EXPLORING QUANTUM COMPUTING POTENTIALS IN SOLVING A COMBINATORIAL OPTIMIZATION PROBLEM TO MINIMIZE EXPOSURE TO COVID-19 DURING A CITY JOURNEY

A. Nourbakhsh[1], M. Jadidi[1*], K. Shahriari[2]

[1] Geomatics Engineering, Lassonde School of Engineering, York University, 4700 Keele Street, Toronto, ON, M3J 1P3, Canada
[2] Intelligent Data Analytics, Toronto, ON, M9A 1R9, Canada

Commission IV, WG IV/8

KEY WORDS: Vehicle routing problem, Quantum computing, Quadratic unconstrained binary optimization, Quantum annealing

ABSTRACT:

COVID-19 is an airborne virus that can be spread directly or indirectly from one person to another. Spreading the virus strongly depends on the location and time and hence, a Spatio-temporal event. Moreover, traffic congestion will increase the spread of the virus not only because of the vicinity but also because of increased temperature and humidity in these spaces for a short or long time. This paper introduces a vehicle routing optimization model to reduce COVID-19 exposure risk during a city journey by solving it as a quadratic unconstrained binary optimization problem on a quantum annealing computer. Indeed, the objective of the COVID-19 prevention optimization problem is to minimize the risk of exposure for a given set of road users between origins and destinations. Microsoft Taxi data from the city of Beijing have been used to simulate road users' movement. The problem has been run onto three different solvers. One of the solvers is executed on classical computers, and two other solvers are executed on hybrid quantum solvers. Hybrid solvers return the solution within less than 0.03 seconds on quantum processing unit time. However, the results will be returned at least 5 seconds after the execution in the classical solver. It is worth mentioning that, as there is no direct access to the quantum computers, it is hard to compare the results on the same scale as the queries will go on a queue in D-wave quantum computers. Applying the proposed model on the trajectory data shows a better distribution of the vehicles on the road network.

1. SUMMARY OF THE RESEARCH

1.1 Introduction

According to the World Health Organization (WHO), approximately 198.9 million people were infected by COVID-19 in 2021 worldwide, among which 3.5 million lost their lives (WHO, 2021). This infection had adverse mental health effects on the society and largely impacted the economy. Controlling the spread of the virus by reducing the exposure to it, is a way to diminish the noxious health and wealth impacts of it. Therefore, this is one of the major tasks that should be taken care of by both individuals in their every-day decision-making; and longer-term, larger-scale guidelines, and regulations by public and authorities decision-makers. To our knowledge, COVID-19 is an airborne virus which can be spread directly or indirectly from one person to another. The direct transmission is by breathing the exhale air of infected nearby persons with distant of less than 1.5 meter apart (WHO, 2020). In the indirect way, an infected persons contaminates objects by touching and the virus last on the surfaces for a while depending the type of material and environmental conditions. Therefore, spreading the virus strongly depends on the location and time and hence, a spatio-temporal event. Moreover, traffic congestion will increase the spread of the virus not only because of the vicinity (Cartenì et al., 2021) but also because of increased temperature and humidity in these spaces for a short or long time (Shi et al., 2020, Tosepu et al., 2020, Xie and Zhu, 2020).

Solving a routing problem involves both individual and public decision makers; one decides the time and the trajectory for their journey in the city from an origin to a destination. In the public level, the transportation availability, guidelines and regulations impacts directly the decisions in individual level. In a pandemic, additional cost functions and constraints should be added to penalize and restraint the exposure to hot zones to attenuate the spread of the disease. The solutions to this NP-hard problem based on classical computers are well-known and already exist (Chen et al., 2020, Lenstra and Kan, 1981). However, they are not fast enough while complexity increased to provide real-time or close to real-time response to neither individual nor public level problems. Based on the current technological developments in Quantum Computers and the very promising results in solving complex problems, they can be the answer to solving large scale routing problems in real-time (Somma et al., 2012, Gibney, 2019).

* Corresponding author

1.2 Problem definition

The core of this study is the Vehicle Routing Problem (VRP), which is one of the well-known NP-hard combinatorial optimization problems in the classical computers. Many heuristic and meta-heuristic algorithms have been proposed to solve the VRP problem in classical computers. However, these algorithms still have trouble solving such problems in real-world applications. On the other hand, quantum computation and, more specifically, adiabatic quantum computation are promising near-term meta-heuristic approaches to solve combinatorial optimization problems.

To be able to solve the problem on current quantum computer hardware, the problem should be transformed to a quadratic unconstrained binary optimization (QUBO). Finding the min-

https://doi.org/10.5194/isprs-archives-XLIII-B4-2022-419-2022

imum value of QUBO is a known NP-hard problem as the Ising model (Neukart et al., 2017). Therefore, this research proposes a Spatio-temporal model that can be converted to the QUBO, and as a result, it can be run on a quantum computer.

