ECMM428 – MSci Research Project

Experimenting with the QAOA on the TSP problem

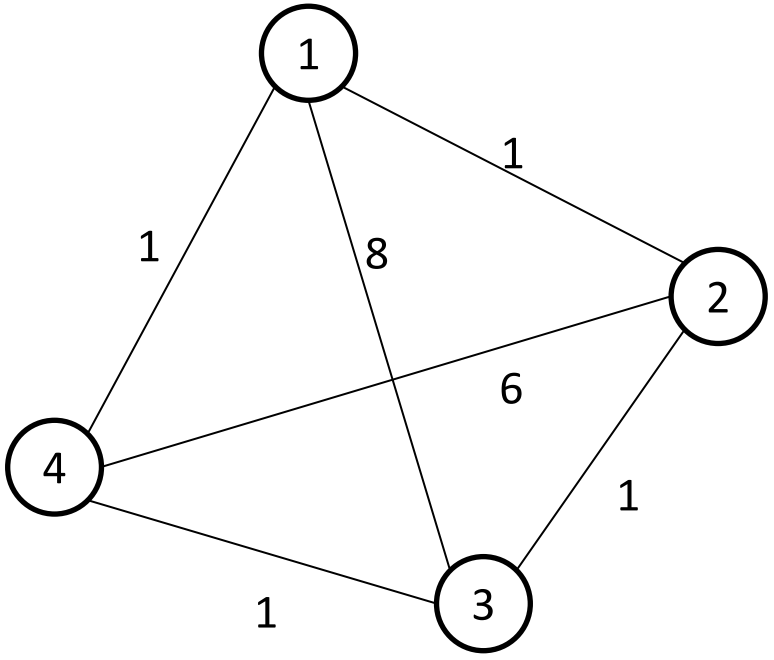
Project Supervisor: Dr. Alberto Moraglio

Table of Content

TRAVELING SALESMAN PROBLEM

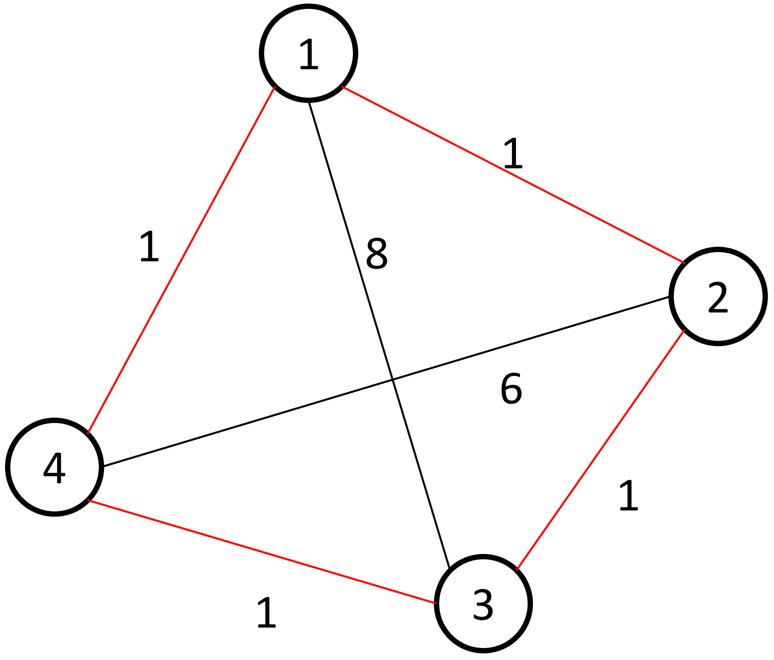
Introduction to the problem

- Given a graph G=(V,E), where V is a list of vertices and E is a list of weighted edges. Find the path with the minimum cost to visit all nodes only once and then return to the start node.
-



Introduction to the problem

- Given a graph G=(V,E), where V is a list of vertices and E is a list of weighted edges. Find the path with the minimum cost to visit all nodes only once and then return to the start node.
- Complexity of N!



TSP NATIVE FORMULATION

QUBO Model

- The Quadratic Unconstrained Binary Optimization Model

$$f(x) = x^T Q x = \sum_{i}^{n} \sum_{j}^{n} Q_{ij} x_i x_j \tag{2.5}$$

TSP Native Formulation

- The Native QUBO formulation of the TSP uses the variables x_{ij} where i is the node and j is the order it occurs in the solution.
- The Native formulation uses 3 constraints

TSP Native Formulation

- *Constraint 1:* Every vertex can only appear once in the solution cycle.

$$c_1 = \sum_{v=0}^{N-1} \left(1 - \sum_{j=0}^{N-1} x_{v,j}\right)^2 \tag{3.1}$$

- *Constraint 2:* For every potential i value in the solution there must be a corresponding i^{th} node.

$$c_2 = \sum_{j=0}^{N-1} \left(1 - \sum_{v=0}^{N-1} x_{v,j}\right)^2 \tag{3.2}$$

- *Constraint 3:* If for the nodes in the expected cycle $x_{u,j}$ and $x_{v,j+1}$ both equal 1, then there should be a penalty if $(uv) \notin E$.

