A QML application: the QUBO formulation of the Graph Colouring problem

L. Cappelli   M. Argenton   E. Calore   S.F. Schifano

INFN & University of Ferrara
laura.cappelli@fe.infn.it

2nd AI-INFN Advanced Hackathon
Pavia, 24-27 November, 2025

Founded by ICSC - Centro Nazionale di ricerca in HPC, Big Data e Quantum Computing
Spoke 10 - Quantum Computing

Combinatorial Optimisation Problems

- **Combinatorial Optimisation** (CO) is one of the most important areas in the field of optimisation, with a wide variety of real-world applications
  - Optimization problems involve making decisions in settings where many yes/no choices must be made
  - Each set of decisions yields a corresponding **objective function value** (e.g. a cost or profit value) that needs to be minimised or maximised
- Typical optimisation problems include:
  - Resource allocation problems
  - Distance minimisation
  - Profit maximisation
  - ...

The Quadratic Unconstrained Binary Optimisation Problem

The mathematical formulation known as **Quadratic Unconstrained Binary Optimization** (QUBO) can cover an extraordinary variety of important CO problems

QUBO formal definition

The QUBO model is expressed as:

$$\text{minimise/maximise} \quad C(x) = x^t Qx + c$$

where:

- x is a column vector of binary decision variables
- Q is a square matrix of constants
- c is a constant

Why QUBO?

- QUBO problems are **NP-hard***
  - Exact solvers will fail to stop in reasonable time; the best approach is to use brute-force
  - Heuristic methods find a high-quality solutions, not necessarily optimal, in a limited amount of time
- The equivalence between the QUBO problem and the Ising model has been shown
  - The Ising Hamiltonian can be used to formulate the QUBO problem as a quantum system
  - **Quantum computing** offers new heuristic methods:
    - Quantum circuit approaches, such as QAOA or VQE
    - Quantum annealing

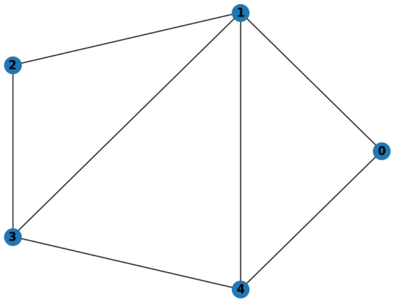
* Definitions in the field of computational complexity theory:
- NP problem: A decision problem whose proposed solutions can be verified in polynomial time.
- NP-complete problem: A problem that is in NP and to which every NP problem can be polynomially reduced.
- NP-hard problem: A problem to which every NP problem can be polynomially reduced, possibly without having solutions verifiable in polynomial time.

The Graph Colouring problem

Pills on graphs

A **graph** $G = (V, E)$ is a pair of sets:

- $V := v_1, v_2, v_3, ..., v_n$ is the set of **nodes**;
- $E := (v_i, v_j)|i, j \in 1, 2, ..., n \subseteq V \times V$ is the set of **edges**.



In the example:
$V = \{0, 1, 2, 3, 4\}$
$E = \{(0, 1), (1, 2), (1, 3), (1, 4), (2, 3), (3, 4), (4, 0)\}$

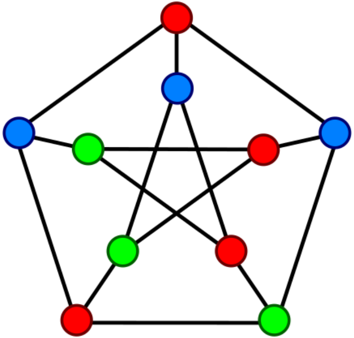
The graph can be described with its **adjacency matrix**:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Graph Colouring

For a graph $G = (V, E)$ and K colours, find a vertex-colour assignment such that two adjacent/connected vertices have different colours.

- several applications:
  - job scheduling,
  - registers allocation,
  - flight-gate assignment,
  - mobile radio frequency assignment,
  - ...



