Write a quantum program with myQLM

17-03-2021

What will we see?

Some features of myQLM

The workflow of a quantum program with myQLM

How to join the myQLM community

Focusing on quantum algorithms

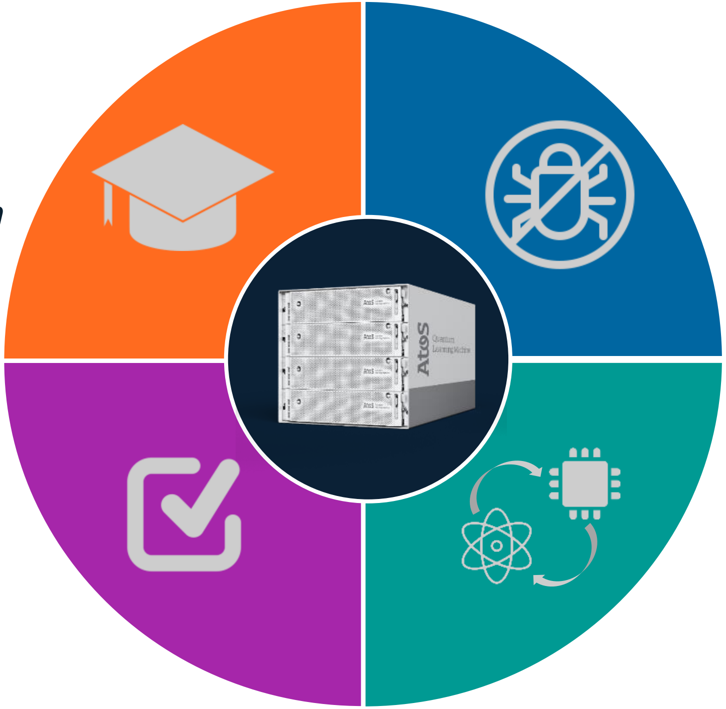
myQLM is great tool to…

LEARN

Get acquainted with quantum computing

TEST

Conceive new programs …
… and debug them



OPTIMIZE

Select the best quantum technology to solve your problem

RUN HYBRID CODE

Off-load the quantum-accelerable parts to the simulated QPU

myQLM packages

PROGRAMMING

AQASM
Assembly language to build quantum circuits

pyAQASM
Python extension to AQASM

CIRC
Binary format of quantum circuits

QLIB
AQASM & pyAQASM libraries
QRAM, oracle emulator, arithmetic libraries and more!

rigetti

Qiskit

INTEROP
Connectors' source codes: build your own!

Open Source

Project Cirq

SIMULATION

Open Source

SIMULATORS

- pyLinalg
- Simulated Annealing

Source code of this simulator: build your own!

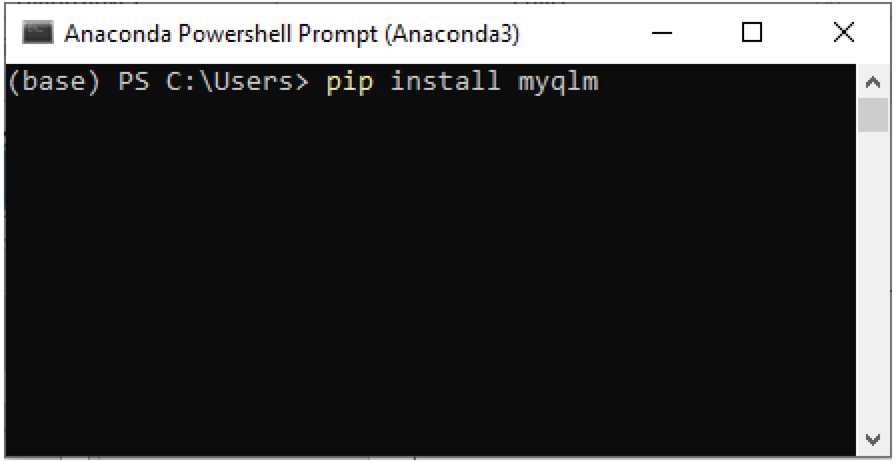
Atos

Installation

- Requirements
  - Python distribution
    - Windows: python 3.6-3.8
    - Linux: python 3.6 or 3.8
    - MacOS: python 3.6-3.8

- Instructions for installing extra packages at

  https://myqlm.github.io



Quantum Approximation Optimization Algorithm (QAOA)