$$c_3 = \sum_{(u,v)\notin E} \sum_{j=0}^{N-1} x_{u,j} x_{v,j+1} \tag{3.3}$$

$$H_A = A(c_1(x) + c_2(x) + c_3(x)) \tag{3.4}$$

TSP Native Formulation

- The Native QUBO formulation also uses a weight function:

$$H_B = B \sum_{(uv)\in E} W_{uv} \sum_{j=0}^{N-1} x_{u,j} x_{v,j+1}$$

- So $H = H_A + H_B$

TSP GPS FORMULATION

TSP GPS Formulation

- The GPS QUBO formulation of the TSP uses the variables x_{ijr} where:
    - $x_{i,j,0} = 1$: The edge (i,j) is not in the tour, and the node i is reached earlier than j.
    - $x_{i,j,1} = 1$: The edge (i,j) is in the tour, and the node i is reached earlier than j.
    - $x_{i,j,2} = 1$: The edge (i,j) is not in the tour, and the node $\boldsymbol{j}$ is reached earlier than $\boldsymbol{i}$.
- The GPS formulation uses 5 constraints

TSP GPS Formulation

- *Constraint 1:* Every vertex can only appear once in the solution cycle.

$$c_1 = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \left(1 - \sum_{r=0}^{2} x_{i,j,r}\right)^2 \tag{3.21}$$

- *Constraint 2:* Each node must be exited once.

$$c_2 = \sum_{i=0}^{N-1} \left(1 - \sum_{j=0}^{N-1} x_{i,j,r}\right)^2 \tag{3.22}$$

- *Constraint 3:* Each node must be reached once.

$$c_3 = \sum_{j=0}^{N-1} \left(1 - \sum_{i=0}^{N-1} x_{i,j,r}\right)^2 \tag{3.23}$$

TSP GPS Formulation

- *Constraint 4:* If vertex i is reached before vertex j, then vertex j is reached after vertex i. This only has to be specified for $r = 2$ due to constraint 1. It also does not apply when $i = j$.

$$c_4 = \sum_{i=1}^{N-1} \sum_{j=i}^{N-1} \left(1 - x_{i,j,2} - x_{j,i,2}\right)^2 \tag{3.24}$$

- *Constraint 5:* This constraint is used to prevent sub-tours. It is only valid if vertex i is reached before vertex j, vertex j is reached before vertex k and vertex i is reached before vertex k. Excluding the cases where $i = j$, $i = k$ and $j = k$.

$$c_5 = \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \sum_{k=1}^{N-1} (x_{j,i,2}x_{k,j,2} - x_{j,i,2}x_{k,i,2} - x_{k,j,2}x_{k,i,2} + x_{k,i,2}) \tag{3.25}$$

TSP GPS Formulation

- The GPS QUBO formulation also uses a weight function:

  - *Distance Objective:*

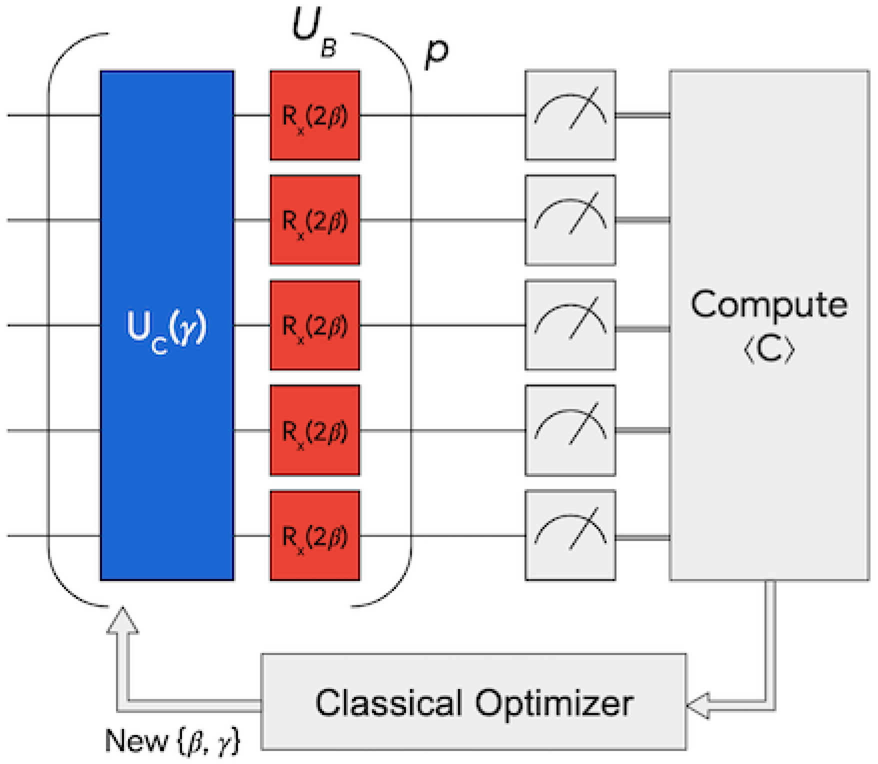
$$H_D = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} W_{i,j} x_{i,j,1} \tag{3.20}$$

- Where $H_A = A(c_1 + c_2 + c_3 + c_4 + c_5), H_B = BH_D$
- $and\ H = H_A + H_B$

QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM

QAOA

- The QAOA uses the Ising Model



(https://quantumai.google/cirq/experiments/qaoa/example_problems)

RESEARCH QUESTIONS

Question 1

- Can the QAOA be used to solve the Native and GPS formulations of the TSP with current quantum technology?

Question 2

- When it comes to solving these problems, how does the performance of QAOA compare to classical algorithms?

AIMS AND OBJECTIVES

Aims and Objectives

- The main aim of the project is to research and experiment with a variety of different methods for solving the TSP.
- The project also aims to test the feasibility of solving the TSP using the QAOA.