Formal Definition

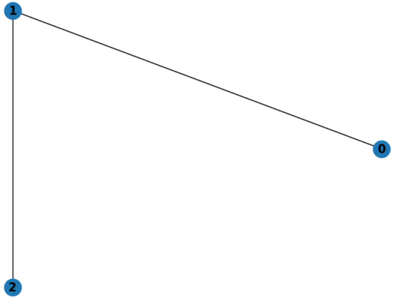
- For a graph $G = (V, E)$ and K colours, the binary variable x_{ip}:
  - is equal to 1 if vertex i has colour p
  - 0 otherwise
- The binary variables are: $N = n \times K$ (with n number of nodes)
- Solving the problem means finding an assignment in $\{0, 1\}$ for each x_{ij} variable that satisfy the following constraints
  1. Each node i is associated to only one colour
  2. Adjacent nodes can't have the same colour p

Graph colouring constraints

1. $\sum_{p=1}^{k} x_{ip} = 1 \qquad \forall i \in \{1 \dots n\}$
2. $x_{ip} + x_{jp} \leq 1 \qquad \forall p \in \{1 \dots k\}, (i, j) \in E$

Example

Let's consider the following simple graph:



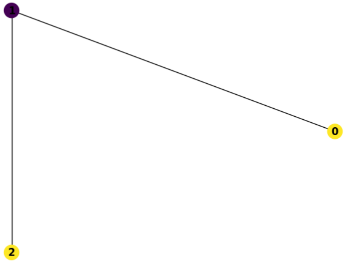
- $V = \{0, 1, 2\}$
- $E = \{(0, 1), (1, 2)\}$
- adjacency matrix:
$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

We want to solve the graph colouring problem with $K = 2$ and $p = \{yellow, purple\}$, meaning that we have to find:

$$\mathbf{x}^T = (x_{0,yellow}, x_{0,purple}, x_{1,yellow}, x_{1,purple}, x_{2,yellow}, x_{2,purple})$$

Example

A possible solution could be:



Problem definition:

- $V = \{0, 1, 2\}, n = 3$
- $E = \{(0, 1), (1, 2)\}$
- $K = 2$
- $p = \{yellow, purple\}$
- $N = n * K = 3 * 2 = 6$

Binary solution:

$$\mathbf{x}^T = (x_{0,yellow}, x_{0,purple}, x_{1,yellow}, x_{1,purple}, x_{2,yellow}, x_{2,purple})$$

$$\mathbf{x}^T = (1, 0, 0, 1, 1, 0)$$

From constraints to quadratic penalties

The graph colouring constraints can be translated into quadratic penalties and used to define the objective function $C(\mathbf{x})$

1. $\sum_{p=1}^{k} x_{ip} = 1 \quad \longrightarrow \quad \mu \left(\sum_{p=1}^{k} x_{ip} - 1 \right)^2 \quad \forall\, i \in \{1 \dots n\}$
2. $x_{ip} + x_{jp} \leq 1 \quad \longrightarrow \quad \mu \left(x_{ip} x_{jp} \right) \quad \forall p \in \{1 \dots k\}, (i,j) \in E$

Cost function

$$C(\mathbf{x}) = \mu \left(\sum_{p=1}^{k} x_{ip} - 1 \right)^2 + \mu(x_{ip} x_{jp}), \qquad \forall i \in \{1 \dots n\}, p \in \{1 \dots k\}, (i,j) \in E$$

where μ is an arbitrary integer positive constant.

The graph colouring cost function

Solving the graph colouring problem means finding the minimum of the binary cost function. In other words, it is a **minimization problem**.

$$\min_{\mathbf{x}\in\{0,1\}^N} C(\mathbf{x}) \qquad\qquad N = n \cdot k$$

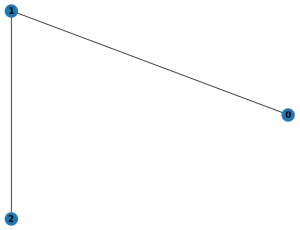
where

$$C(\mathbf{x}) = \mu \left(\sum_{p=1}^{k} x_{ip} - 1 \right)^2 + \mu (x_{ip} x_{jp}), \qquad \forall i \in \{1 \ldots n\}, p \in \{1 \ldots k\}, (i,j) \in E.$$

By solving the cost function separating the quadratic, linear, and constant terms, we obtain the QUBO formulation of the graph colouring problem.

$$\min_{\mathbf{x}\in\{0,1\}^N} C(\mathbf{x}) = x^t Q x + c$$

Example: defining the penalties



Considering the previous example where

$$V = \{0, 1, 2\}, E = \{(0, 1), (1, 2)\}, K = 2$$

and considering the penalties definition:

1. $\sum_{p=1}^{k} x_{ip} = 1 \longrightarrow P\left(\sum_{p=1}^{k} x_{ip} - 1\right)^2, \forall\, i \in \{1 \dots n\}$
2. $x_{ip} + x_{jp} \leq 1 \longrightarrow P\left(x_{ip}x_{jp}\right), \forall p \in \{1 \dots k\}, (i, j) \in E$

