International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Special Issue of ICRTET-2023, pp 131-135, June-2023

International Conference of Recent Trends in Engineering & Technology (ICRTET-2023)

Flight Gate Allocation Problem Using Hybrid Quantum Solver

[1]Simran Satish Kulkarni, [2]Soumil Krishnanand Heble, [3]Dr. Eric Rotenberg

[1,2,3]Department of Electrical & Computer Engineering, North Carolina State University, Raleigh, USA

***Abstract* - The paper aims to optimize the flight gate allocation problem to ensure maximum utilization of limited number of gates at an airport for a large number of flights. Taking all these variables in consideration and formulating an allocation strategy makes allocation complex, which is why most algorithms consider only one or two of the heuristics for problem formulation and deal with others such as emergencies and changes in real time. These types of allocation problems are usually cast as a graph coloring problem and the most optimum solution yields the desired configuration/allocation. There are classical algorithms that can solve this problem for smaller number of gates and flight combinations. However, as these numbers increase, the computation time for finding the optimum solution increases significantly. The classical algorithms that approach the problem are Genetic algorithm, Greedy algorithm and Backtracking algorithm.**

Keywords: Flight, Gate Allocation, Hybrid Quantum Solver.

1. Introduction

There are classical algorithms that can solve this problem for smaller number of gates and flight combinations. However, as these numbers increase, the computation time for finding the optimum solution increases significantly. The classical algorithms that approach the problem are Genetic algorithm, Greedy algorithm and Backtracking algorithm.

The heuristics involved in the allotment process are:

- Ground time for aircrafts
- Space and service requirements
- Walking distance for passengers
- Gates assigned to specific airlines
- Emergencies or Change in schedules

Taking all these variables in consideration and formulating an allocation strategy makes allocation complex, which is why most algorithms consider only one or two of the heuristics for problem formulation and deal with others such as emergencies and changes in real time. These types of allocation problems are usually cast as a graph coloring problem and the most optimum solution yields the desired configuration/allocation.

2. Quantum Approach to Problem Statement

The Flight Gate Allocation (FGA) problem can be mapped to a graph $G(V, E)$, where the vertices (V) denote the flights on ground and the edges (E) denote the overlap between the ground time of the flights. One approach towards solving the graph is graph multi-coloring, where the colors denote the gates. Therefore, the number of colors is equivalent to the number of gates available. The number of colors are calculated by calculating the chromatic number $\chi(G)$. The upper bound for this value is given by $\chi(G) \leq 2m + 1$, where m is the number of edges in the graph.

QUBO and Graph Coloring

In this project, we implement a method to formulate a cost function for a k-coloring graph using combinatorial optimizations, that can further be used to solve the flight gate allocation. Then we derive a Hamiltonian matrix from this cost function which can be solved by a quantum computer to find the minimum value of the matrix. A variational quantum eigen solver (VQE) algorithm is used to solve the eigen values of the Hamiltonian matrix H and binary optimization problems. The H matrix is used to convert an optimization problem into an adiabatic quantum computation. The desired optimal solution is calculated by finding the minimum energy state corresponding to the lowest eigenvalue.

With increasing number of nodes, the graph multi-colouring problem becomes an NP-hard problem. One approach to solving it is a heuristic algorithm which works on a trial and error approach running on a hybrid quantum computer. For this purpose, quadratic unconstrained binary optimization (QUBO) is a technique that can be used to formulate an equation to represent any problem.

$$J = min\{X^T QX + g^T X + c\} = min\{X^T HX + c\}$$

Where Q is a square matrix with real numbers of the order n and X is a state vector with element of $\{0, 1\}^n$.

For our graph $G(V, E)$, $V(G) = \{1, 2, ..., n\}$ and $E(G) = \{e_{ij}\}$, let $x_{ij} = 1$ if color j has been assigned to node i. Applying this constraint,

$$\sum x_{ij} = 1,\ for\ i = \{1, 2, ...n\},\ where\ n = no.\ of\ nodes$$

The second constraint for our problem would be that adjacent vertices cannot be assigned the same color.

$$x_{ip} + x_{jp} =\leq 1,\ p = \{1, 2...K\},\ where\ K = no.\ of\ colors$$

Applying the above two constraints to the equation (1), we get the QUBO optimization problem. We may add penalties P, where P is a very large positive number.

Therefore, the cost function for the graph multi-coloring is formulated as below:

$$min\{X^T QX\} = \sum(P(x_{ij} - 1)^2 + Px_{ik})$$

x_{ij} is node i with color j and x_{ik} are the nodes connected to each other but should not have the same color.

Therefore, we get new set of variables $(x_1, ..x_{n*k}) = (x_{11}, x_{12}, ..., x_{1k}, x_{21}, ..., x_{n1}, . . . , x_{nk})$, where $n_* k$ gives us number of qubits required to solve the problem and each node is represented by k qubits.

