PASQAL

Towards efficient hybrid algorithms using neutral atoms

Hybridation Quantum Processor Unit QPU / CPU – GPU
January 11 2023

PASQAL
www.pasqal.com
office@pasqal.com
2, av. Augustin Fresnel
91120 Palaiseau
France

Copyright Qu & Co BV

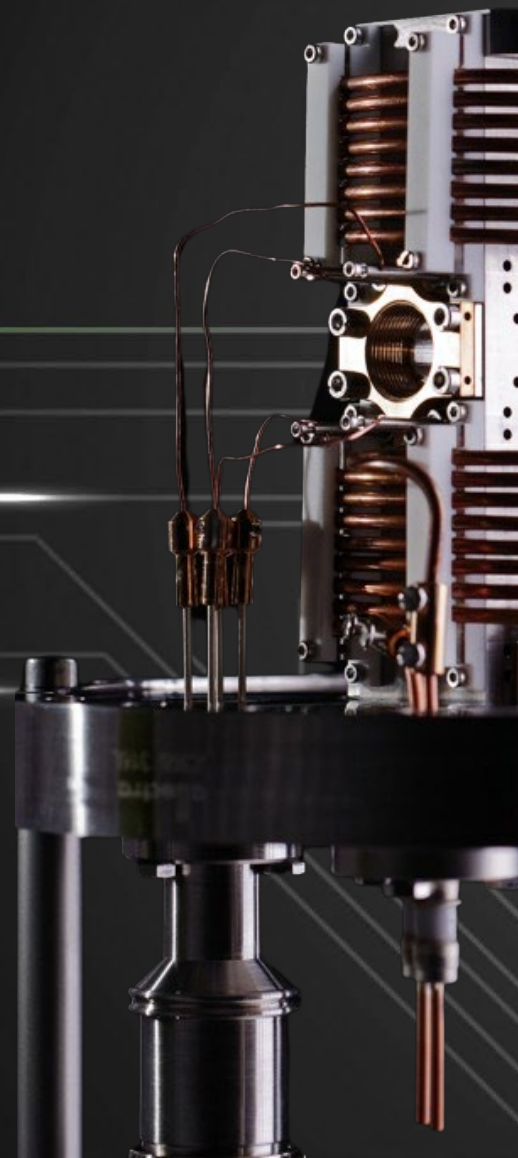
Agenda

1. Why do we need hybrid architectures
2. An example of hybrid algorithm
3. How would it work in practice

PASQAL

Copyright PASQAL SAS

About PASQAL



PASQAL in a few words

- **Our Legacy**
  - A word leading hub of Quantum science & Technology in Paris-Saclay.
  - Spin-out from Antoine BROWAEYS & Thierry LAHAYE's Lab
  - Cofounded by Alain Aspect

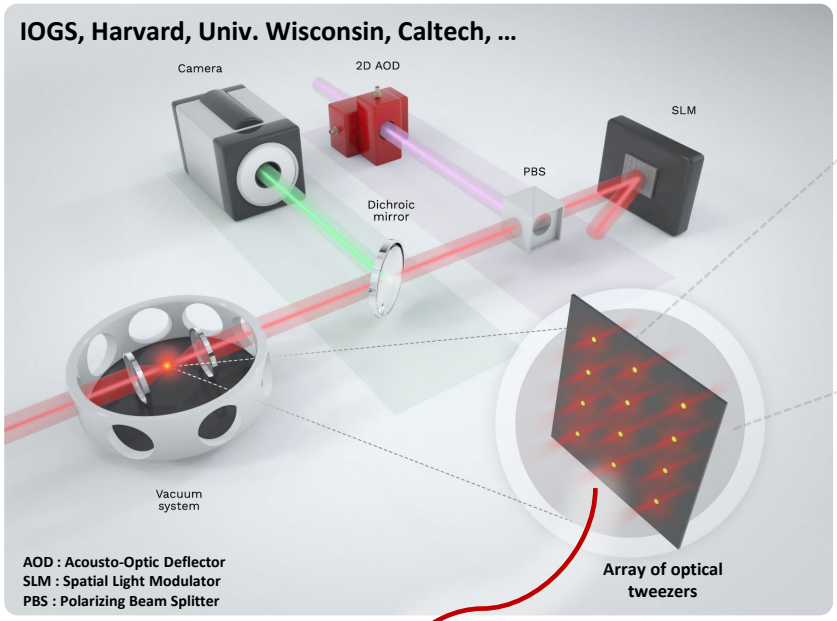
- **Our Solutions**
  - Neutral atoms (^{87}Rb) trapped with optical tweezers, driven by lasers and interacting using Rydberg physics.
  - Multi-purpose, flexible, 100 – 1000 qubit Quantum Processing Units (QPUs) for analog and digital QC.

- **Our Strengths**
  - Neutral atoms QPUs with many qubits of tunable geometries with unrivaled performance.
  - A full surrounding software stack for QPUs ready to be exploited on-premise or from the Cloud.