DATASETS AND SOFTWARE PLATFORMS

Dataset

- The only dataset used in the project was the Florida State University TSP five_d problem instance. (https://people.sc.fsu.edu/~jburkardt/datasets/tsp/tsp.html)
- five_d has a minimum tour of 19.

Software Platforms

- The project made use of D-Wave Dimod package, and the Qiskit library.
- The project also utilized Qiskits ‘qasm_simulator’ and IBMQs Quantum Sytems, for running the QAOA.

EXPERIMENTS

Experiments

- The project involved conducting 5 experiments.

Experiment 1

- Test the different QUBO formulations using simulated annealing.

Experiment 2

- Test the effect that the QAOA angle parameters have on the energy landscape.

Experiment 3

- The test the effect the depth of the QAOA (p) has on the ability to find solutions.

Experiment 4

- Run the QAOA on the 5-node TSP instance and compare the results to Simulated Annealing.

Experiment 5

- Run the QAOA on an actual quantum machine and analyze the result.

RESULTS

Experiment 1 Results

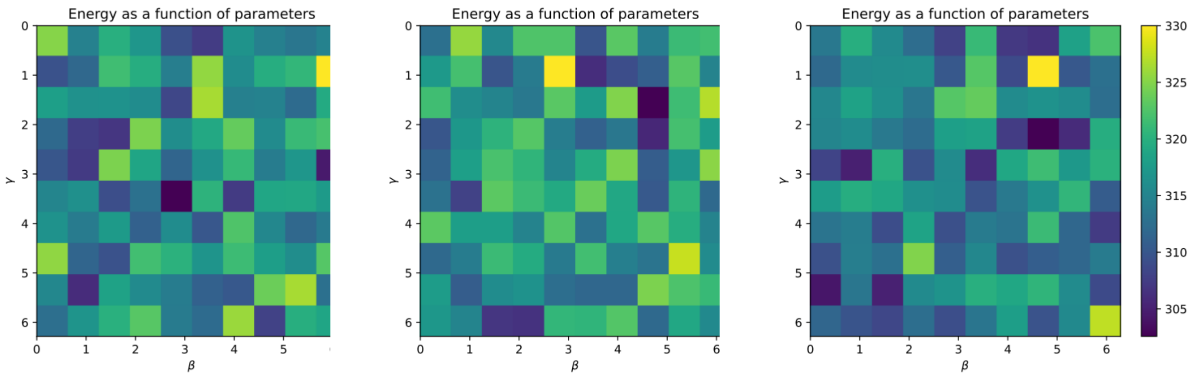
- Test the different QUBO formulations using simulated annealing.

Model	Instance	Time(ms)	Cost	Energy	Validity
GPS	five_d	3.172000	22.000000	26.020833	0.500000
	tsp_3_0	1.856000	12.000000	12.000000	0.000000
	tsp_4_0	2.248500	17.800000	21.825000	0.500000
Native	five_d	2.004300	21.800000	21.800000	0.000000
	tsp_3_0	1.155300	12.000000	12.000000	0.000000
	tsp_4_0	1.271800	18.800000	18.800000	0.000000

Table 4.1: QUBO Formulation Test

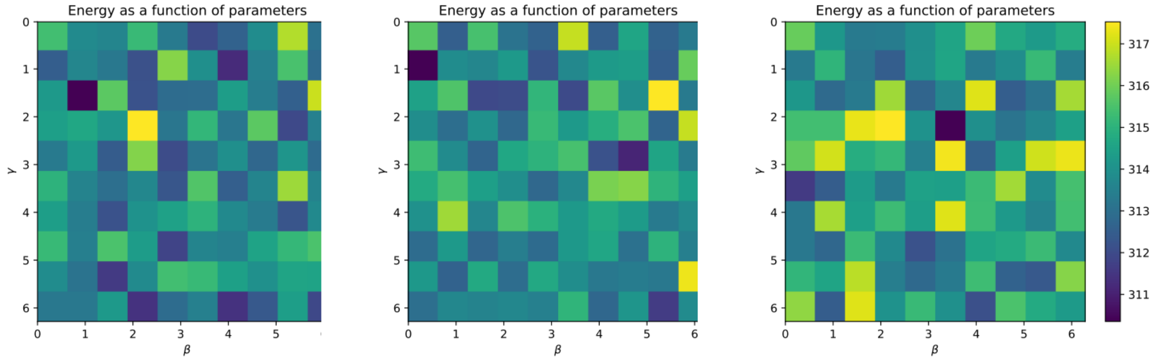
Experiment 2

- Test the effect that the QAOA angle parameters have on the energy landscape.



(a) Energy Landscape 1: Shots = 1000 (b) Energy Landscape 2: Shots = 1000 (c) Energy Landscape 2: Shots = 1000

Figure 4.1: Energy Landscapes: Method=Native, Instance=tsp_3_0, Shots=1000

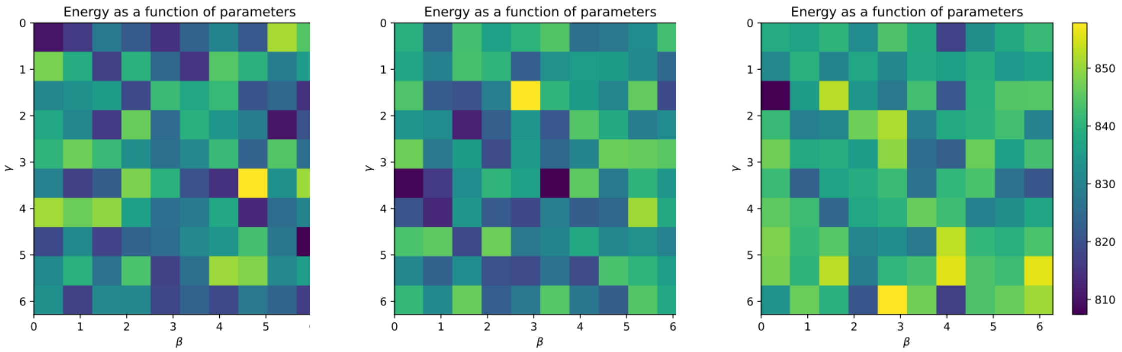


(a) Energy Landscape 1: Shots = 10000(b) Energy Landscape 2: Shots = 10000 (c) Energy Landscape 2: Shots = 10000

