1**	**0**	**0**	**0**
0	0	1	0	3
0	**0**	**0**	**0**	**0**

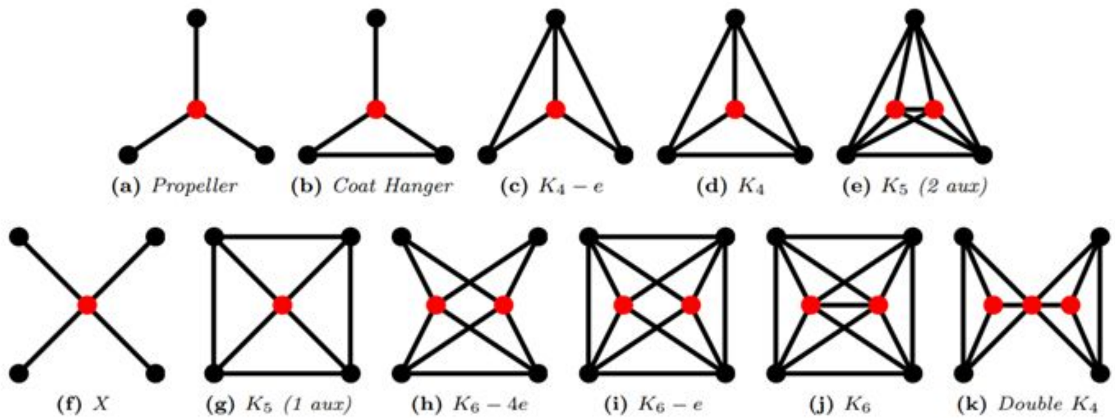
[1] Boros, Endre, and Peter L. Hammer. "Pseudo-boolean optimization." Discrete applied mathematics 123.1-3 (2002): 155-225.

QUBO - Quadratization and logical functions

We can follow the previous reduction to obtain a Quadratically constrained program from any arbitrary problem.

There are many clever way of “Quadratizing” problems [1]

- Cubic
- Quartic



Logical gates

AND: $x_i \wedge x_j = x_i x_j = x_{ij} \mapsto 3x_{ij} + x_i x_j - 2x_{ij}x_i - 2x_{ij}x_j$

OR: $x_i \vee x_j = x_i + x_j - x_i x_j = x_i + x_j - x_{ij} \mapsto x_i + x_j - 3x_{ij} - x_i x_j + 2x_{ij}x_i + 2x_{ij}x_j$

NOT: $\neg x = y \mapsto 2xy - x - y$

[1] Dattani, N. (2019-01-14), Quadratization in discrete optimization and quantum mechanics, arXiv:1901.04405

QUBO – Unconstraining

Lagrange Multipliers

For LP problems we had

$$\min_{\mathbf{x}\geq 0} \mathbf{c}^\top \mathbf{x} \quad \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{b} \quad \Longleftrightarrow \quad \min_{\mathbf{x}\geq 0} \mathbf{c}^\top \mathbf{x} + \lambda^\top(\mathbf{b} - \mathbf{A}\mathbf{x})$$

Where determining the multipliers can be written as

$$\max_\lambda \mathcal{L}(\lambda) = \max_\lambda \lambda^\top \mathbf{b} \quad s.t.\ \lambda^\top \mathbf{A} \leq \mathbf{c}^\top$$

and where the following holds:

$$\text{Strong duality: } \lambda^{*\top} \mathbf{b} = \mathbf{c}^\top \mathbf{x}^*$$

For IP problems, strong duality does not hold, but we can still use the multipliers to convert the problem into unconstrained

$$\min_{\mathbf{x}\in\{0,1\}^n} \mathbf{c}^\top \mathbf{x} \quad \text{s.t. } g(\mathbf{x}) \leq 0 \quad (\lambda), \quad h(\mathbf{x}) = 0 \quad (\rho)$$

$$\min_{\mathbf{x}\in\{0,1\}^n} \mathbf{c}^\top \mathbf{x} + \lambda^\top g(\mathbf{x}) + \rho^\top h(\mathbf{x}) \quad \text{where } \lambda_i^\top \geq 0$$

*QU*BO – Unconstraining

We can transform any arbitrary constraint in terms of binaries into multilinear, and then quadratize them.