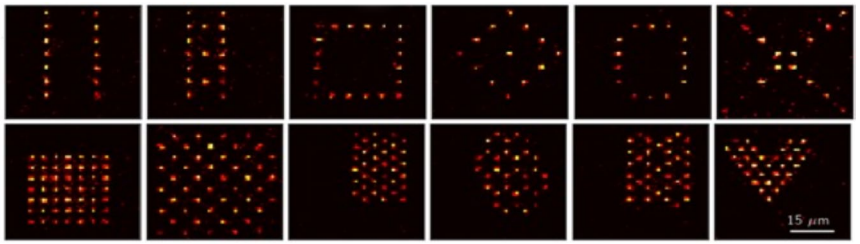
- **Our Team**
  - 120+ people; more than 80 Ph.D. Researchers and Engineers
  - Based on offices worldwide: France (Headquarter), Netherlands, United Kingdom, Canada, USA, …

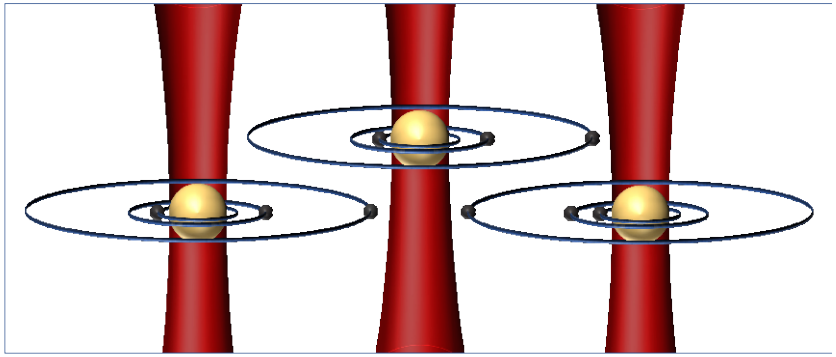
Copyright PASQAL SAS

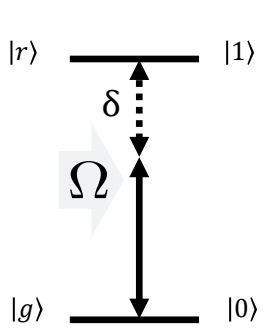
Rydberg atoms

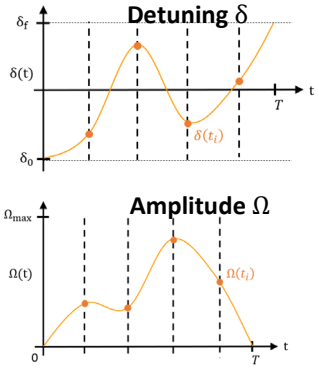


Schlosser et al., Nature (2001)









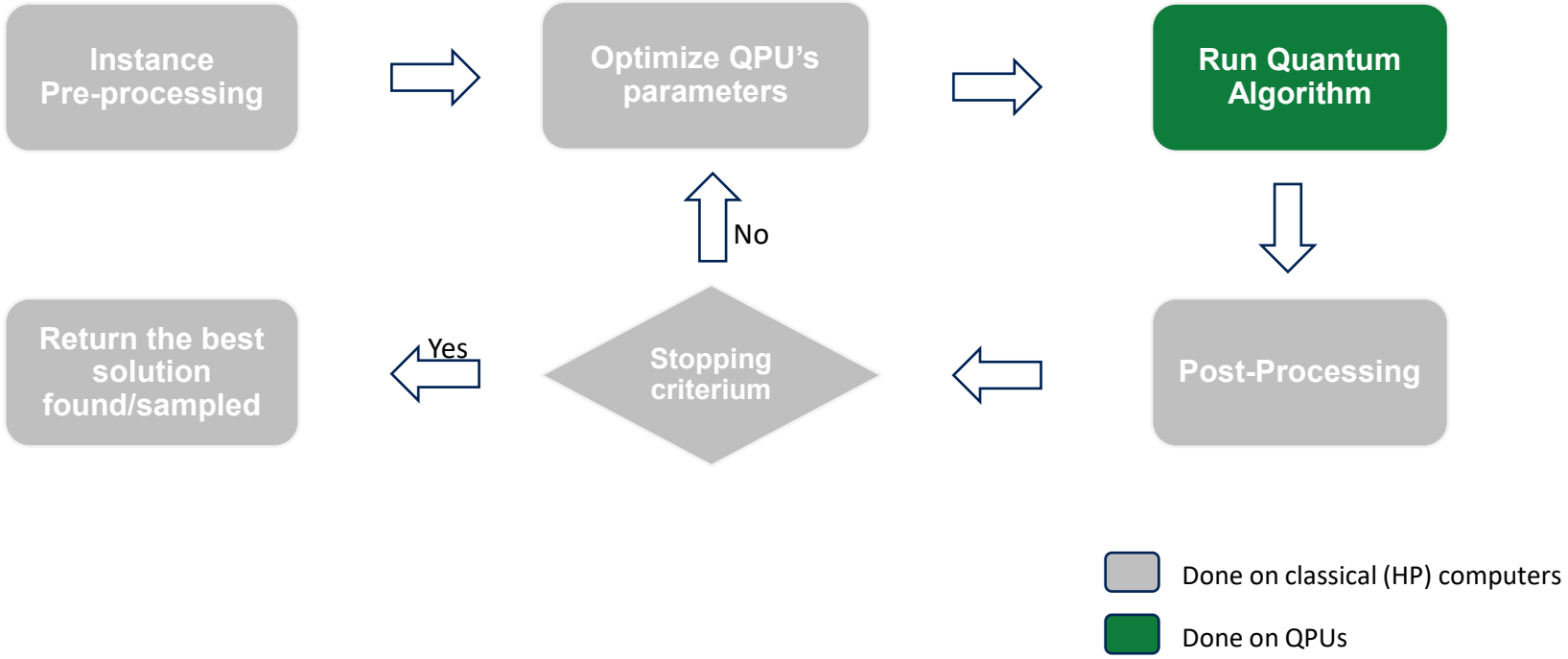
- **Strengths of neutral atoms platform :**
  - *High scalability, tunable connectivity, versatility (different settings)*
- **What can we do today?**
  - *Analog Quantum Computing*
  - *Quantum Simulation*