Figure 4.2: Energy Landscapes: Method=Native, Instance=tsp_3_0, Shots=10000

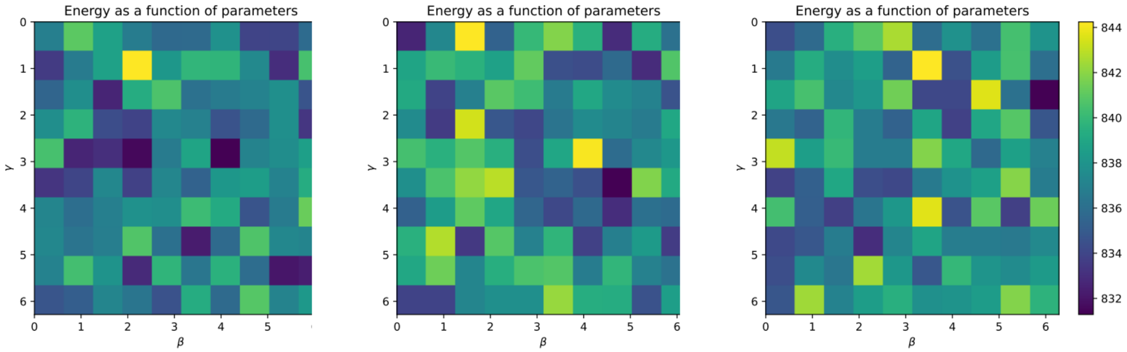
Experiment 2

- Test the effect that the QAOA angle parameters have on the energy landscape.



(a) Energy Landscape 1: Shots = 1000 (b) Energy Landscape 2: Shots = 1000 (c) Energy Landscape 2: Shots = 1000

Figure 4.3: Energy Landscapes: Method=GPS, Instance=tsp_3_0, Shots=1000



(a) Energy Landscape 1: Shots = 10000(b) Energy Landscape 2: Shots = 10000 (c) Energy Landscape 2: Shots = 10000

Figure 4.4: Energy Landscapes: Method=GPS, Instance=tsp_3_0, Shots=10000

Experiment 3

- The test the effect the depth of the QAOA (p) has on the ability to find solutions.

	Model	Instance	p	Time(ms)	Cost	Energy	Validity
1	Native	tsp_3_0	3	8534.007000	12.000000	12.000000	0.000000
2	Native	tsp_3_0	11	26867.112000	12.000000	12.000000	0.000000
3	GPS	tsp_3_0	3	44632.667000	12.000000	36.333333	4.000000
4	GPS	tsp_3_0	11	123478.407000	8.000000	20.166667	2.000000

Table 4.2: Testing p values on QAOA

Experiment 4

- Run the QAOA on the 5-node TSP instance and compare the results to Simulated Annealing.

	Model	Instance	p	Time(ms)	Cost	Energy	Validity
1	Native	five_d	3	82127.823000	0.000000	28.083333	2.000000

Table 4.3: Testing the QAOA on an 5 node instance

	Model	Instance	p	Time(ms)	Cost	Energy	Validity
1	Native	tsp_3_0	3	8534.007000	12.000000	12.000000	0.000000
2	Native	tsp_3_0	11	26867.112000	12.000000	12.000000	0.000000
3	GPS	tsp_3_0	3	44632.667000	12.000000	36.333333	4.000000
4	GPS	tsp_3_0	11	123478.407000	8.000000	20.166667	2.000000

Table 4.2: Testing p values on QAOA

		Time(ms)	Cost	Energy	Validity
Model	Instance				
GPS	five_d	3.172000	22.000000	26.020833	0.500000
	tsp_3_0	1.856000	12.000000	12.000000	0.000000
	tsp_4_0	2.248500	17.800000	21.825000	0.500000
Native	five_d	2.004300	21.800000	21.800000	0.000000
	tsp_3_0	1.155300	12.000000	12.000000	0.000000
	tsp_4_0	1.271800	18.800000	18.800000	0.000000

Table 4.1: QUBO Formulation Test

Experiment 5

- Run the QAOA on an actual quantum machine and analyze the result.