Casting Problem as a Graph Coloring Problem

To cast the flight gate allocation problem to a graph coloring problem let us assume a small example. Figure 1 shows a time bar graph where there are 6 flights (A to F) that are required to be assigned one of 3 gates at an airport.

We map this to a graph where we assume that each aircraft is a node and the flights whose timings overlap have an edge between the nodes that represent them. So, we get a graph similar to one in Figure 1.

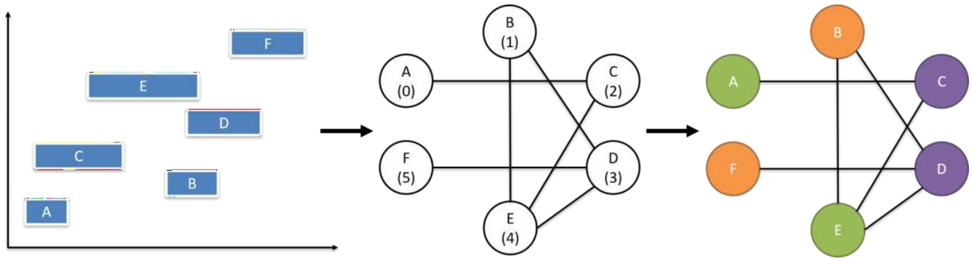


Figure 1: Flight Gate Allocation Example

Let the flights $A\{, B, C, D, E, F\}$ be expressed by $\{0, 1, 2, 3, 4, 5\}$ and the index i represents the flights. Let the gates be expressed by $\{0, 1, 2\}$ and the index j represents the gates. The number of qubits required for this computation is the number of flights * number of gates = 6 * 3 = 18 Qubits. Assuming x_{ij} to be the binary variable which denotes that flight i has been assigned the gate j. For simplicity let us assume that the binary variables $\{x_{00}, x_{01}, x_{02}, x_{10}, ..., x_{52}\}$ be represented by

$$x\{0, x_1, x_2, x_3, ..., x_{17}\}$$

The first constraint is that a node/flight must be only allocated one gate/color, the following expression represents that:

$$P\sum_{i=0}^{5}(1 - \sum_{j=0}^{2} x_{ij})^2,\ where\ P\ is\ the\ penalty\ value$$

Using a generic equation:

$$P(1-(x_a+x_b+x_c))^2 = P(x_a^2+x_b^2+x_c^2+2x_a^2*x_b^2+2x_a^2*x_c^2+2x_b^2*x_c^2-2x_a-2x_b-2x_c)$$

Where x_a, x_b, x_c denotes the binary variables for a flight having a particular color.

Since x is a binary variable $x^2 = x$, the equation becomes,

$$P(2x_a*x_b+2x_a*x_c+2x_b*x_c-x_a-x_b-x_c)$$

Using the above equation for each of the subsets {xa, xb, xc} −→ {0, 1, 2}, {3, 4, 5}, {6, 7, 8}, {9, 10, 11} {12, 13, 14}{15, 16, 17} and assuming P = 4, we get:

$$(8x0*x1+8x0*x2+8x1*x2-4x0-4x1-4x2)+(8x3*x4+8x3*x5+8x4*x5-4x3-4x4-4x5)$$

$$+(8x6*x7+8x6*x8+8x7*x8-4x6-4x7-4x8)+(8x9*x10+8x9*x11$$

$$+8x10*x11-4x9-4x10-4x11)+(8x12*x13+8x12*x14+8x13*x14-4x12-4x13-4x14)$$

$$+(8x15*x16+8x15*x17+8x16*x17-4x15-4x16-4x17)$$

The second constraint is that adjacent nodes/flights must not be assigned the same colors/gates, the following expression represents that constraint:

$$P\sum_{(i,k)\epsilon\ E}\left(\sum_{j=0}^{2}(x_{ij}*x_{kj})\right)$$

Expanding the constraints we get:

(4x0 ∗x6) + (4x1 ∗x7) + (4x2 ∗x7) + (4x3 ∗x9) + (4x4 ∗x10) + (4x5 ∗x11) + (4x3 ∗x12) + (4x4 ∗x13) + (4x5 ∗x14) + (4x6 ∗x12)