The penalties in this example are:

$$x_{01} + x_{02} = 1 \rightarrow \mu(x_{01} + x_{02} - 1)^2$$
$$x_{11} + x_{12} = 1 \rightarrow \mu(x_{11} + x_{12} - 1)^2$$
$$x_{21} + x_{22} = 1 \rightarrow \mu(x_{21} + x_{22} - 1)^2$$

$$x_{01} + x_{11} \leq 1 \rightarrow \mu(x_{01}x_{11})$$
$$x_{02} + x_{12} \leq 1 \rightarrow \mu(x_{02}x_{12})$$

$$x_{11} + x_{21} \leq 1 \rightarrow \mu(x_{11}x_{21})$$
$$x_{12} + x_{22} \leq 1 \rightarrow \mu(x_{12}x_{22})$$

Example: defining the cost function

Then, we define the objective function

$$C(x) = \mu((x_{01}+x_{02}-1)^2+(x_{11}+x_{12}-1)^2+(x_{21}+x_{22}-1)^2+x_{01}x_{11}+x_{02}x_{12}+x_{11}x_{21}+x_{12}x_{22})$$

Let us map x_{ik} into $y_{(i*K)+k}$:

$$[x_{01}, x_{02}, x_{11}, x_{12}, x_{21}, x_{22}] \longleftrightarrow [y_1, y_2, y_3, y_4, y_5, y_6]$$

and, taking into account that $y_i^2 = y_i$, we rewrite the objective function as

$$\begin{aligned} C(y) &= \mu((y_1 + y_2 - 1)^2 + (y_3 + y_4 - 1)^2 + (y_5 + y_6 - 1)^2 + y_1 y_3 + y_2 y_4 + y_3 y_5 + y_4 y_6) \\ &= \mu(+2y_1 y_2 + 2y_3 y_4 + 2y_5 y_6 + y_1 y_3 + y_2 y_4 + y_3 y_5 + y_4 y_6)+ \\ &\quad + \mu(-y_1 - y_2 - y_3 - y_4 - y_5 - y_6) + 3\mu \end{aligned}$$

Example: defining the QUBO formulation

The cost function

$$C(y) = \mu(+2y_1y_2 + 2y_3y_4 + 2y_5y_6 + y_1y_3 + y_2y_4 + y_3y_5 + y_4y_6) + \\ + \mu(-y_1 - y_2 - y_3 - y_4 - y_5 - y_6) + 3\mu$$

can be then written as

$$C(\mathbf{y}) = \mathbf{y}^{\mathbf{T}}\mathbf{Q}\mathbf{y} + \mathbf{g}^{\mathbf{T}}\mathbf{y} + \mathbf{c}$$

where

$$Q = \begin{bmatrix} 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{g} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix}, c = 3\mu$$

How can **quantum computing** help us to solve the graph colouring problem starting from its QUBO formulation?

The Ising Model

- The Ising model is well-known in the fields of theoretical and computational physics. Originally, it was applied to describe ferromagnetic materials and their phase transitions
- The **Ising Hamiltonian** describes a system consisting of N particles with spin $1/2$, where the nearest neighbours $\langle ij \rangle$ interact through a quadratic term. A linear term models the effect of an external field

Ising Model

$$\mathcal{H} = -\sum_{\langle ij \rangle} J_{ij}\sigma_i\sigma_j - \mu \sum_j h_j\sigma_j \tag{1}$$

J_{ij} and h_j are the coupling constants related to the interactions, and σ_k is the Pauli (usually Z) operator measuring the k-th spin with possible outcomes $\{-1, +1\}$

From QUBO to Ising

The cost function and the Ising Hamiltonian are similar in form. To derive the latter from the former, we first perform a change of variables.

QUBO formulation

$$\min_{\mathbf{y}\in\{0,1\}^N} C(\mathbf{y})$$

Equivalent Ising problem

$$\min_{\mathbf{z}\in\{-1,+1\}^N} E(\mathbf{z})$$

From QUBO to Ising (2)

Given the QUBO definition

$$C(\mathbf{y}) = \mathbf{y}^T Q\mathbf{y} + \mathbf{g}^T\mathbf{y} + c$$

using the mapping $y_i = \frac{1-z_i}{2}$, $z_i \in \{-1, 1\}$, we define the function $E(\mathbf{z})$

$$E(\mathbf{z}) = \mathbf{z}^T\mathbf{Q_z}\mathbf{z} + \mathbf{g_z}^T\mathbf{z} + c_z$$

where

$$\begin{aligned}
\mathbf{Q_z} &= 0.25 \cdot \mathbf{Q} \\
\mathbf{g_z}^T &= -0.25 \cdot \mathbf{1}^T\left(\mathbf{Q}^T + \mathbf{Q}\right) - 0.5\mathbf{g}^T \\
c_z &= 0.25 \cdot \mathbf{1}^T\mathbf{Q}\mathbf{1} + 0.5\mathbf{g}^T\mathbf{1}
\end{aligned}$$