1.3 Contribution

The main contribution of this research is to introduce an optimization model to reduce COVID-19 exposure risk by solving it as a QUBO problem on a QA computer. Indeed, the objective of the COVID-19 prevention optimization problem is to minimize the risk of exposure for a given set of road users between origins and destinations. Cost function on an individual segment is defined based on a quadratic function with two parameters. The first parameter is the number of COVID-19 cases for two nodes of a segment, and the second is the number of road users on each segment at a specific time interval. The main challenge in this study is to investigate transforming a real-world discrete geospatial routing problem into a quantum computer. To illustrate the proof of the concept, Microsoft Taxi data from city of Beijing have been used to simulate road users movement.

2. BACKGROUND

There are four main computational models in quantum computing; 1) Gate-based/circuit-based models(Michielsen et al., 2017), 2) Adiabatic models(Albash and Lidar, 2018), 3) Measurement based models(Briegel et al., 2009), and 4) Topological models(Roy and DiVincenzo, 2017). In this research, Quantum Annealing (QA) is being used. Since QA is a sub-category of adiabatic quantum models, this section first describes adiabatic models and then provides details of the QA. Generally, adiabatic quantum models rely on the adiabatic theorem in quantum mechanics(Kato, 1950). The main characteristic of the adiabatic models that makes them appropriate for solving optimization models is the energy of the system. The energy of the system consists of kinetic and potential energy. In order to define the energy of the adiabatic models in a system, the Hamiltonian method(Nagaj, 2012) has been used (Eq 1). Assuming that the quantum computer system has N qubits and are set up to the ground state of H_0. H_0 should be transferred to a new Hamiltonian that is called final Hamiltonian (H_f).

$$H(t) = (1 - t)H_0 + tH_f \tag{1}$$

where t is time and it is gradually increasing from 0 to 1. H_0 is an initial Hamiltonian, and H_f is the final Hamiltonian.

QA is a meta-heuristic approach that uses the nature of adiabatic quantum models to find the low-energy of the system. QA does not apply to all problems. The functional problems can be divided into two parts. In some applications, the main goal is to find minimum energy; on the other hand, in other applications, the purpose is to find all low-energy samples. The first problems are called optimization problems, and the second one is called probabilistic sampling problems.

The concepts of energy in physics can bring the optimization models into an energy optimization problem by using the fact that everything tries to reach a minimum energy level. For example, Each object around us slides downhill (local optimal). QA uses a similar concept in quantum physics to determine the low-energy of a problem and finds the optimal or near-optimal combination of all possibilities. However, in probabilistic sampling problems, the system tries to find samples from many low-energy states and creates a landscape of the energies. This approach is suitable for machine learning problems since it helps build a probabilistic model of the real world. In order to better understand the scope of this research, an overview of the above-mentioned details is shown on the Fig1.

In general, to formulate an optimization problem to be able to run on a QA computer, based on the nature of the problem one of the following classes should be used:

- Binary Quadratic Models (BQM)(Kochenberger et al., 2014)
- Constrained Quadratic Models (CQM)(Scokaert and Rawlings, 1998)
- Discrete Quadratic Models (DQM) (Hogg, 2003)

In this paper, BQM is used. Two main formulas are mostly used to define the objective function of the problem: The Ising model and Quadratic Unconstrained Binary Optimization (QUBO). Conversion between these two formulas is trivial. The following section describes the proposed method by using QUBO method.

3. PROPOSED METHODOLOGY

This research proposes a Spatio-temporal model that helps decreasing exposure to the Covid-19 virus by reducing traffic congestion and number of Covid-19 cases in each area. The case study of the proposed model is Beijing city. Data layers that are being used in this research contain COVID-19 cases and vehicle's trajectory. The final results of the proposed model is compared to the current state of the city. The evaluation will be done by comparing two different states, with and without using the proposed model.

Quantum computers can be classified based on different technologies and different computational models. In this research, a D-Wave quantum computer is being used. D-Wave quantum computers are built based on a QA model, a subcategory of the adiabatic models. Adiabatic quantum computation models and specifically QA hardware are promising near-term meta-heuristic approaches to solve combinatorial optimization problems. QA is a meta-heuristic algorithm that finds the minimum of a discrete cost function. The problem should be formulated as either an Ising problem or a QUBO problem to take advantage of current QA computers such as D-Wave. These two models are convertible to each other, and the QUBO is often simpler to handle mathematically (Eq 2).

$$Costfunction = x^T.Q.x \tag{2}$$

where x = an N x 1 vector of binary variables
Q = an N x N upper-triangular matrix of real weights

Eq 2 can be expressed more precisely as Eq 3.