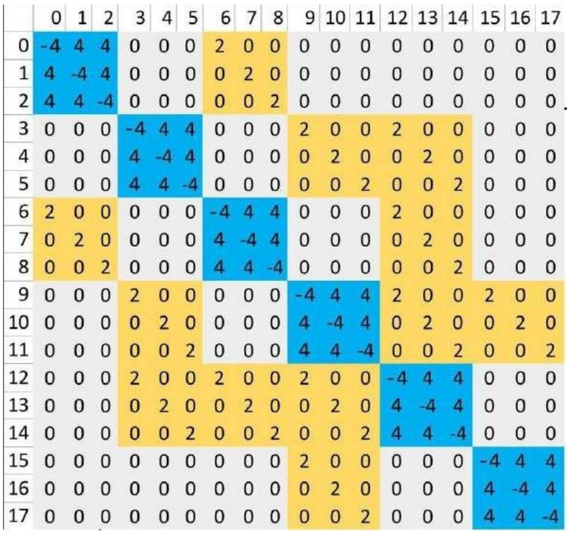
+(4x7 ∗x13) + (4x8 ∗x14) + (4x9 ∗x12) + (4x10 ∗x13) + (4x11 ∗x14) + (4x9 ∗x15) + (4x10 ∗x16) + (4x11 ∗x17)

From the above expression(s) we can extract the coefficients of the Q matrix as:

0.5 ∗ coefficient of xm ∗ xn = Q(m, n) = Q(n, m)

And coefficient of xm = Q(m, m)

Using the above method we can construct the Q matrix, as in Figure 2.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	-4	4	4	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0
1	4	-4	4	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
2	4	4	-4	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
3	0	0	0	-4	4	4	0	0	0	2	0	0	2	0	0	0	0	0
4	0	0	0	4	-4	4	0	0	0	0	2	0	0	2	0	0	0	0
5	0	0	0	4	4	-4	0	0	0	0	0	2	0	0	2	0	0	0
6	2	0	0	0	0	0	-4	4	4	0	0	0	2	0	0	0	0	0
7	0	2	0	0	0	0	4	-4	4	0	0	0	0	2	0	0	0	0
8	0	0	2	0	0	0	4	4	-4	0	0	0	0	0	2	0	0	0
9	0	0	0	2	0	0	0	0	0	-4	4	4	2	0	0	2	0	0
10	0	0	0	0	2	0	0	0	0	4	-4	4	0	2	0	0	2	0
11	0	0	0	0	0	2	0	0	0	4	4	-4	0	0	2	0	0	2
12	0	0	0	2	0	0	2	0	0	2	0	0	-4	4	4	0	0	0
13	0	0	0	0	2	0	0	2	0	0	2	0	4	-4	4	0	0	0
14	0	0	0	0	0	2	0	0	2	0	0	2	4	4	-4	0	0	0
15	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	-4	4	4
16	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	4	-4	4
17	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	4	4	-4

Figure 2: Q Matrix

3. Quantum Implementation

To find a solution to our problem we used existing methods provided by Qiskit Aqua and Aer.

Experimental Framework

The experimental framework is as follows, we use network x to create a graph which takes in an edge list to construct the graph. The graph information is used to construct a weight matrix which is used for constructing the cost function. A custom python method was written that takes in a list of the binary variables, the number of nodes (flights), number of colors (gates), the weight matrix and a penalty value (an arbitrarily large positive value) to return a linear expression that is the cost function for the problem.

We then minimize this cost expression and use docplex to generate an ising hamiltonian to use with the variational quantum eigensolver algorithm. The docplex method call returns an object that contains the hamiltonian/operator and an offset which is to be offset from the expectation. We made some progress in understanding how the docplex method works, it carefully dissects and reduces the cost function to generate a list of weighted pauli matrices that are used to construct the Hamiltonian. This method has some limitations on the type of cost function it can take in to generate the Ising Hamiltonian.

A solution to the problem was searched using the exact eigensolver algorithm from Qiskit Aqua. For the Quantum Simula- tion we used the VQE method provided by Qiskit Aqua. The SPSA optimizer was used as the classical optimization algorithm to be run on a classical computer. A circuit of depth 5, made of parameterized RY gates is used with linear entanglement which means only qubits i and *i + 1* are entangled. A VQE class object was created with the library and simulated with the QASM simulator to run 1024 shots and display the solution using max cut library.

Execution

The three experiments run generated the solution using the exact eigensolver but only experiment 1 was able to run and generate the solution using the qasm simulator and it took about 162 seconds to run. All the experiments were also run on the IBM QASM Backend (IBMQ) but the simulation did not yield the results there as well. THe IBMQ account showed that the simulation was completed but the histogram results did not agree with the eigen solver results even for the experiment 1 result which generated correct solution when run on local system.

Inputs for the Problem Statement

We tested our algorithm and script with three different experiments varying from three nodes to six nodes (which is the problem explained above). Table 1 describes the parameters of the graph and the coloring problem.

Table 1: Experiments Run

Experiment #	Nodes (Flights)	Colors (Gates)	Penalty Value	Graph Figure	Experiment Figure
1	3	2	3	3	4
2	4	3	4	5	6
3	6	3	4	7	8

4. Results and Analysis

This section contains the figures that describe the experiment graph and the graph solution output by the script. The qubit sets for particular nodes are marked with a yellow bar above. Qubits having same configuration have the same color and the graphs are colored according to the results obtained. The algorithm as designed colors adjacent nodes with different colors which means it assigns different gates to different flights. The bar chart presents flights which require a gate allocation and have conflicting time requirements.

