Second Assignment - Hybrid Quantum-Classical Systems

Deadline: 19/12/2024, 10:00

Rules of the game:

- You are supposed to work alone. **Plagiarism will not be tolerated**;
- The use of generative AI for anything different from rephrasing text **is not permitted**;
- Submissions coming after the deadline will not be considered;
- Each solution has to be properly explained, otherwise no points will be granted;
- Unless differently specified, you can use either Qiskit or Pennylane for the implementation of the coding tasks.
- For each submission, we expect to receive the code files and an associated report, in .pdf, using the attached template.
- Code can be submitted either as jupyter notebooks or simple python scripts, including everything that is needed for executing the code (requirements.txt and a README file with everything that is needed).
- **Non-reproducible code will result in 0 points on the exercise**.
- For any question, feel free to reach out by using the TUWEL forum;
- Please submit your solution using the following format: HQCS-NAME-SURNAME-ASSIGNMENT2.{zip, tar, bz2, gz}.

Exercise 1: (15 Pt.)

Let's think about VQA's from a more theoretical standpoint for now. Consider the optimization loop, the applied updates and changes in loss we would like to observe, as well as runtime complexity of the algorithms. Answer the following questions, then.

1. How should the step size during optimization be chosen (i.e., how much should the parameters change in every iteration)? Can it be (a) too small or (b) too big? What

would the consequences (from a loss landscape viewpoint) be if (a) or (b) apply? **[5 Pt.]**

2. Where could variational algorithms provide an advantage over other iterative (gradient-based) algorithms? **[5 Pt.]**

3. Quantum Computing advantages often talk about speed-ups. Do VQA's have a better runtime complexity than classical (gradient-based) algorithms (e.g., classical neural networks or other optimization algorithms have a similar update loop)? If so, how? If not, why not? Is it easy to prove speed-ups for variational methods? **[5 Pt.]**

Exercise 2: (30 Pt.)

The bin packing problem (BPP) is an optimization problem. The final goal is to fit items of different sizes into a finite number of bins, each of a fixed given capacity, minimizing the number of bins used.

The objective function of BPP is to minimize the number of bins used, more formally,

$$\text{min} \quad \sum_{j=1}^{K} B_j \tag{1}$$

$$\text{subject to} \quad g(x_{ij}) = \sum_{j=1}^{K} x_{ij} = 1 \qquad \forall i \tag{2}$$

$$h(x_{ij}) = \sum_{i=1}^{N} w_i x_{ij} \leq C * B_j \qquad \forall j \tag{3}$$

The constraint $g(x)$ implies that each item can be packed into at most one bin, while the constraint $h(x_{ij})$ says that the size of each item allocated to the bins cannot exceed the given C capacity. First, we focus on generating instances of BPP to be used for the evaluation. Each instance is structured as: $[[w_0, w_1, ..., w_n]$, items, B, C], such that: w_i is the weight of item i; C is the capacity of each bin; B is the number of bins; *items* is the number of items.

We generated the instances for you and saved them in the attached binary file `bpp_instances.pkl`. You can read them in your code by using the following code:

```
import pickle
def read_list(filename):
    with open(filename, 'rb') as fp:
        n_list = pickle.load(fp)
    return n_list

loaded_instances = read_list('bpp_instances.pkl')
```

In this task, you are requested to solve the above BPP instances using quantum algorithms by following the steps:

1. Create a Quadratic Model for BPP instances **[5 Pt.]**
2. Transform the BPP quadratic instances to Quadratic Unconstrained Binary Optimization (QUBO) form **[5 Pt.]**
3. Provide the classical solution as a baseline (e.g., cplex, gurobi, etc.) **[5 Pt.]**
4. Provide the solution via Quantum Approximate Optimization Algorithm **[5 Pt.]**
5. Provide the solution via one other variant of QAOA (hint: Warm-start QAOA) **[10 Pt.]**
6. Bonus **[5 Pt.]** Provide the solution via Quantum Annealing (hint: D-Wave)

Evaluation should be performed on the following backends:

- Noiseless simulator (Aer)
- Noisy simulator (choose a noise model)

Both VQE and QAOA should be evaluated using different configurations:

- QAOAAnsatz with $p = [1,3]$
- Optimizers: COBYLA, SLSQP, SPSA
- Shots: {200,400, 600, 800}
- Optimizers' iteration: {250, 500, 750, 1000}

Discuss your solution quality with respect to the classical solution.

Exercise 3: (55 Pt.)

In this task, you are requested to work on the implementation of Grover's algorithm that you saw in the lecture. Your implementation is supposed to work on the provided inputs.