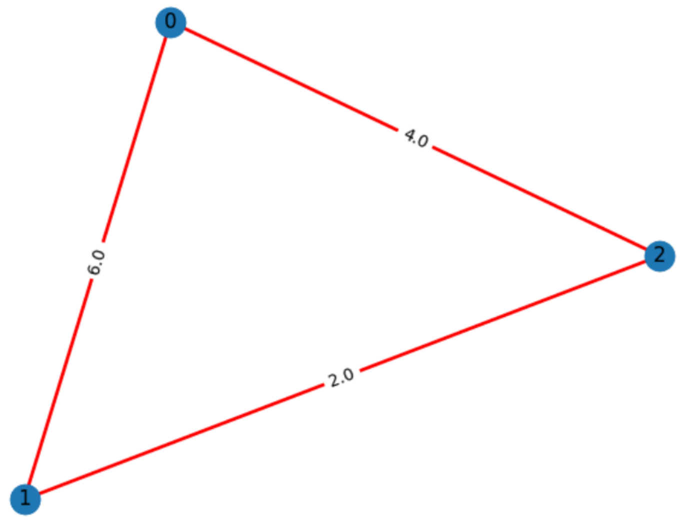


Figure 4.5: QAOA Run On IBMQ

- *Solution Energy:* 12
- *Solution Cost:* 12
- *Time:* 24minutes 25.245749seconds

FUTURE DIRECTIONS

Future Directions

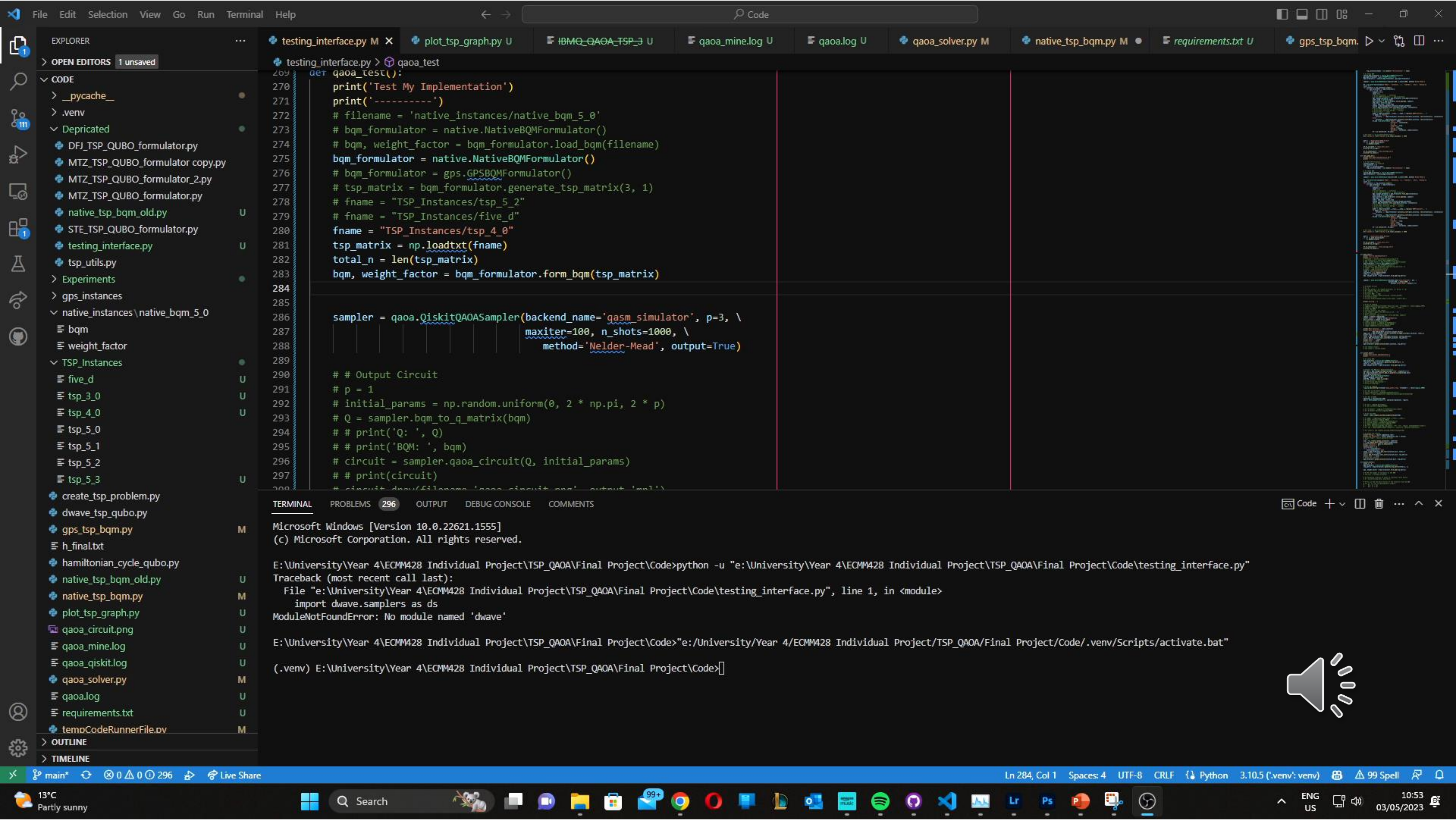
- Hard code the start node variables in the GPS formulation to reduce the overall number of variables.
- Mathematically analyze the solution space of the GPS model to potentially find ways to improve it.
- Run the QAOA on larger quantum machines.
- Take more time, and run more statistically viable tests on the QAOA with the formulations used.