We can then transform inequalities into equalities by introducing slack variables, which by scaling the constraints become integers

FInally we can binarize these slacks

$$
\begin{array}{l} \min_{\mathbf{x}\in\{0,1\}^n} \mathbf{c}^\top \mathbf{x} \\ \text{s.t. } g(\mathbf{x}) \le 0 \\ \quad h(\mathbf{x}) = 0 \end{array}
\Longrightarrow
\begin{array}{l} \min_{\mathbf{y}\in\{0,1\}^n} \mathbf{d}^\top \mathbf{y} \\ \text{s.t. } \mathbf{y}^\top G\mathbf{y} \le 0 \quad (\lambda) \\ \quad \mathbf{y}^\top H\mathbf{y} = 0 \quad (\rho) \end{array}
\Longrightarrow
\begin{array}{l} \min_{\mathbf{y}\in\{0,1\}^n, \mathbf{s}\in\mathbb{Z}_+} \mathbf{d}^\top \mathbf{y} \\ \text{s.t. } \mathbf{y}^\top G\mathbf{y} + \mathbf{s} = 0 \quad (\lambda) \\ \quad \mathbf{y}^\top H\mathbf{y} = 0 \quad (\rho) \end{array}
$$

Considering that we can only deal with discrete variable in QUBO, we cannot change the value of the multipliers.

- Set the multipliers to be constant penalty factor!
- Example: Integer linear inequalities become penalties by “squaring” them

$$
\begin{array}{l} \min_{\mathbf{x}\in\mathbb{Z}_+} \mathbf{c}^\top \mathbf{x} \\ \text{s.t. } \mathbf{A}\mathbf{x} \le \mathbf{b} \end{array}
\Longrightarrow
\min_{\mathbf{x}\in\mathbb{Z}_+, \mathbf{s}\in\mathbb{Z}_+} \mathbf{c}^\top \mathbf{x} + \rho^\top (\mathbf{A}\mathbf{x} + \mathbf{s} - \mathbf{b})^\top (\mathbf{A}\mathbf{x} + \mathbf{s} - \mathbf{b})
$$

QUBO - Unconstraining

Penalty factor

In general, we represent the constrained problem with a feasible region

$$F = \bigcap_j F_j = \bigcap_j \{\mathbf{x} \in \{0,1\}^n : g_j(\mathbf{x}) \leq 0\}$$

We are mapping it into a QUBO problem

$$\min_{\mathbf{y}\in\{0,1\}^n} \sum_{(ij)\in E(G)} y_i Q_{ij} y_j + \sum_{i\in V(G)} Q_{ii} y_i + c_Q$$

Where the $\mathbf{y}$ variables include both the original variables $\mathbf{x}$ and ancillas $\mathbf{a}$

The penalization factor for each constraint should read

$$\min_{\mathbf{a}} Pen_{F_j}(\mathbf{x}, \mathbf{a}) \begin{cases} = 0 \text{ if } \mathbf{x} \in F_j \\ \geq g \text{ if } \mathbf{x} \notin F_j \end{cases}$$

Where $g > 0$ is the gap between feasible and infeasible solutions.

Therefore we could in general find but in practice this becomes an MILP and its as tough to solve as the original QUBO

$$\begin{aligned} &\max_{g,Q,c_Q} g \\ &\text{s.t. } \mathbf{y}^\top Q \mathbf{y} + c_Q \geq 0 \quad \forall \mathbf{x} \in F, \forall \mathbf{a} \\ &\quad \mathbf{y}^\top Q \mathbf{y} + c_Q \geq g \quad \forall \mathbf{x} \notin F, \forall \mathbf{a} \\ &\quad \exists \mathbf{a} : \mathbf{y}^\top Q \mathbf{y} + c_Q = 0 \quad \forall \mathbf{x} \in F \\ &\quad \underline{Q} \leq Q \leq \overline{Q}, \underline{c_Q} \leq c_Q \leq \overline{c_Q} \end{aligned}$$

[1] Bian, Zhengbing, et al. "Discrete optimization using quantum annealing on sparse Ising models." Frontiers in Physics 2 (2014): 56.

QUBO - Examples

Binary Linear Programming

$$\min_{\mathbf{x}\in\{0,1\}^n} \mathbf{c}^\top \mathbf{x}$$

$$s.t.\ \mathbf{A}\mathbf{x} = \mathbf{b}$$

We can write the energy function in two terms $H = H_A + H_B$

- Constraint satisfaction $(H_A = 0)$

$$H_A = \rho \sum_{j=1}^{m} \left(\sum_{i=1}^{n} A_{ij} x_i - b_j\right)^2$$

- Objective function

$$H_B = \sum_{i=1}^{n} c_i x_i$$

How to determine the penalty? Guarantee that the minimal change in infeasibility is larger than the maximal case in objective

$$\rho \geq \frac{\Delta H_B^{\max}}{\Delta H_A^{\min}} \qquad \begin{aligned} \Delta H_B^{\max} &= \sum_{i=1}^{n} \max\{c_i, 0\} \\ \Delta H_A^{\min} &= \min_{\sigma_i \in \{0,1\}, j} \left(\max\left[1, \tfrac{1}{2} \sum_{i=1}^{n} (-1)^{\sigma_i} A_{ij}\right]\right) \end{aligned}$$

It can be tightened by using properties of the constraints!