PASQAL

Copyright PASQAL SAS

How CPUs and QPUs are normally coupled to solve optimization problems

- Many QC algorithms involve sighnificant pre- or post-processing
- Many also include costly close-loop optimization

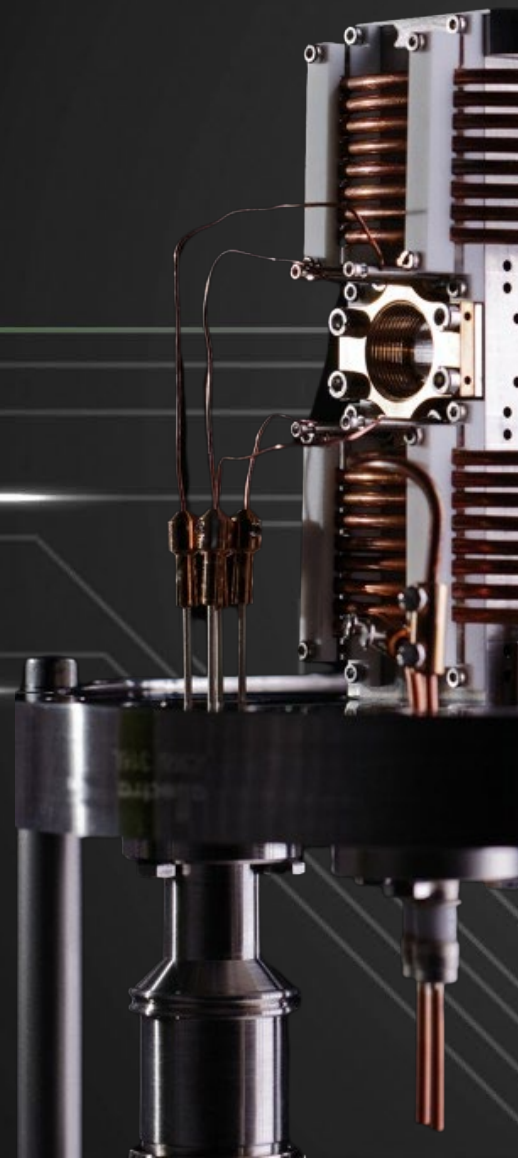


- New hybrid, with more intricate combination of hardware are being developped
- We present here an example of such an approach

Copyright PASQAL SAS

Hybrid Computing

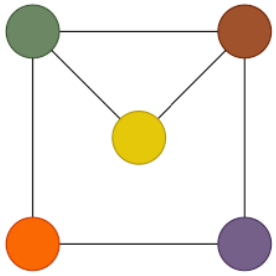
A quantum sampler to speed up classical solvers



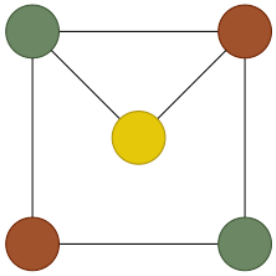
The Minimum Vertex Coloring problem

Let G = (V,E) be a graph with a set V of nodes and a E of edges.
Also, let C be a set of available colors.

The **Minimum Vertex Coloring Problem** (MVCP) consists in **coloring the vertices of G with exactly one color from C** in a such way that **the number of used colors is minimized** while ensuring that **no two adjacent vertices have the same color**.



(a) Trivial coloring.



(b) Optimal coloring.

Graph coloring enjoys many practical applications such as
- Scheduling problems
- Portfolio optimization
- Resources assignment
- Sudoku puzzles

Any sub-set of nodes colored with the same color is an Independent Set* !!

* Independent Set : a sub-set of vertices that are not connected to each other.

Copyright PASQAL SAS

An extended formulation for the Minimum Vertex Coloring Problem

Let **G** be an input **graph** with a **set V of nodes** and **a set E of edges**

Let **S** be the set of **all independent sets** in the graph

Let $\mathbf{y_s}$ **be a binary variable that takes 1 if the independent set s ∈ S is selected**; 0 otherwise.

Let $\mathbf{b_{us}}$ **be a parameter that holds 1 if the vertex u ∈ V** is present in the independent set s ∈ S; 0 otherwise.

$$min \sum_{s \in S} y_s$$

$$\sum_{s \in S} b_{us} y_s = 1, \qquad \forall u \in \mathcal{V}$$

$$y_s \in \{0, 1\}, \qquad \forall s \in S$$

Finding all independent sets in a graph is a very hard task and can be very time and resource-consuming, even for small graphs.

- The number of such sets, and therefore the number of **y** variables, can be extremely large.
- Most of them are not selected in any feasible solution, especially in the optimal one.