1. Implement a function for the phase shift **[5 Pt.]**
2. Implement a function for the inversion around the mean **[5 Pt.]**
3. Implement a function for the oracle that you saw in the lecture **[5 Pt.]**
4. Using the functions that you implemented in the previous steps, test execution of Grover algorithm for searching a number in a list **[5 Pt.]**
5. Dr. Erwin Zweistein wants to throw a party with all his friends. Since Q1, his telco provider, raised again the prices of phone calls, he devised a new invitation scheme: he calls a friend A, who will call another friend B, and so on until the last of his friends will call him back to tell him that everyone has been invited. **Please note that not all his friends know each other**. Assuming that you know how each of his friends are

connected, use Grover's algorithm to determine whether Zweistein's invitation scheme is feasible for the provided instances. Instances for this exercise can be found in the file `instances-zweistein-invitation`. **[35 Pt.]**

6. Assuming that the invitation scheme worked fine, you should know that Dr. Erwin Zweistein enjoys offering music-themed presents to his guests. Because of this, to encourage a sharing culture among his guests, he decides to distribute a music CD to each of his guest, in a way that people who like each other will get **different** music CD. Conveniently, in the city of Schrödingen, where he and his friends live, people listen only to Classical, Rock, and Jazz. Assuming you know the table plan in advance, use Grover's algorithm to decide how to distribute music CDs to the guests. Instances for this exercise can be found in the file `instances-zweistein-music`. **[10 bonus Pt.]**

The instances are formatted in this way: for the Exercise 3.5, each line contains the phone book of each person. For example:

Erwin Zweistein: 'Niels Bohring', 'Damon Maxwell', 'Sue Entangly'
Marie Curiosity: 'Niels Bohring', 'Sue Entangly', 'Belle Quanta'
Belle Quanta: 'Sue Entangly', 'Marie Curiosity', 'Wanda Wavefunction', 'Richard Funnyman', 'Damon Maxwell'
Sue Entangly: 'Marie Curiosity', 'Erwin Zweistein', 'Richard Funnyman', 'Niels Bohring', 'Damon Maxwell', 'Belle Quanta'
Wanda Wavefunction: 'Niels Bohring', 'Damon Maxwell', 'Belle Quanta'
Damon Maxwell: 'Sue Entangly', 'Wanda Wavefunction', 'Erwin Zweistein', 'Niels Bohring', 'Belle Quanta'
Niels Bohring: 'Sue Entangly', 'Marie Curiosity', 'Wanda Wavefunction', 'Erwin Zweistein', 'Damon Maxwell'
Richard Funnyman: 'Sue Entangly', 'Belle Quanta'

Means that Erwin Zweistein has the phone number of Niels Bohring, Damon Maxwell and Sue Entangly, Marie Curiosity has the number of Niels Bohring, Sue Entangly and Belle Quanta, and so on. Clearly, each person can call only people who are in his/her phone book. A possible solution could be:

Erwin Zweistein $\rightarrow$ Sue Entangly $\rightarrow$ Richard Funnyman $\rightarrow$ Belle Quanta $\rightarrow$ Marie Curiosity $\rightarrow$ Niels Bohring $\rightarrow$ Wanda Wavefunction $\rightarrow$ Damon Maxwell.

On the other hand, for the Exercise 3.6, each line contains the people that each person likes. For example:

Erwin Zweistein: 'Sue Entangly', 'Wanda Wavefunction', 'Belle Quanta', 'Marie Curiosity'
Marie Curiosity: 'Erwin Zweistein', 'Sue Entangly'
Belle Quanta: 'Erwin Zweistein'
Sue Entangly: 'Richard Funnyman', 'Erwin Zweistein', 'Marie Curiosity', 'Niels Bohring'
Wanda Wavefunction: 'Erwin Zweistein'
Damon Maxwell: {∅}
Niels Bohring: 'Sue Entangly'
Richard Funnyman: 'Sue Entangly'

Means that Erwin Zweistein likes Sue Entangly, Wanda Wavefunction, Belle Quanta and Marie Curiosity, while Marie Curiosity likes Erwin Zweistein and Sue Entangly, and so on.

A possible assignment could be:

{'Erwin Zweistein': 'Rock', 'Marie Curiosity': 'Classical', 'Belle Quanta': 'Classical', 'Sue Entangly': 'Jazz', 'Wanda Wavefunction': 'Classical', 'Damon Maxwell': 'Rock', 'Niels Bohring': 'Rock', 'Richard Funnyman': 'Rock'}

In your final report, **you should provide an answer for each one of the provided instances.** Please note that some instances have no valid solution. If that is the case, you have to say in your answer that the instance has no solution.