[1] Lucas, Andrew. "Ising formulations of many NP problems." Frontiers in Physics 2 (2014): 5.

Putting it all together

Transforming IP into QUBO

Let’s go to the code

https://colab.research.google.com/github/bernalde/QuIP/blob/master/notebooks/Notebook%205%20-%20QUBO.ipynb

QUBO - Examples

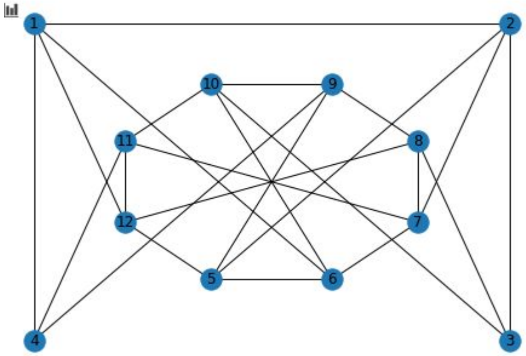
Coloring with colors $K = \{1, \cdots, k\}$

Groebner basis polynomial

$$\mathcal{S} = \{x_i^{|K|} = 1, \forall i \in V$$
$$x_u^{|K|} - x_v^{|K|} = 0, \forall (u,v) \in E\}$$

IP formulation (SAT)

$$\min_{\mathbf{x}} 1$$
$$s.t. \sum_{j \in K} x_{ij} = 1, \forall i \in V$$
$$x_{uj} + x_{vj} \leq 1, \forall j \in K, \forall (u,v) \in E$$
$$x_{ij} \in \{0,1\}, \forall j \in K, \forall i \in V$$

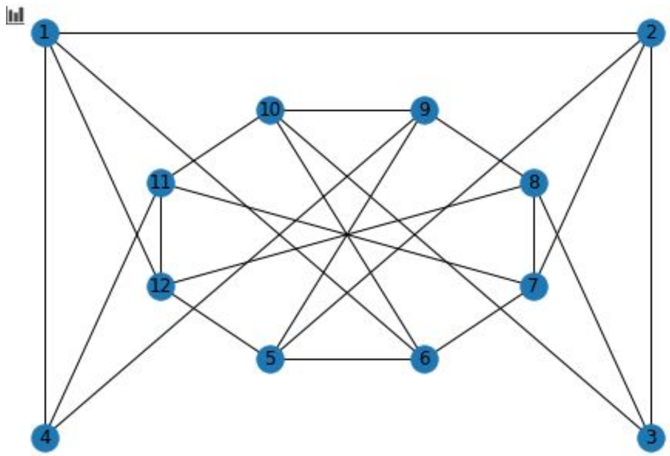
$\mathcal{G}(V,E) =$ 

QUBO Formulation

$$\min_{\mathbf{x}} \sum_{i \in V} \left(1 - \sum_{j \in K} x_{ij}\right)^2 + \sum_{(uv) \in E} \sum_{j \in K} x_{uj} x_{vj}$$
$$x_{ij} \in \{0,1\}, \forall j \in K, \forall i \in V$$

[1] Lucas, Andrew. "Ising formulations of many NP problems." Frontiers in Physics 2 (2014): 5.

QUBO for coloring initial graph



with 3 colors

https://colab.research.google.com/github/bernalde/QuIP/blob/master/notebooks/Notebook%205%20-%20QUBO.ipynb

QUBO - Quadratization

Algebraic geometry

Reduction to QUBOs without slack variables

Consider the quadratic polynomial

$$H_{ij} := Q_i P_j + S_{i,j} + Z_{i,j} - S_{i+1,j-1} - 2\,Z_{i,j+1},$$

with the binary variables $P_j, Q_i, S_{i,j}, S_{i+1,j-1}, Z_{i,j}, Z_{i,j+1}$.

- The goal is solve H_{ij} (obtain its zeros) as a QUBO (eg., using DWave)
- We can square H_{ij} and reduce using slack variables !
- Or, instead, we compute a Groebner basis $\mathcal{B}$ of the system

$$\mathcal{S} = \{H_{ij}\} \cup \{x^2 - x,\ x \in \{P_j, Q_i, S_{i,j}, S_{i+1,j-1}, Z_{i,j}, Z_{i,j+1}\}\},$$

  and look for a positive quadratic polynomial $H_{ij}{}^{+} = \sum_{t \in \mathcal{B}|\, deg(t) \leq 2} a_t t$. Note that global minima of $H_{ij}{}^{+}$ are the zeros of H_{ij}.