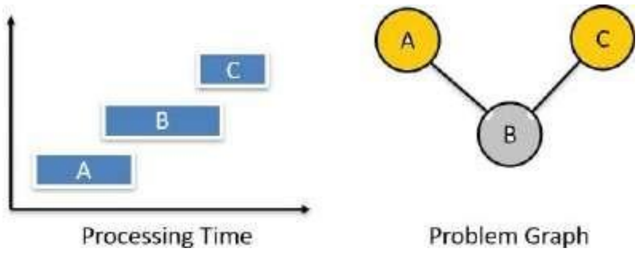


Figure 3: Experiment 1 Graph

```
Solution: [0. 1. 1. 0. 0. 1.]
```

Figure 4: Experiment 1 Result

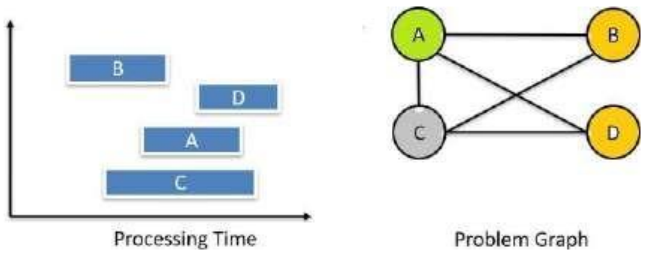


Figure 5: Experiment 2 Graph

```
Solution: [0. 1. 0. 1. 0. 0. 0. 0. 1. 1. 0. 0.]
```

Figure 6: Experiment 2 Result

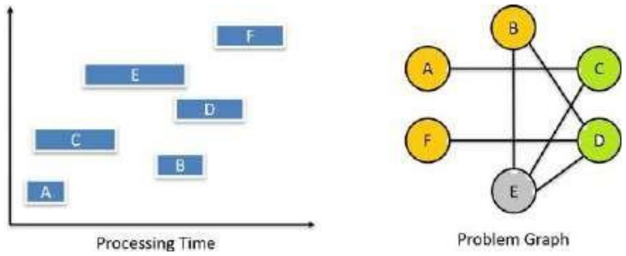
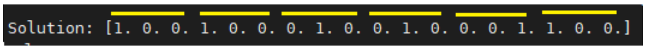


Figure 7: Experiment 3 Graph



```
Solution: [1. 0. 0. 1. 0. 0. 0. 1. 0. 0. 1. 0. 0. 0. 1. 1. 0. 0.]
```

Figure 8: Experiment 3 Result

5. Conclusion

This project presents an approach towards attempting to solve the flight gate allotment problem. For smaller, less connected graphs we may use classical algorithms such as Greedy or Genetic or Backtracking algorithms. However, the approach we propose is more useful for solving larger, more complex graphs based on the flight's time on the ground. We derive a cost function from the graph coloring problem by applying the appropriate constraints. We derive a Hamiltonian matrix where the elements represent the energy levels of the states. A variational quantum eigensolver (VQE) algorithm is operates on the Hamiltonian matrix to derive the lowest energy state, obtaining the most optimal solution to the gate allocation problem. We noticed that with the growing number of qubits the quantum simulator is unable to produce a solution for the problem but all of the experiments produced the desired solution when solved using the Exact Eigensolver Algorithm.

REFERENCES

[1] Fred W. Glover and Gary A. Kochenberger, "A tutorial on formulating QUBO models", CoRR abs/1811.11538 (2018).

[2] Andrew Lucas, "Ising formulations of many np problems", Frontiers in Physics2 (2014), 5.

[3] Young-Hyun Oh, Hamed Mohammadbagherpoor, Patrick Dreher, Anand Singh, Xianqing Yu, and Andy J. Rindos, "Solving Multi-Coloring Combinatorial Optimization Problems Using Hybrid Quantum Algorithms", arXiv e-prints (2019), arXiv:1911.0059.

[4] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J.Love, Aln Aspuru-Guzik, and Jeremy L. OBrien, "A variational eigenvalue solver on a photonic quantum processor, Nature Communications" (2014), no. 1.

Citation of this Article:

Simran Satish Kulkarni, Soumil Krishnanand Heble, Dr. Eric Rotenberg, "Flight Gate Allocation Problem Using Hybrid Quantum Solver" in proceeding of International Conference of Recent Trends in Engineering & Technology ICRTET - 2023, Organized by SCOE, Sudumbare, Pune, India, Published in IRJIET, Volume 7, Special issue of ICRTET-2023, pp 131-135, June 2023.

****