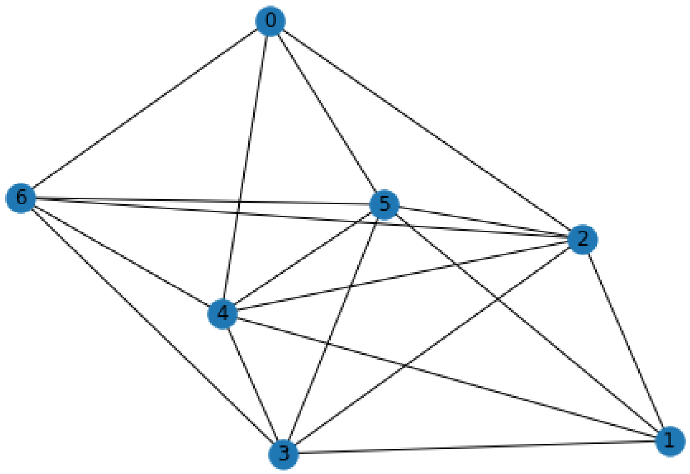
1

Atos

Max-cut problem

Let's keep it practical

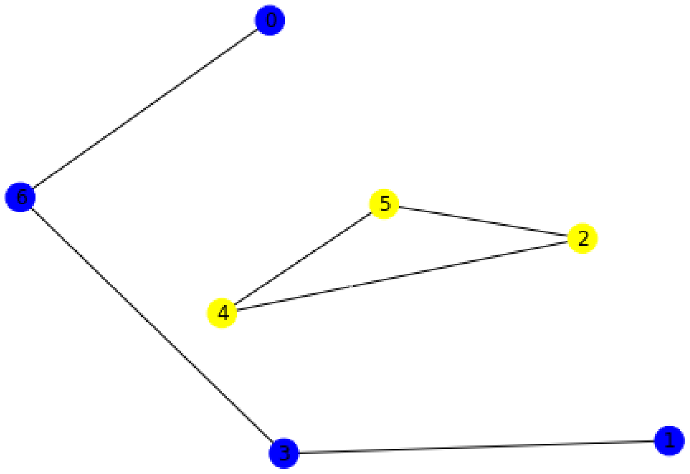
- **Max-Cut:** Divide a graph into two subsets such that the number of edges removed is maximized.



Max-cut problem

Let's keep it practical

- **Max-Cut:** Divide a graph into two subsets such that the number of edges removed is maximized.



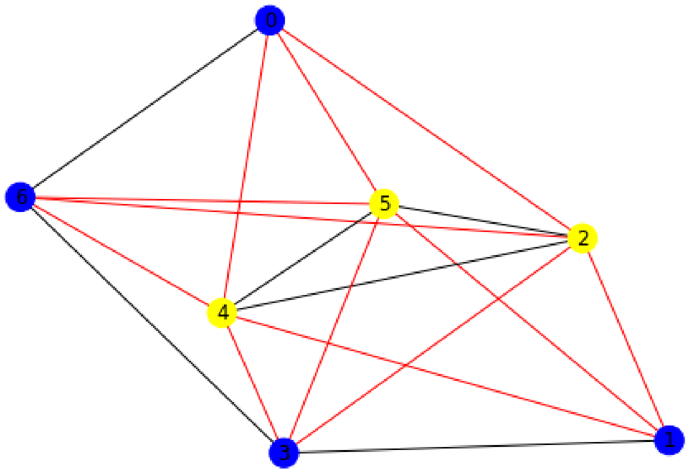
Max-cut problem

Let’s keep it practical

- **Max-Cut:** Divide a graph into two subsets such that the number of edges removed is maximized.

- NP-Hard problem that consists in maximizing the objective function:

$$O(\boldsymbol{z}) = \sum_{(i,j)\in E} \frac{1 - z_i z_j}{2}, \qquad \boldsymbol{z} \in \{-1, 1\}^N$$



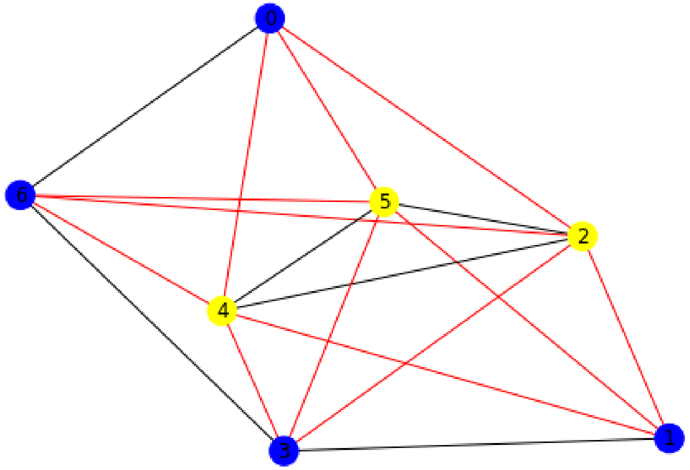
Max-cut problem

Let’s keep it practical

- **Max-Cut:** Divide a graph into two subsets such that the number of edges removed is maximized.