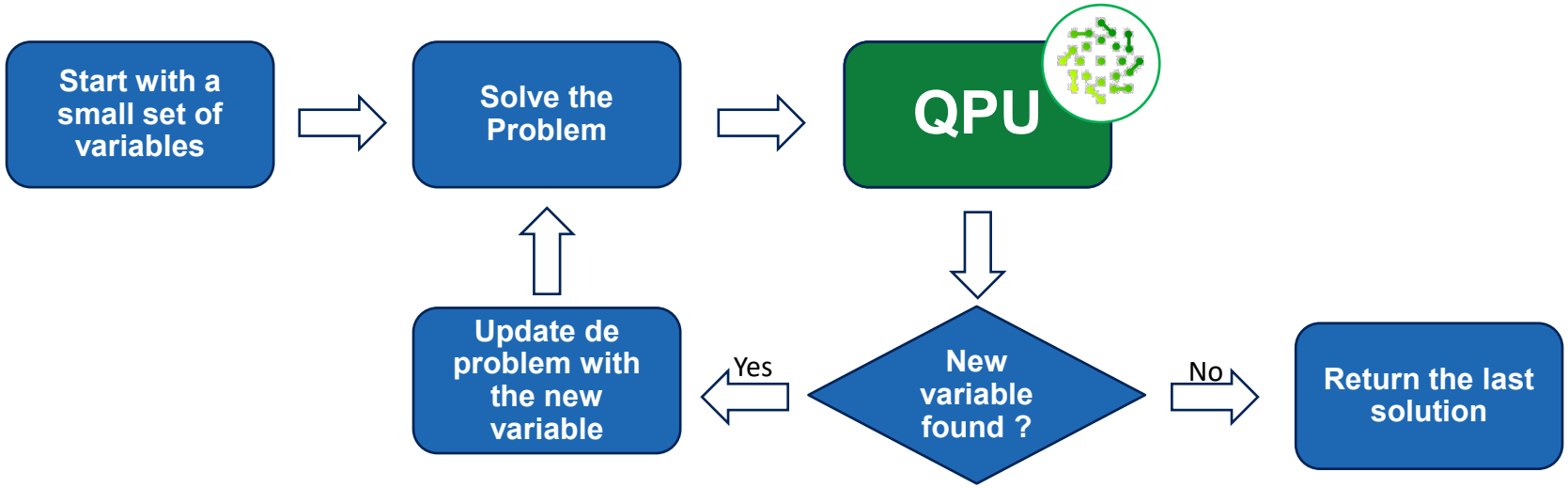
PASQAL

* Independent Set : a sub-set of vertices that are not connected to each other.

Copyright PASQAL SAS

A column generation for the Minimum Vertex Coloring Problem

We don’t need to know all elements (independent sets) in advance



But, how to **efficiently generate new elements** to be converted into variables within the mathematical formulation?

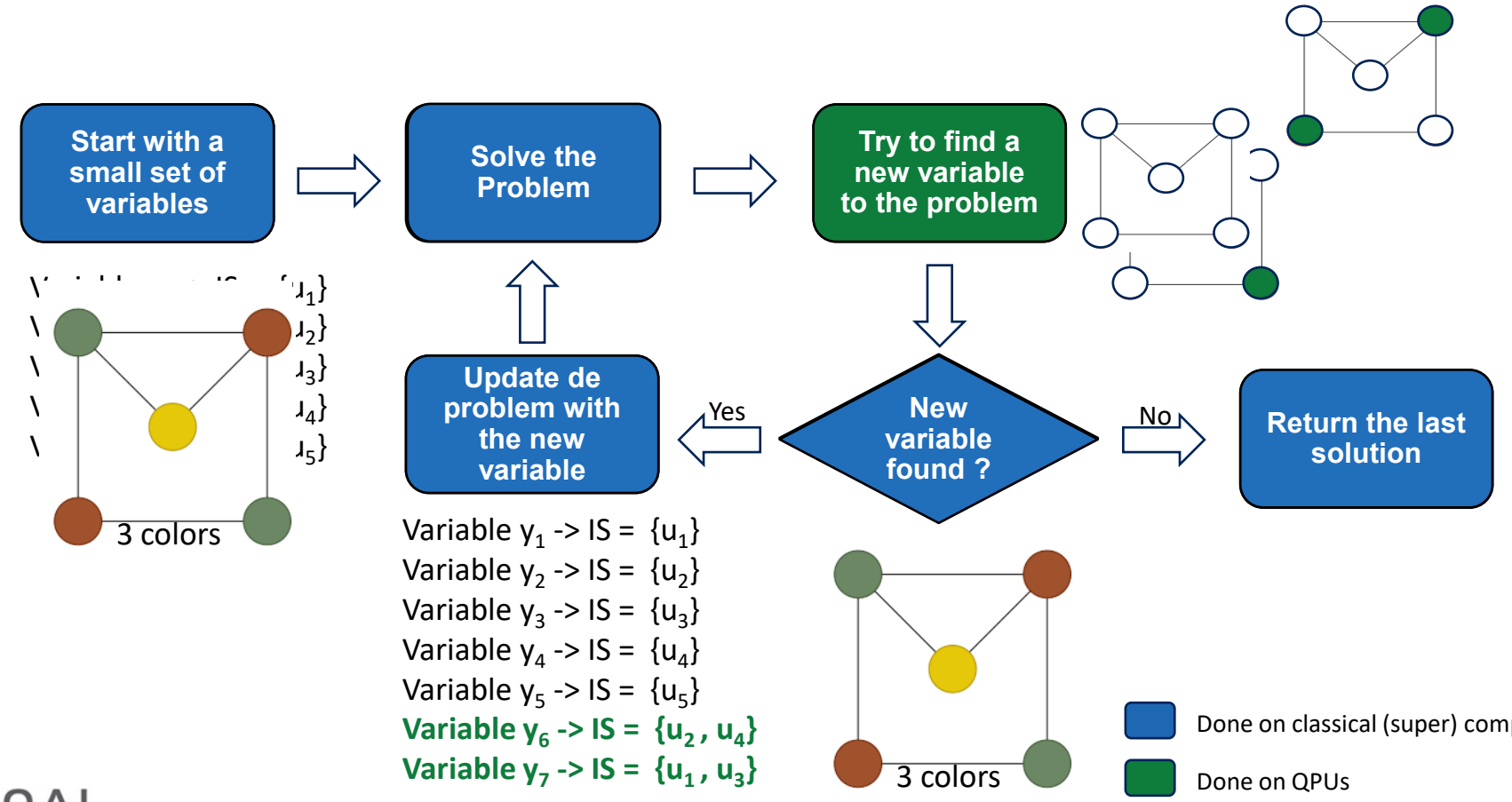
PASQAL

* Independent Set : a sub-set of vertices that are not connected to each other.