From QUBO to Ising (3)

The energy function $E(\mathbf{z})$ can be translated into a Hamiltonian $\mathcal{H}$ for an N-qbit system by replacing the quadratic and the linear terms in $E(\mathbf{z})$ as follows:

- quadratic terms $z_i z_j \to P_{ij} = \mathbf{1}_1 \otimes \ldots \otimes \underbrace{\sigma_i^Z}_{i^{th}\text{ factor}} \otimes \cdots \otimes \underbrace{\sigma_j^Z}_{j^{th}\text{ factor}} \otimes \cdots \otimes \mathbf{1}_N$
- linear terms $z_i \to P_i = \mathbf{1}_1 \otimes \cdots \otimes \underbrace{\sigma_i^Z}_{i^{th}\text{ factor}} \otimes \cdots \otimes \mathbf{1}_N$

where σ_k^Z is a **Pauli Z-matrix** and $\mathbf{1}$ is the identity matrix.

Ising Hamiltonian

$$\mathcal{H} = \sum_i^N \sum_j^N Q_z^{[i,j]} P_{ij} + \sum_i^N g_z^{[i]} P_i + c_z$$

The Hamiltonian $\mathcal{H}$ we derived is an **operator** that allows us to extract the energy of the system in a given state ψ. With the Dirac notation:

$$\text{energy} = \langle\psi|\,\mathcal{H}\,|\psi\rangle$$

Solving the graph colouring problem means finding the **ground state** of the system, or finding the state ψ that minimizes the system's energy.

$$\min_{\psi} \langle\psi|\,\mathcal{H}\,|\psi\rangle$$

Searching the ground state of an Ising Hamiltonian

- Quantum computation offers several methods for searching the ground state of an Ising Hamiltonian
- In this presentation (and in the exercise of this afternoon) we will explore the **Variational Quantum Eigensolver** (VQE)
  - An iterative hybrid classical-quantum algorithm used to find the ground state of a system
  - It uses the gate-based paradigm and it is based on a variational quantum circuit
- Extra exercise for this afternoon: solve the problem using a quantum annealer

Building blocks of VQE

The building blocks of the VQE are:

- A variational quantum circuit, or **ansatz**, $U(\theta_i)$
- A set of parameters θ_i that dictate the behaviour of the quantum gates
- A classical optimizer (e.g. SPSA) that tries to find a new set of rotation angles θ_{i+1} that improves the result

The algorithm is hybrid, which means we need the following:

- A **QPU** to evolve the input state $|\psi_i\rangle$ to an output state $|\psi_{i+1}\rangle$ using the ansatz and the parameters and measures the system;
- A **CPU** to run the optimizer.

Steps of VQE

- At the beginnin $|\psi_0\rangle = |0\rangle$ and θ_0 is initialized randomly
- While the energy estimate is not satisfactory, the iterative scheme with the ansatz $U(\theta_i)$ is:
  1. The state $|\psi_i\rangle$ is fed to the **QPU**, which evolves it to the output state $|\psi_{i+1}\rangle = U(\theta_i)\,|\psi_i\rangle$
  2. The **QPU** measurers the energy of the state $|\psi_{i+1}\rangle$
  3. On the **CPU** the classical optimizer selects a new set of rotation angles θ_{i+1} for the next iteration with the goal to minimize the system's energy. The next iteration starts from the state $|\psi_{i+1}\rangle$

$$\psi(\theta_1) = U(\theta_0)|0\rangle \qquad \psi(\theta_2) = U(\theta_1)|\psi(\theta_1)\rangle \quad \dots \quad \psi(\theta_{i+1}) = U(\theta_i)\psi(\theta_i)$$

- By the end of the loop, ψ_n should correspond to the state of minimum energy. Note: VQE is an **heuristic method**, not always the solution is the global minimum

Conclusion

Conclusion

- Combinatorial Optimisation (CO) problems are of interest to industries
- **Quantum machine learning** (QML) offers alternative and novel heuristic approaches to tackle CO problems
  - Using gate-based quantum computers, e.g. VQE
  - Using quantum annealers
- This is a **hot topic**: there is a lot of active research on QML algorithms for CO problems since the field is quite new. The main research topics are:
  - How to find appropriate ansatz
  - How to initialize the ansatz parameters
  - Find better heuristic algorithms