- NP-Hard problem that consists in <u>minimizing</u> the cost function:

$$C(\boldsymbol{z}) = \sum_{(i,j) \in E} z_i z_j \,, \qquad \boldsymbol{z} \in \{-1,1\}^N$$



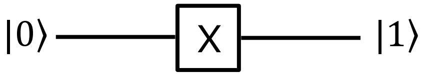
How to design cost functions with quantum objects?

$$C(\boldsymbol{z}) = \sum_{(i,j)\,\in\,E} z_i z_j \,, \qquad \boldsymbol{z} \in \{1,1\}^N$$

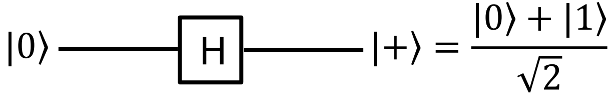
- **z** is a bitstring, $C(\boldsymbol{z})$ is a sum of numbers
- Quantum objects:
  - States (vectors): $|\boldsymbol{z}\rangle = |10010\rangle$, $|\boldsymbol{z}_s\rangle = \frac{1}{\sqrt{2}}(\,|10000\rangle + |10001\rangle\,)$
  - Gates (operators that transform states): H, CNOT
  - Observables (measurable operators or matrices)

Quantum gates and circuits

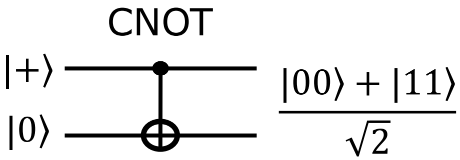
- Quantum gate: logical operation that transforms the input state of some qubits



$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

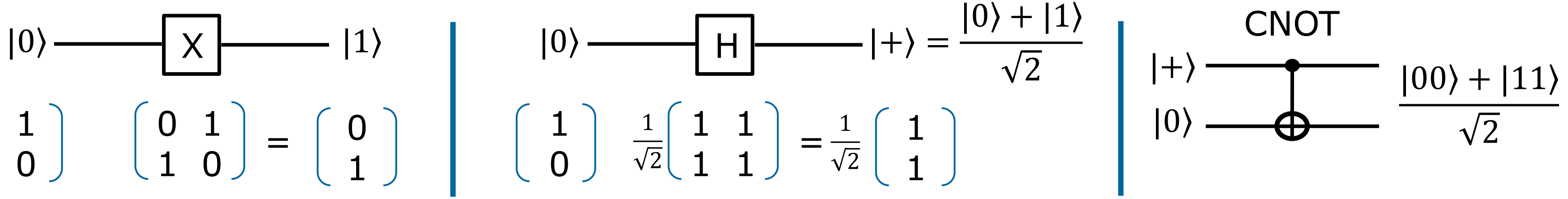


$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$



Quantum gates and circuits

- Quantum gate: logical operation that transforms the input state of some qubits



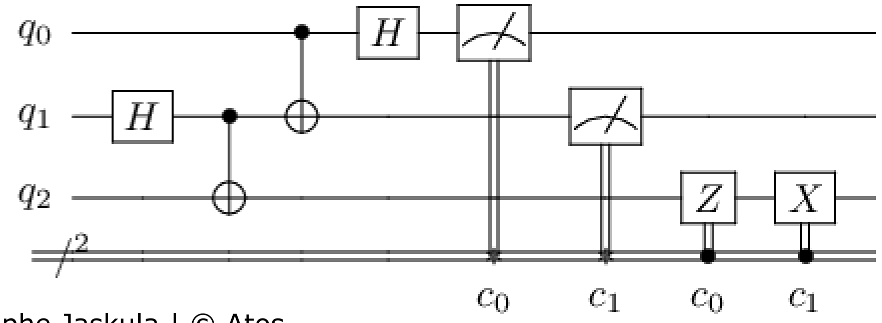
$$|0\rangle \xrightarrow{X} |1\rangle \qquad \begin{pmatrix}1\\0\end{pmatrix} \quad \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix} = \begin{pmatrix}0\\1\end{pmatrix}$$

$$|0\rangle \xrightarrow{H} |+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \qquad \begin{pmatrix}1\\0\end{pmatrix} \quad \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\1 & 1\end{pmatrix} = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$$

$$\text{CNOT}: |+\rangle, |0\rangle \rightarrow \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