Copyright PASQAL SAS

A column generation for the Minimum Vertex Coloring Problem

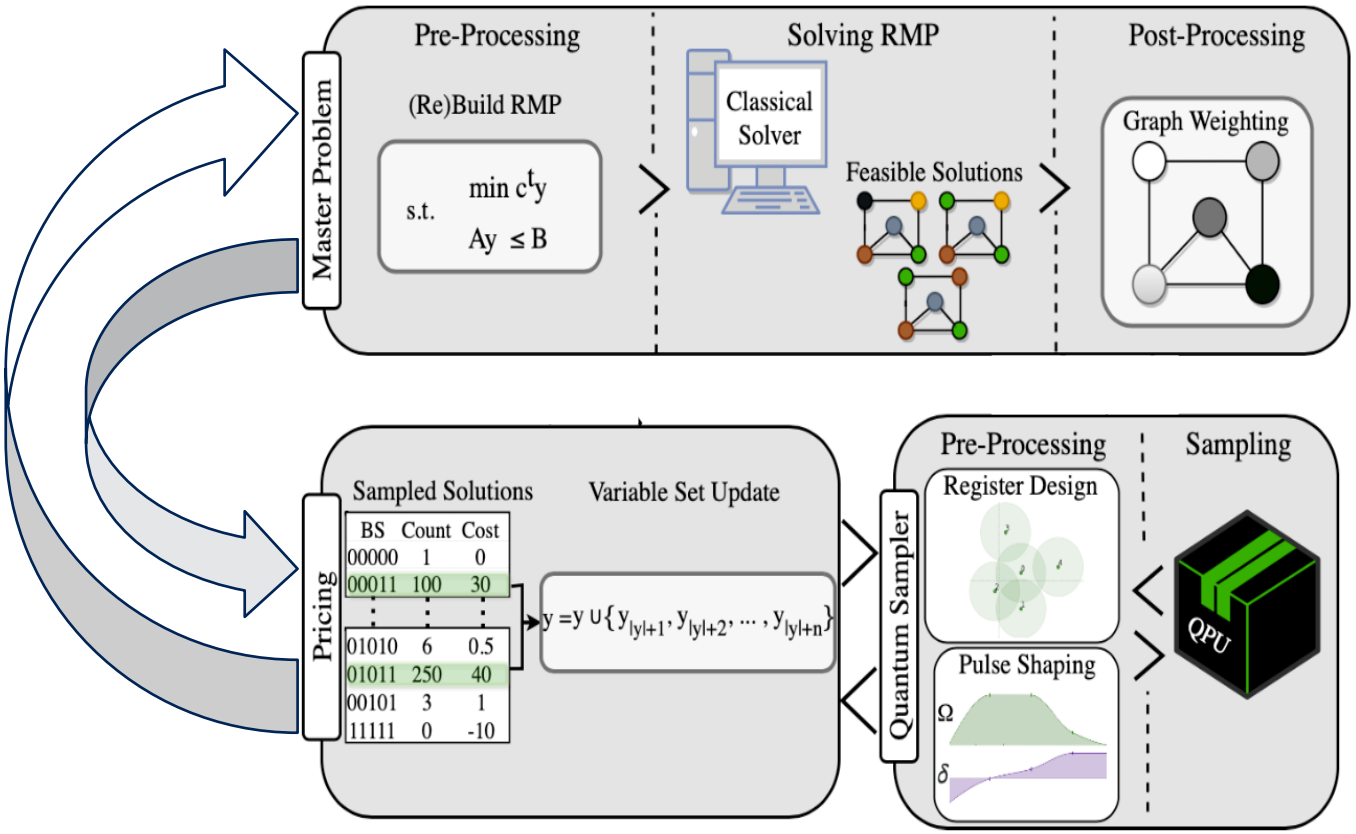
In the proposed hybrid framework, a quantum sampler is specifically tailored to provide new variables to the master problem



Independent Set (IS) : a sub-set of vertices that are not connected to each other.

Copyright PASQAL SAS

An overview of the hybrid column generation framework for solving hard combinatorial problems



Classical Computer

Finds the best coloring with the current set of elements (variables)

