

# Solution to Second Assignment

19/12/2024

## 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.]**

## Solution 1:

### 1.1

#### Too small:

If the step size is too small, progress through the loss landscape will be slow, causing the algorithm to require more iterations to reach an optimal solution. This increases computational cost and time. From the viewpoint of a loss landscape the algorithm might get stuck in local minima or saddle points because it fails to explore the loss landscape effectively.

#### Too big:

If the step size is too big, there is the risk of overshooting the minimum, causing potential divergence or oscillations as the updates might skip over the optimal solution. This can be a problem especially in narrow or rugged regions of the loss landscape.

**Conclusion:** Choosing a balanced step size is crucial to ensure efficient convergence without getting stuck or diverging.

## 1.2

Variational Quantum Algorithms (VQAs) are well-suited for high-dimensional, non-convex optimization problems often encountered in quantum systems and quantum chemistry. They can provide advantages in specific scenarios:

- Where the problem involves quantum-native structures, for example as we saw in the lecture solving eigenvalue problems for quantum Hamiltonians.
- Where they can access and exploit high-dimensional quantum feature spaces that are inaccessible to classical algorithms, offering better solutions for quantum chemistry, material science, and certain machine learning models.

## 1.3

VQAs are hybrid algorithms that rely on quantum circuits and classical optimization loops. Their runtime complexity depends on factors like the size of the quantum circuit, the number of measurements, and the classical optimization routine. While VQAs are not guaranteed to offer speed-ups for all problems, they might outperform classical methods for problems that map naturally to quantum states, like problems in quantum chemistry, combinatorial optimization, or finding the ground state energy of molecular Hamiltonian.

Proving speed-ups requires assessing the combined system, which makes theoretical guarantees challenging because the quantum component may have advantages for specific problems, such as sampling from quantum states or representing certain distributions, however, the classical optimization part is often bottlenecked by issues like barren plateaus or slow convergence, which can diminish potential speed-ups.

VQAs are problem-specific. Speed-ups, if they exist, depend on: the problem's structure, the size and complexity of the quantum circuit. Therefore, much of the current focus is on demonstrating practical quantum advantages through experiments and simulations.

### **Exercise 2: (30 Pt.)**

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.

### **Solution 2:**

Solutions for Exercise 2 are found in the file `Exercise_2/exercise_2.ipynb`

#### **2.1**

Look at `Exercise_2/exercise_2.ipynb`

#### **2.2**

Look at `Exercise_2/exercise_2.ipynb`

#### **2.3**

Look at `Exercise_2/exercise_2.ipynb`

#### **2.4**

Look at `Exercise_2/exercise_2.ipynb`

#### **2.5**

Look at `Exercise_2/exercise_2.ipynb`

## 2.6

Not implemented

### 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.]

### Solution 3:

#### 3.1

Implemented in `Exercise_3/exercise_3_grover.ipynb` -> `apply_phase_shift()`

#### 3.2

Implemented in `Exercise_3/exercise_3_grover.ipynb` -> `apply_inversion_about_mean()`

### **3.3**

Implemented in `Exercise_3/exercise_3_grover.ipynb` → `oracle()`

### **3.4**

Implemented in `Exercise_3/exercise_3_grover.ipynb` → `grover_search()`

### **3.5**

The problem to solve determining whether Zweistein's invitation scheme is feasible for the provided instances can be reduced to the problem of finding a hamiltonian cycle in a graph. My implementation for it can be found in `Exercise_3/exercise_3_hamilton.py`

### **3.6**