CONCLUSIONS

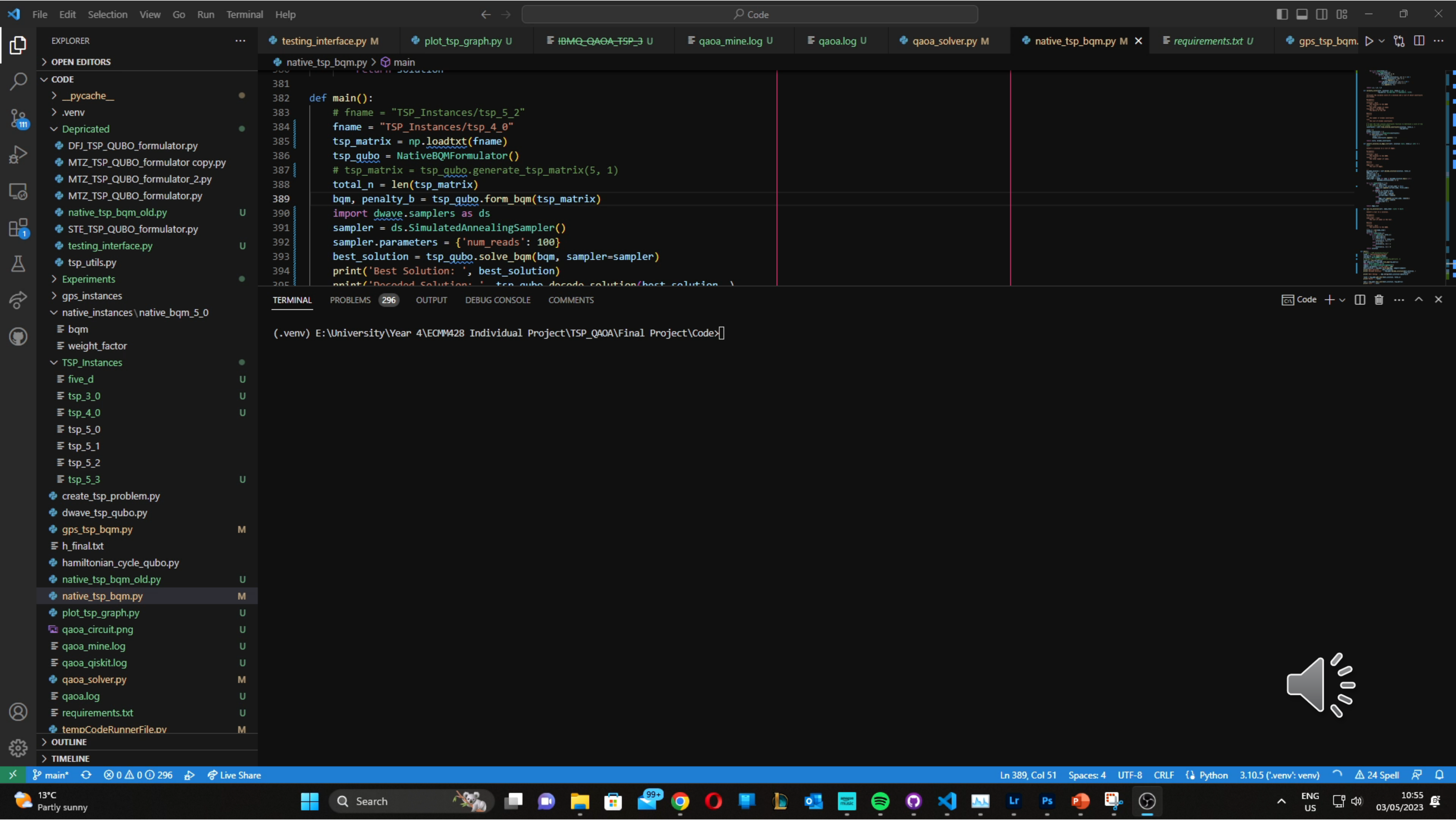
Conclusion

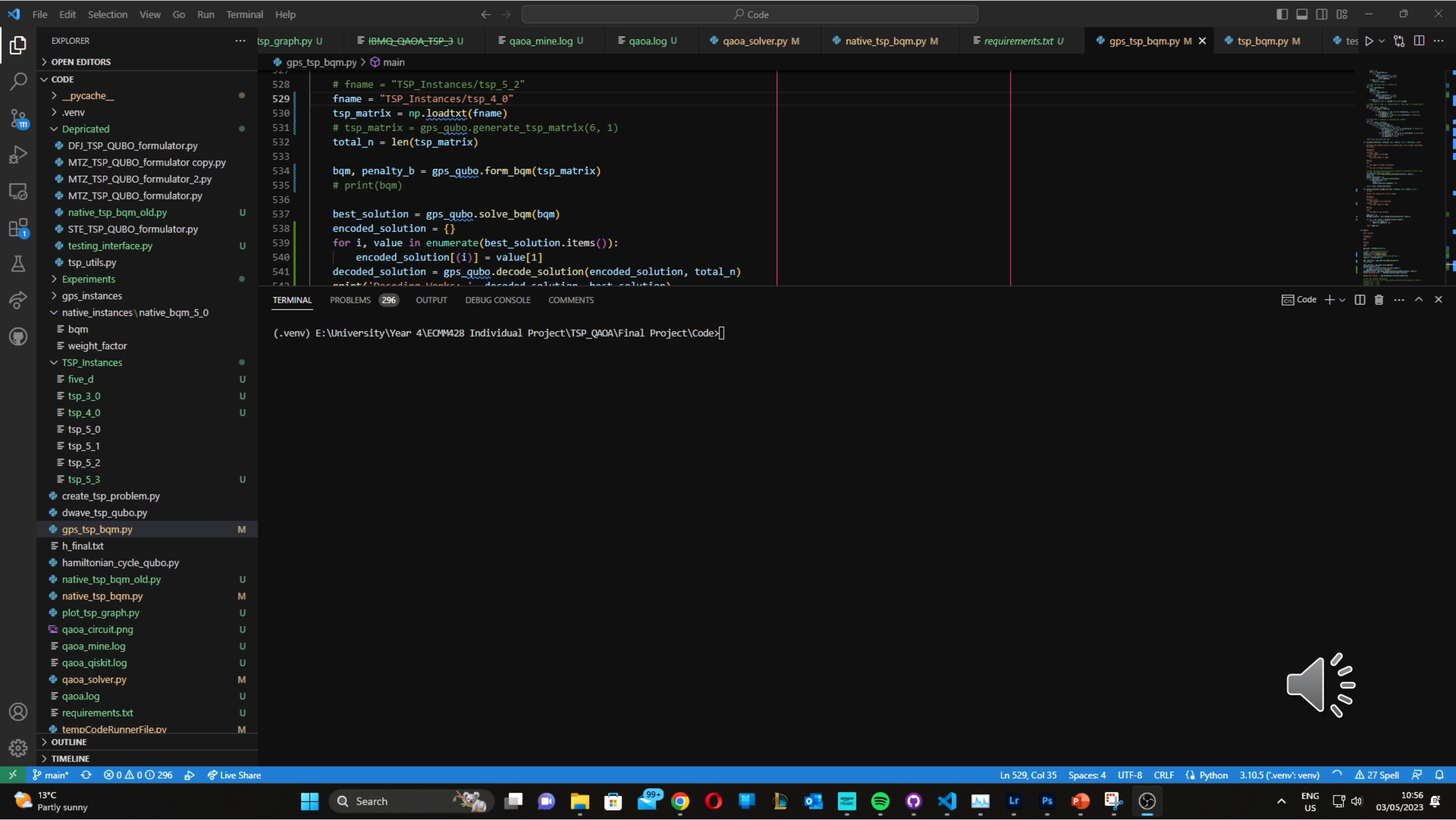
- Overall, I think the development, implementation and testing of the project was a success. The project research questions have been answered and the aims and objectives were achieved.
- In this project I've shown the ability for the QAOA to successfully solve different formulations of the TSP.
- I have also shown that the QAOA works both on quantum simulators and quantum hardware.
- Finally, even though in my testing the QAOA has not shown to improve upon the classical algorithms it will be interesting to see if this changes as the technology to run it develops.