Provides instances to the pricing routine

Runs the Atom Register and Pulse Sequence design algorithms

Quantum Computer

Samples good solutions for each pricing sub-problem

Provides new elements to the RMP as new variables

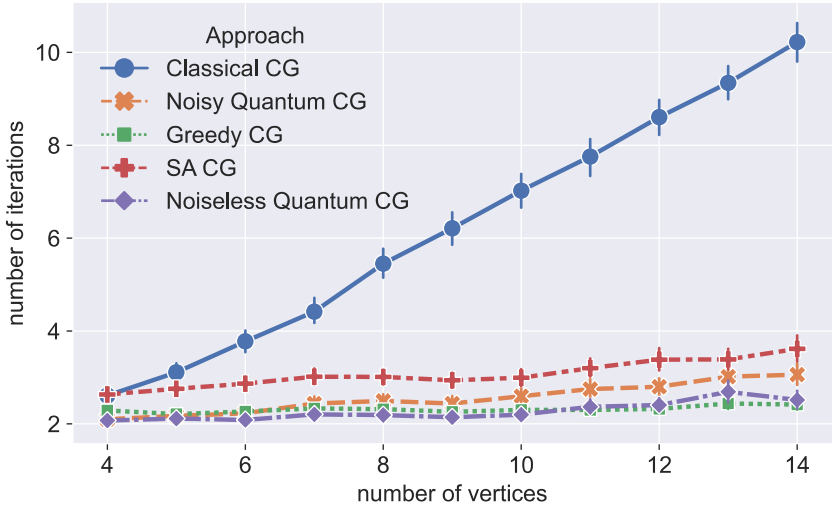
QPU : Quantum Processing Unit
RMP : Reduced Master Problem
BS: Bitstring

Copyright PASQAL SAS

Results

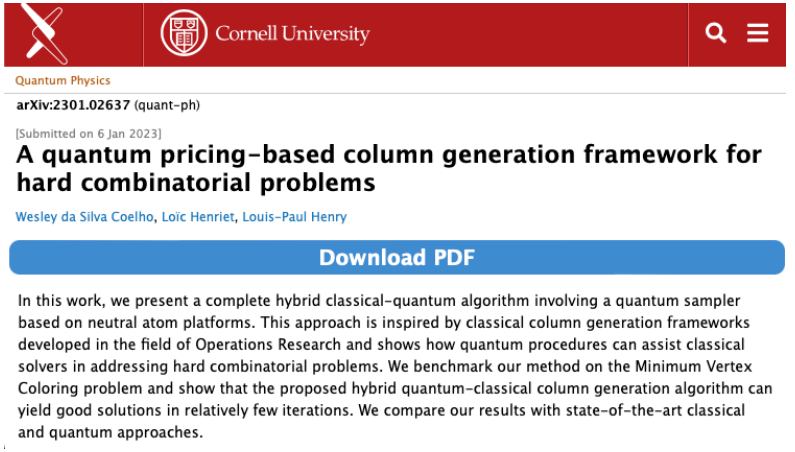
We compared our hybrid approach against different column generation frameworks where the pricing sub-problems were solved by:

- Classical Solver (exactly solving the associated MWIS problem on each pricing instance)
- Stochastic Greedy algorithm
- Simulated Annealing-based sampler



Number of iterations before finding the optimal solution by applying different approaches on different graph orders (from 4 up to 14 vertices), classes (UD and non-UD) and densities (20%, 50% 80%).

For more details, please check our just-published paper


Cornell University

Quantum Physics

arXiv:2301.02637 (quant-ph)

[Submitted on 6 Jan 2023]

A quantum pricing-based column generation framework for hard combinatorial problems

Wesley da Silva Coelho, Loïc Henriet, Louis-Paul Henry

Download PDF

In this work, we present a complete hybrid classical-quantum algorithm involving a quantum sampler based on neutral atom platforms. This approach is inspired by classical column generation frameworks developed in the field of Operations Research and shows how quantum procedures can assist classical solvers in addressing hard combinatorial problems. We benchmark our method on the Minimum Vertex Coloring problem and show that the proposed hybrid quantum-classical column generation algorithm can yield good solutions in relatively few iterations. We compare our results with state-of-the-art classical and quantum approaches.

PASQAL

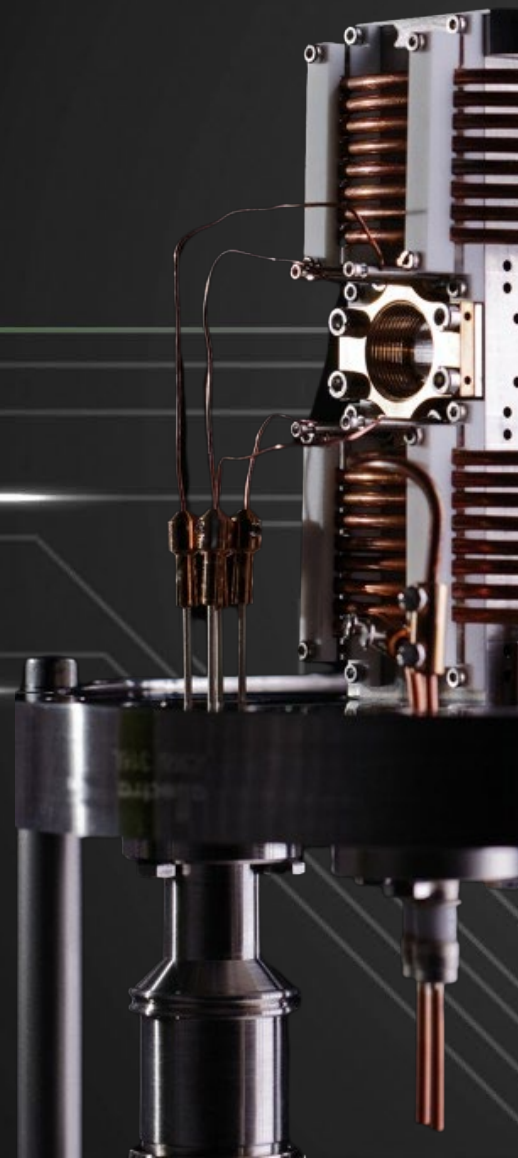
MWIS : Maximum Weighted Independent Set
SA : Simulated Annealing