[1] Dridi, Raouf, and Hedayat Alghassi. "Prime factorization using quantum annealing and computational algebraic geometry." Scientific reports 7 (2017): 43048.

QUBO - Quadratization

Algebraic geometry

The Groebner basis $\mathcal{B}$ is

$$t_1 := Q_i P_j + S_{i,j} + Z_{i,j} - S_{i+1,j-1} - 2\,Z_{i,j+1}, \tag{8}$$

$$t_2 := \left(-Z_{i,j+1} + Z_{i,j}\right) S_{i+1,j-1} + \left(Z_{i,j+1} - 1\right) Z_{i,j}, \tag{9}$$

$$t_3 := \left(-Z_{i,j+1} + Z_{i,j}\right) S_{i,j} + Z_{i,j+1} - Z_{i,j+1} Z_{i,j}, \tag{10}$$

$$t_4 := \left(S_{i+1,j-1} + Z_{i,j+1} - 1\right) S_{i,j} - S_{i+1,j-1} Z_{i,j+1}, \tag{11}$$

$$t_5 := \left(-S_{i+1,j-1} - 2\,Z_{i,j+1} + Z_{i,j} + S_{i,j}\right) Q_i - S_{i,j} - Z_{i,j} + S_{i+1,j-1} + 2\,Z_{i,j+1}, \tag{12}$$

$$t_6 := \left(-S_{i+1,j-1} - 2\,Z_{i,j+1} + Z_{i,j} + S_{i,j}\right) P_j - S_{i,j} - Z_{i,j} + S_{i+1,j-1} + 2\,Z_{i,j+1}, \tag{13}$$

$$\text{in addition to 3 more cubic polynomials,} \tag{14}$$

We take $H_{ij}{}^{+} = \sum_{t \in \mathcal{B} |\, deg(t) \leq 2} a_t t$, and solve for the a_t. We can require that the coefficients a_t are subject to the dynamic range allowed by the quantum processor (eg., the absolute values of the coefficients of H_{ij}^{+}, with respect to the variables $P_j, Q_i, S_{i,j}, S_{i+1,j-1}, Z_{i,j}$, and $Z_{i,j+1}$, be within $[1-\epsilon,\ 1+\epsilon]$).

Carnegie Mellon University
Tepper School of Business

[1] Dridi, Raouf, and Hedayat Alghassi. "Prime factorization using quantum annealing and computational algebraic geometry." Scientific reports 7 (2017): 43048.

William Larimer Mellon, Founder

QUBO - Quadratization

scientific reports

View all Nature Research journals | Search | Login

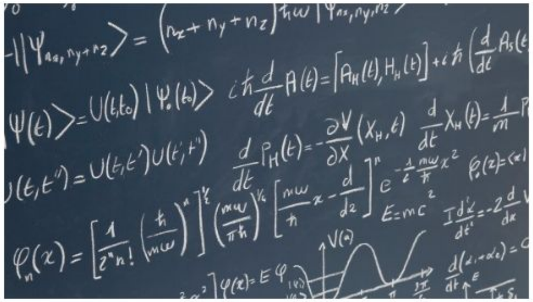
Explore our content ⌄ Journal information ⌄

nature › scientific reports › collection

COLLECTION | 11 OCTOBER 2019

Editor's choice: quantum computing

Quantum computers have developed enormously in the past decade, moving beyond theoretical calculations to real-world applications. Systems which use comparatively low numbers of qubits have been accessible to researchers for some time and, despite the inherent… show more

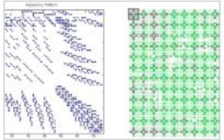


Quantum Annealing

ARTICLE
OPEN ACCESS
21 FEB 2017
Scientific Reports

Prime factorization using quantum annealing and computational algebraic geometry

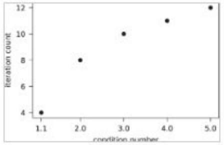
Raouf Dridi & Hedayat Alghassi



ARTICLE
OPEN ACCESS
16 JUL 2019
Scientific Reports

Quantum annealing for systems of polynomial equations

Chia Cheng Chang, Arjun Gambhir ··· Shigetoshi Sota



ARTICLE

Quantum Annealing for Prime Factorization

[1] Dridi, Raouf, and Hedayat Alghassi. "Prime factorization using quantum annealing and computational algebraic geometry." Scientific reports 7 (2017): 43048.

Integer Factorization by Raouf and Hedayat

Initial code

https://colab.research.google.com/github/bernalde/QuIP/blob/master/notebooks/Notebook%205%20-%20QUBO.ipynb