CODE DEMO



testing_interface.py M
plot_tsp_graph.py U
IBMQ_QAOA_TSP_3 U
qaoa_mine.log U
qaoa.log U
qaoa_solver.py M
native_tsp_bqm.py M
requirements.txt U
gps_tsp_bqm.
testing_interface.py > qaoa_test
def qaoa_test():
print('Test My Implementation')
print('----------')
# filename = 'native_instances/native_bqm_5_0'
# bqm_formulator = native.NativeBQMFormulator()
# bqm, weight_factor = bqm_formulator.load_bqm(filename)
bqm_formulator = native.NativeBQMFormulator()
# bqm_formulator = gps.GPSBQMFormulator()
# tsp_matrix = bqm_formulator.generate_tsp_matrix(3, 1)
# fname = "TSP_Instances/tsp_5_2"
# fname = "TSP_Instances/five_d"
fname = "TSP_Instances/tsp_4_0"
tsp_matrix = np.loadtxt(fname)
total_n = len(tsp_matrix)
bqm, weight_factor = bqm_formulator.form_bqm(tsp_matrix)
sampler = qaoa.QiskitQAOASampler(backend_name='qasm_simulator', p=3, \
maxiter=100, n_shots=1000, \
method='Nelder-Mead', output=True)
# # Output Circuit
# p = 1
# initial_params = np.random.uniform(0, 2 * np.pi, 2 * p)
# Q = sampler.bqm_to_q_matrix(bqm)
# # print('Q: ', Q)
# # print('BQM: ', bqm)
# circuit = sampler.qaoa_circuit(Q, initial_params)
# # print(circuit)
EXPLORER
OPEN EDITORS 1 unsaved
CODE
__pycache__
.venv
Depricated
DFJ_TSP_QUBO_formulator.py
MTZ_TSP_QUBO_formulator copy.py
MTZ_TSP_QUBO_formulator_2.py
MTZ_TSP_QUBO_formulator.py
native_tsp_bqm_old.py
STE_TSP_QUBO_formulator.py
testing_interface.py
tsp_utils.py
Experiments
gps_instances
native_instances \ native_bqm_5_0
bqm
weight_factor
TSP_Instances
five_d
tsp_3_0
tsp_4_0
tsp_5_0
tsp_5_1
tsp_5_2
tsp_5_3
create_tsp_problem.py
dwave_tsp_qubo.py
gps_tsp_bqm.py
h_final.txt
hamiltonian_cycle_qubo.py
native_tsp_bqm_old.py
native_tsp_bqm.py
plot_tsp_graph.py
qaoa_circuit.png
qaoa_mine.log
qaoa_qiskit.log
qaoa_solver.py
qaoa.log
requirements.txt
tempCodeRunnerFile.py
OUTLINE
TIMELINE
TERMINAL
PROBLEMS 296
OUTPUT
DEBUG CONSOLE
COMMENTS
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.
E:\University\Year 4\ECMM428 Individual Project\TSP_QAOA\Final Project\Code>python -u "e:\University\Year 4\ECMM428 Individual Project\TSP_QAOA\Final Project\Code\testing_interface.py"
Traceback (most recent call last):
File "e:\University\Year 4\ECMM428 Individual Project\TSP_QAOA\Final Project\Code\testing_interface.py", line 1, in <module>
import dwave.samplers as ds
ModuleNotFoundError: No module named 'dwave'
E:\University\Year 4\ECMM428 Individual Project\TSP_QAOA\Final Project\Code>"e:/University/Year 4/ECMM428 Individual Project/TSP_QAOA/Final Project/Code/.venv/Scripts/activate.bat"
(.venv) E:\University\Year 4\ECMM428 Individual Project\TSP_QAOA\Final Project\Code>
main*
Live Share
Ln 284, Col 1
Spaces: 4
UTF-8
CRLF
Python
3.10.5 ('.venv': venv)
99 Spell
13°C
Partly sunny
Search
ENG
US
10:53
03/05/2023