CG : Column Generation
UD : Unit-Disk

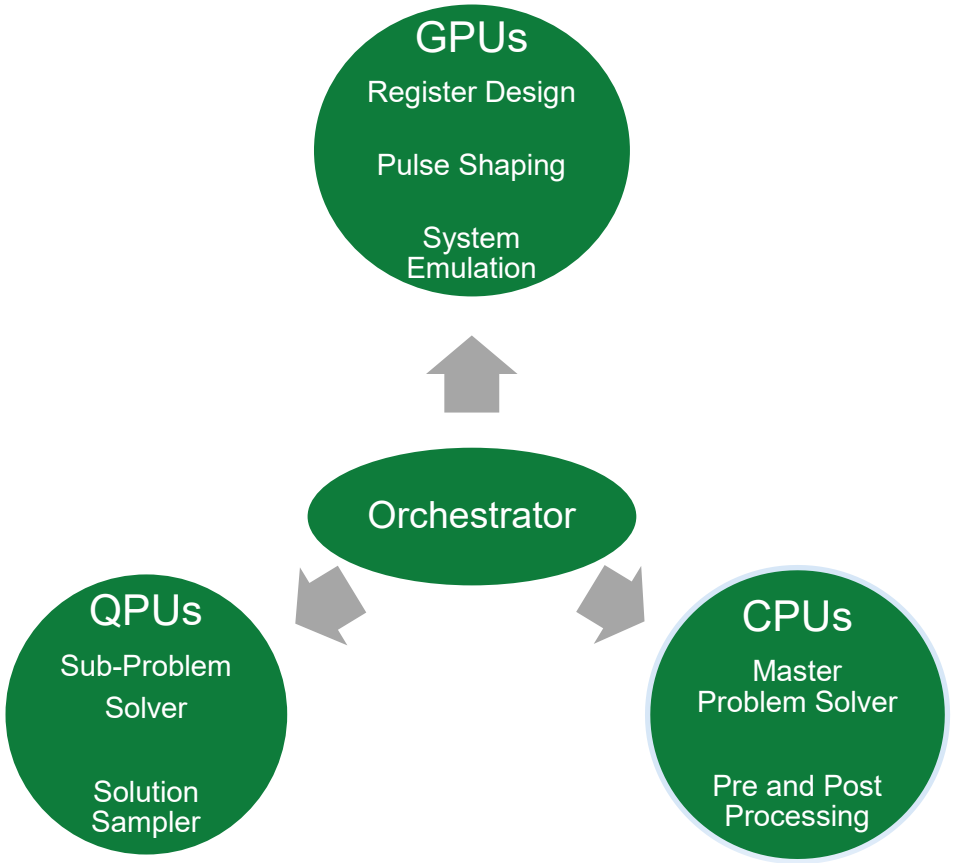
Copyright PASQAL SAS

HPC environment

How to integrate QPUs into HPC frameworks



HPC Environment



- Classical computers (CPUs and GPUs) are still important in solving hard optimization problems
  - They can solve efficiently several of them in a reasonable runtime
  - Import helper in optimizing the QPU's parameters
  - They are useful in emulating quantum systems
  - Still needed to pre- and post-processing the data provided by QPUs
- Quantum computers can solve more efficiently some classes of problems
  - They can be "plugged" to classical computers to help them in solving complex problems
  - They can provide valuable/inaccessible information about the problem

Copyright PASQAL SAS

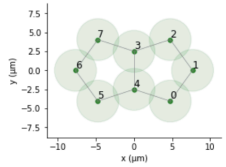
How to program the QPU?

Pulser

https://github.com/pasqal-io

Pulser is a framework for composing, simulating and executing pulse sequences for neutral-atom quantum devices.

```
pentagons_register.draw(draw_graph=True, blockade_radius=a*1.1, draw_half_radius=True)
```

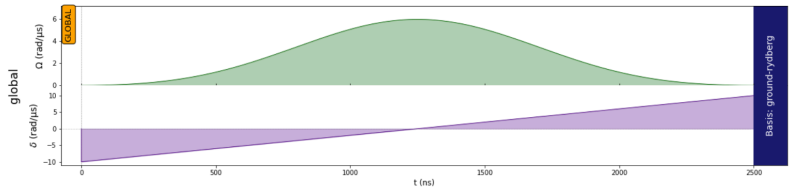



```
duration = 2500 # 2500 ns = 2.5 µs
Omega = 2*np.pi # rad/µs
delta = 10.
```

```
from pulser import Pulse
from pulser.waveforms import BlackmanWaveform, RampWaveform

amp_wf = BlackmanWaveform(duration, Omega)
det_wf = RampWaveform(duration, -delta, delta)

pulse = Pulse(amp_wf, det_wf, phase=0)
```

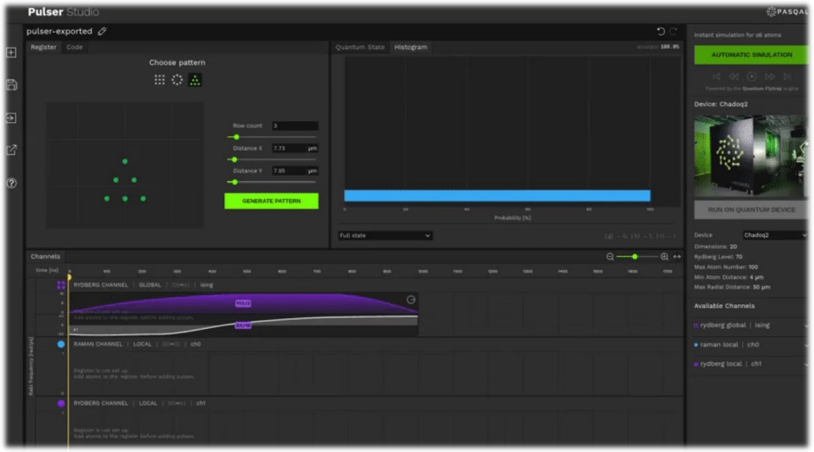
```
from pulser.devices import Chadoq2

device = Chadoq2
```

```
from pulser import Sequence

pentagons_sequence = Sequence(pentagons_register, device)
pentagons_sequence.available_channels

pentagons_sequence.declare_channel('global','rydberg_global')

pentagons_sequence.add(pulse, 'global')

pentagons_sequence.measure(basis="ground-rydberg")


pentagons_sequence.draw()
```