- Quantum circuit is a sequence of gates

Quantum teleportation protocol



Quantum Approximation Optimization Algorithm
Cost function

An **observable** is an operator representing a physical quantity that can be measured
- z component of a qubit (noted Z, S_z, σ_z)

With a single qubit, $\langle 0|Z|0\rangle = +1$, $\langle 1|Z|1\rangle = -1$

With two qubits, $\langle 00|Z_0Z_1|00\rangle = +1 * +1$, $\langle 10|Z_0Z_1|10\rangle = -1 * +1$, $\langle \boldsymbol{z}|Z_0Z_1|\boldsymbol{z}\rangle = z_0z_1$

$C(\boldsymbol{z}) = \sum\langle \boldsymbol{z}|Z_iZ_j|\boldsymbol{z}\rangle = \langle \boldsymbol{z}|\sum Z_iZ_j|\boldsymbol{z}\rangle = \langle \boldsymbol{z}|C|\boldsymbol{z}\rangle$ expectation value of a well-chosen observable C is our cost function

$|\boldsymbol{z}\rangle$ will be constructed with a parametrized quantum circuit

Quantum Approximation Optimization Algorithm

Ansatz construction

EXPLORATION

$$U(B,\beta) = e^{-i\beta B} = \prod_{j=1}^{n} e^{-i\beta X_j}$$

EXPLOITATION

$$U(C,\gamma) = e^{-i\gamma C} = \prod_{(k,l)} e^{-i\gamma Z_k Z_l}$$

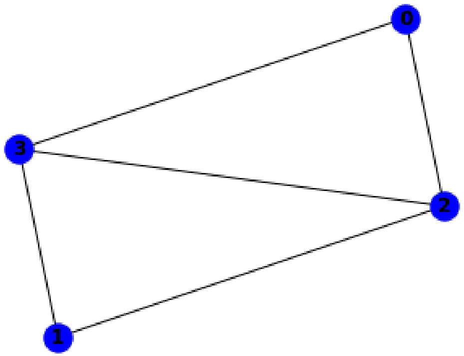
Quantum Approximation Optimization Algorithm

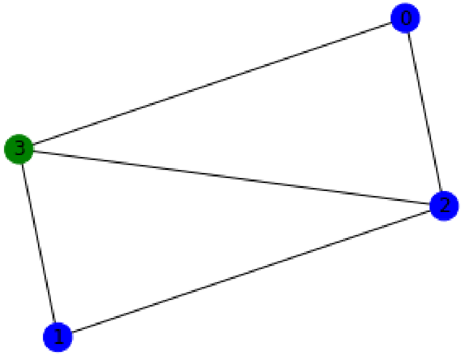
Step 1: exploration

Use of mixers to explore potential solutions within neighboring states

$|0000\rangle$

$e^{-i\beta X_3}|0000\rangle = g(\beta)|0000\rangle + f(\beta)|0001\rangle$





EXPLORATION

$$U(B,\beta) = e^{-i\beta B} = \prod_{j=1}^{n} e^{-i\beta X_j}$$

Continuous driving:

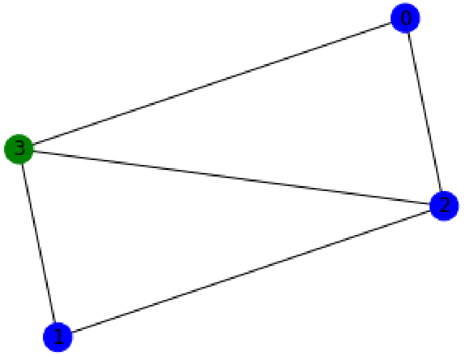
Superposition of possible states

Quantum Approximation Optimization Algorithm

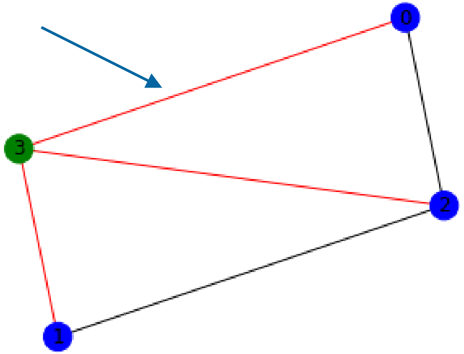
Step 2: exploitation

Promoting states that satisfy the constraints of the problem

$$|\psi\rangle = \alpha_1|0000\rangle + \alpha_2|0001\rangle$$



$$Z_0 Z_3|\psi\rangle = \alpha_1|0000\rangle - \alpha_2|0001\rangle$$



EXPLOITATION

$$U(C,\gamma) = e^{-i\gamma C} = \prod_{(k,l)} e^{-i\gamma Z_k Z_l}$$

Atos

Quantum Approximation Optimization Algorithm

Ansatz construction

EXPLORATION

$$U(B,\beta) = e^{-i\beta B} = \prod_{j=1}^{n} e^{-i\beta X_j}$$

EXPLOITATION

$$U(C,\gamma) = e^{-i\gamma C} = \prod_{(k,l)} e^{-i\gamma Z_k Z_l}$$

Alternating between exploitation and exploration with different strengths (γ and β):

$$\underbrace{|\boldsymbol{z}_{\boldsymbol{\beta},\boldsymbol{\gamma}}\rangle}_{\text{Ansatz}} = \underbrace{U(B,\beta_p)U(C\ \gamma_p)\ldots U(B,\beta_1)U(C,\gamma_1)}_{\text{Circuit (Alternating gates)}}\underbrace{|\boldsymbol{z}_{\boldsymbol{0}}\rangle}_{\text{Initial state}}$$

Quantum Approximation Optimization Algorithm

Optimization

EXPLORATION

$$U(B,\beta) = e^{-i\beta B} = \prod_{j=1}^{n} e^{-i\beta X_j}$$

EXPLOITATION

$$U(C,\gamma) = e^{-i\gamma C} = \prod_{(k,l)} e^{-i\gamma Z_k Z_l}$$

Alternating between exploitation and exploration with different strengths (γ and β):

$$\underbrace{|\boldsymbol{z}_{\boldsymbol{\beta},\boldsymbol{\gamma}}\rangle}_{\text{Ansatz}} = \underbrace{U(B,\beta_p)U(C\ \gamma_p)\ldots U(B,\beta_1)U(C,\gamma_1)}_{\text{Circuit (Alternating gates)}}\underbrace{|\boldsymbol{z}_{\boldsymbol{0}}\rangle}_{\text{Initial state}}$$

Solution: $\boldsymbol{min}_{\boldsymbol{\beta},\boldsymbol{\gamma}}\ \langle \boldsymbol{z}_{\boldsymbol{\beta},\boldsymbol{\gamma}}|\boldsymbol{C}|\boldsymbol{z}_{\boldsymbol{\beta},\boldsymbol{\gamma}}\rangle$ with a classical optimizer

Atos

Quantum Approximation Optimization Algorithm

Summary

1. Prepare a circuit that outputs a parametrized state $|z_{\beta,\gamma}\rangle$

Coding

Quantum Approximation Optimization Algorithm

Summary

1. Prepare a circuit that outputs a parametrized state $|z_{\beta,\gamma}\rangle$

Coding

2. Choose some parameters and compile this circuit $|\boldsymbol{z_{1,0}}\rangle$

3. Evaluate the cost function from this state $\langle \boldsymbol{z_{1,0}} | C | \boldsymbol{z_{1,0}} \rangle$

Execution

2′-3′. Modify the parameters and optimize the expectation value of the cost operator $\langle \boldsymbol{z_{1,0.1}} | C | \boldsymbol{z_{1,0.1}} \rangle$

Quantum Approximation Optimization Algorithm
Summary

1. Prepare a circuit that outputs a parametrized state $|z_{\beta,\gamma}\rangle$

Coding

2. Choose some parameters and compile this circuit $|z_{1,0}\rangle$

3. Evaluate the cost function from this state $\langle z_{1,0}|C|z_{1,0}\rangle$

Execution

2′-3′. Modify the parameters and optimize the expectation value of the cost operator $\langle z_{1,0.1}|C|z_{1,0.1}\rangle$

4. Compile a new circuit with the optimal parameters and retrieve a solution state $|\boldsymbol{z_{opt}}\rangle \rightarrow \boldsymbol{z_{opt}}$

Analysis

Let’s code it with myQLM

Workflow

	Important Classes	Tasks
Implementation	• Program() • Circuit()	• Allocate qubit register • Implement quantum algorithms
Execution	• Job() • Plugins and PyLinalg()	• Pre- and post-treatment • Simulation
Analysis	• Results()	• Sample analysis • Visualization

Demo

Let’s dive into QAOA

News and Community

Slack forums

Github external contributions

https://atos.net/myqlm

Thank you

For more information please contact:

myqlm@atos.net