File Edit Selection View Go Run Terminal Help
testing_interface.py M
plot_tsp_graph.py U
IBMQ_QAOA_TSP_3 U
qaoa_mine.log U
qaoa.log U
qaoa_solver.py M
native_tsp_bqm.py M
requirements.txt U
native_tsp_bqm.py > main
def main():
# fname = "TSP_Instances/tsp_5_2"
fname = "TSP_Instances/tsp_4_0"
tsp_matrix = np.loadtxt(fname)
tsp_qubo = NativeBQMFormulator()
# tsp_matrix = tsp_qubo.generate_tsp_matrix(5, 1)
total_n = len(tsp_matrix)
bqm, penalty_b = tsp_qubo.form_bqm(tsp_matrix)
import dwave.samplers as ds
sampler = ds.SimulatedAnnealingSampler()
sampler.parameters = {'num_reads': 100}
best_solution = tsp_qubo.solve_bqm(bqm, sampler=sampler)
print('Best Solution: ', best_solution)
EXPLORER
OPEN EDITORS
CODE
__pycache__
.venv
Depricated
DFJ_TSP_QUBO_formulator.py
MTZ_TSP_QUBO_formulator copy.py
MTZ_TSP_QUBO_formulator_2.py
MTZ_TSP_QUBO_formulator.py
native_tsp_bqm_old.py
STE_TSP_QUBO_formulator.py
testing_interface.py
tsp_utils.py
Experiments
gps_instances
native_instances \ native_bqm_5_0
bqm
weight_factor
TSP_Instances
five_d
tsp_3_0
tsp_4_0
tsp_5_0
tsp_5_1
tsp_5_2
tsp_5_3
create_tsp_problem.py
dwave_tsp_qubo.py
gps_tsp_bqm.py
h_final.txt
hamiltonian_cycle_qubo.py
native_tsp_bqm_old.py
native_tsp_bqm.py
plot_tsp_graph.py
qaoa_circuit.png
qaoa_mine.log
qaoa_qiskit.log
qaoa_solver.py
qaoa.log
requirements.txt
tempCodeRunnerFile.py
OUTLINE
TIMELINE
TERMINAL PROBLEMS 296 OUTPUT DEBUG CONSOLE COMMENTS
(.venv) E:\University\Year 4\ECMM428 Individual Project\TSP_QAOA\Final Project\Code>
main* Live Share Ln 389, Col 51 Spaces: 4 UTF-8 CRLF Python 3.10.5 ('.venv': venv) 24 Spell
13°C Partly sunny
Search
ENG US 10:55 03/05/2023


File Edit Selection View Go Run Terminal Help
gps_tsp_bqm.py > main
# fname = "TSP_Instances/tsp_5_2"
fname = "TSP_Instances/tsp_4_0"
tsp_matrix = np.loadtxt(fname)
# tsp_matrix = gps_qubo.generate_tsp_matrix(6, 1)
total_n = len(tsp_matrix)
bqm, penalty_b = gps_qubo.form_bqm(tsp_matrix)
# print(bqm)
best_solution = gps_qubo.solve_bqm(bqm)
encoded_solution = {}
for i, value in enumerate(best_solution.items()):
encoded_solution[(i)] = value[1]
decoded_solution = gps_qubo.decode_solution(encoded_solution, total_n)
EXPLORER
OPEN EDITORS
CODE
__pycache__
.venv
Depricated
DFJ_TSP_QUBO_formulator.py
MTZ_TSP_QUBO_formulator copy.py
MTZ_TSP_QUBO_formulator_2.py
MTZ_TSP_QUBO_formulator.py
native_tsp_bqm_old.py
STE_TSP_QUBO_formulator.py
testing_interface.py
tsp_utils.py
Experiments
gps_instances
native_instances\native_bqm_5_0
bqm
weight_factor
TSP_Instances
five_d
tsp_3_0
tsp_4_0
tsp_5_0
tsp_5_1
tsp_5_2
tsp_5_3
create_tsp_problem.py
dwave_tsp_qubo.py
gps_tsp_bqm.py
h_final.txt
hamiltonian_cycle_qubo.py
native_tsp_bqm_old.py
native_tsp_bqm.py
plot_tsp_graph.py
qaoa_circuit.png
qaoa_mine.log
qaoa_qiskit.log
qaoa_solver.py
qaoa.log
requirements.txt
tempCodeRunnerFile.py
OUTLINE
TIMELINE
TERMINAL PROBLEMS 296 OUTPUT DEBUG CONSOLE COMMENTS
(.venv) E:\University\Year 4\ECMM428 Individual Project\TSP_QAOA\Final Project\Code>
main* Live Share Ln 529, Col 35 Spaces: 4 UTF-8 CRLF Python 3.10.5 ('.venv': venv) 27 Spell
13°C Partly sunny Search ENG US 10:56 03/05/2023

Video Link

- Final Presentation.mp4
- https://universityofexeteruk-my.sharepoint.com/:v:/r/personal/stps201_exeter_ac_uk/Documents/University/Year%204/Individual%20Project/Final%20Presentation.mp4?csf=1&web=1&e=LAK265