Pulser studio

https://pulserstudio.pasqal.cloud/



- Enables users to graphically build quantum registers and design pulse sequences without coding knowledge.
- The platform allows the creation of novel insights into quantum computing and neutral atoms using an original user experience.
- Pulser Studio includes a built-in emulator that will simulate sequences for small systems directly in the browser.
- These registers and pulse sequences can be executed on quantum processors.
- Pulser Studio is already open and free to corporate and academic users
- A powerful tool to quickly try your ideas

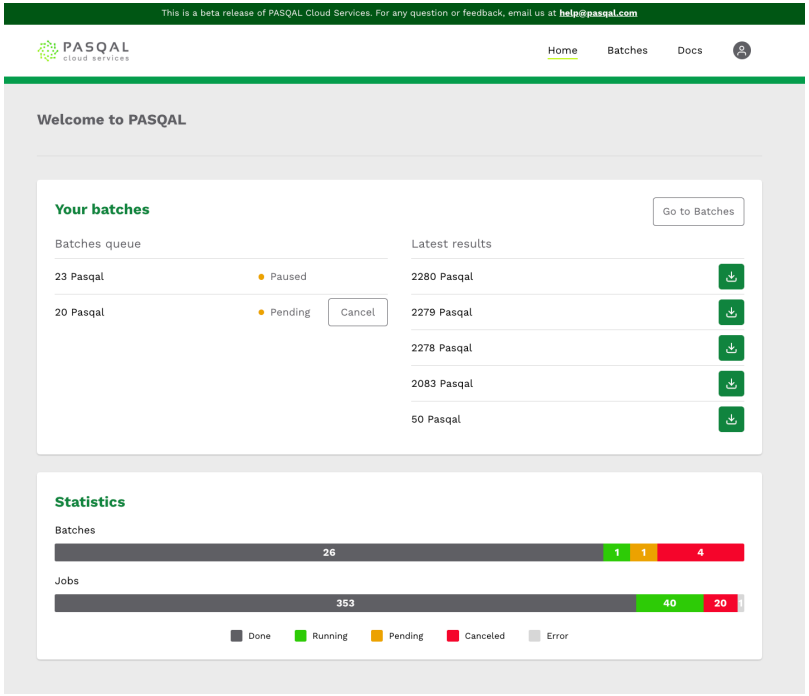
Copyright PASQAL SAS

How to use PASQAL Cloud Services to send Pulser jobs on the QPU?

User Portal

https://portal.pasqal.cloud

- Account & Project administration
- Get API keys
- Monitor jobs
- Download results



Python SDK

https://github.com/pasqal-io/cloud-sdk

- Pulser sequences (jobs) that share common parameters or register layouts can be grouped into a single batch.
- SDK is used to send a batch for execution on the QPU via the cloud platform
- SDK is used to retrieve batch & jobs results

```python
# Define Pulser sequence
reg = Register.square(2, prefix="q")
seq = Sequence(reg, devices.Chadoq2)
seq.declare_channel("raman", "raman_local")
seq.target("q0", "raman")
simple_pulse = Pulse.ConstantPulse(200, 2, -10, 0)
seq.add(simple_pulse, "raman")
sequence_builder = seq.serialize()

# Init connection to PASQAL cloud
endpoints: Endpoints = Endpoints(
    core="https://apis.dev.pasqal.cloud/core",
    account="https://apis.dev.pasqal.cloud/account",
)
client_id = ""
client_secret = ""
sdk = SDK(
    client_id=client_id,
    client_secret=client_secret,
    endpoints=endpoints,
)

jobs = []
for _ in range(10):
    jobs.append({"runs": 10})

# Launch batch and wait for results
batch = sdk.create_batch(
    sequence_builder, jobs, device_type=DeviceType.EMU_FREE, wait=True
)
```


Copyright PASQAL SAS

Hybrid in HPC centers

PASQAL works in close collaboration with HPC centers, in order to develop hybrid platforms

Sold 2 QPUs to be installed directly
into HPC centers
in France and Germany

GENCI
Le calcul intensif au service de la connaissance

TGCC

JÜLICH
Forschungszentrum

Created **QuaTERA** (Quantum Technologies Energy Result Accelerator)
alongside EDF, Exaion Inc. and the Quantum Innovation Zone

"The first open center of excellence to develop sustainable energy solutions using the combined capabilities of HPC and quantum computing"

exaion
GROUPE EDF

ZONES D'INNOVATION QUÉBEC
SHERBROOKE QUANTIQUE

Copyright PASQAL SAS

PASQAL

Thank you

PASQAL
www.pasqal.com
office@pasqal.com
2, av. Augustin Fresnel
91120 Palaiseau
France

Copyright Qu & Co BV