$$CostFunction = \sum_i Q_{i,i}x_i + \sum_{i<j} Q_{i,j}x_i x_j \tag{3}$$

https://doi.org/10.5194/isprs-archives-XLIII-B4-2022-419-2022 |

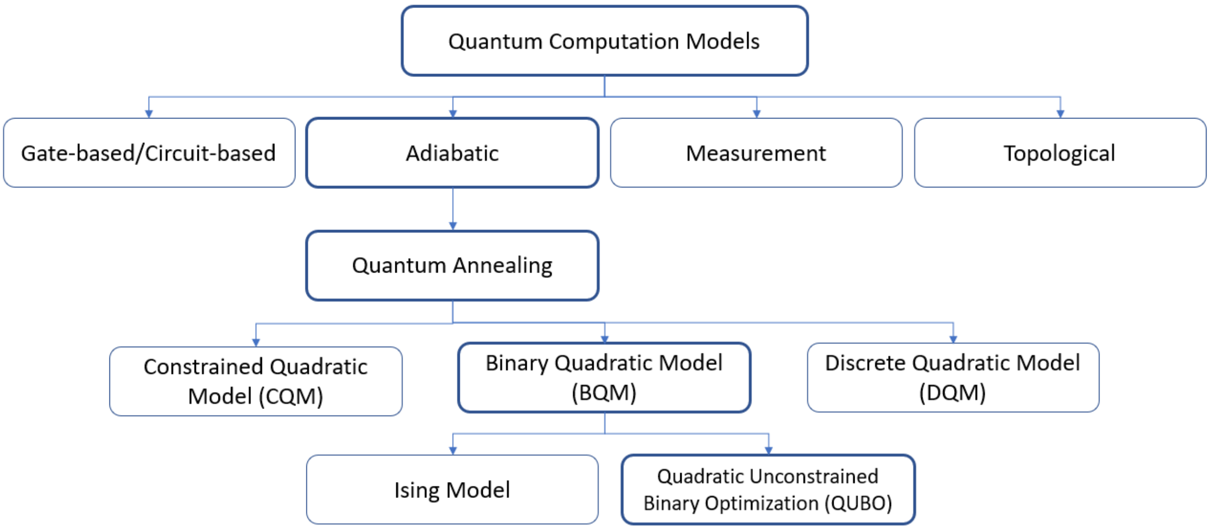


Figure 1. The overview of the scope of the paper. The Quadratic Unconstrained Binary Optimization (QUBO) is used.

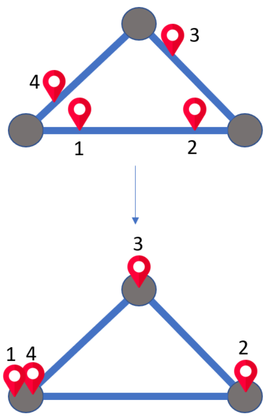


Figure 2. Mapping trajectory data to road network. Red icons represent trajectory data and grey nodes are the intersections in the road network.

where $Q_{i,i}$ = are the linear coefficients
$Q_{i,j}$ = are the quadratic coefficients

For each car, there are multiple alternative routes from which only few of them are considered as shortest paths. We selected three shortest alternative routes in order to identify the safest route. Each route contains multiple segments (edges) of the graph. To formulate the cost function for COVID-19 prevention, we define a summation of the number of COVID-19 cases assigned to the nodes of each edge (segment) and the shared segments between assigned routes for each car. By doing this, both historical data (COVID-19 statistics) and dynamic data (current traffic flow) play a significant role in distributing cars within the road network. We define a new function to minimize the risk of exposure (ROE) to the virus as follows (Eq 4).

$$ROE = \sum_{s \in S} (\sum_{v} \sum_{r} \sum_{s \in S_r} q_{vr}.(n_1 + n_2))^2 \qquad (4)$$

where s = segment
S = all the segments in road network
v = number of vehicles
r = number of alternative routes
$n1, n2$ = are normalized value of COVID-19 cases of start and end of a segment
q_{vr} = binary variable

The fact that QUBO problems are unconstrained does not mean that it is impossible to add constraints into the model; indeed, there are no constraints on the variables other than those defined in Q. The constraint of the model is defined based on the fact that each car cannot take more than one route at the time. To be more specific, given two cars and two alternative routes for each, all the possible states for qubits are shown in Table 1. In Table 1, the states are acceptable that the row's total values are equal one. In this regard, the final cost function have been revised as Eq 5.

	q1	q2	Status
Car 1	0	0	No
Car 1	0	1	Yes
Car 1	1	0	Yes
Car 1	1	1	No
Car 2	0	0	No
Car 2	0	1	Yes
Car 2	1	0	Yes
Car 2	1	1	No

Table 1. Possible states for two cars and two routes. Only records are acceptable with "Yes" value for Status column that means each car can only take one route at each time.

$$Costfunction = ROE + \sum_{v} (\sum_{r} q_{vr} - 1)^2 \qquad (5)$$

3.1 Pseudo Code

The pseudo code of the proposed method is provided in algorithm 1: Q construction Pseudo Code Table.

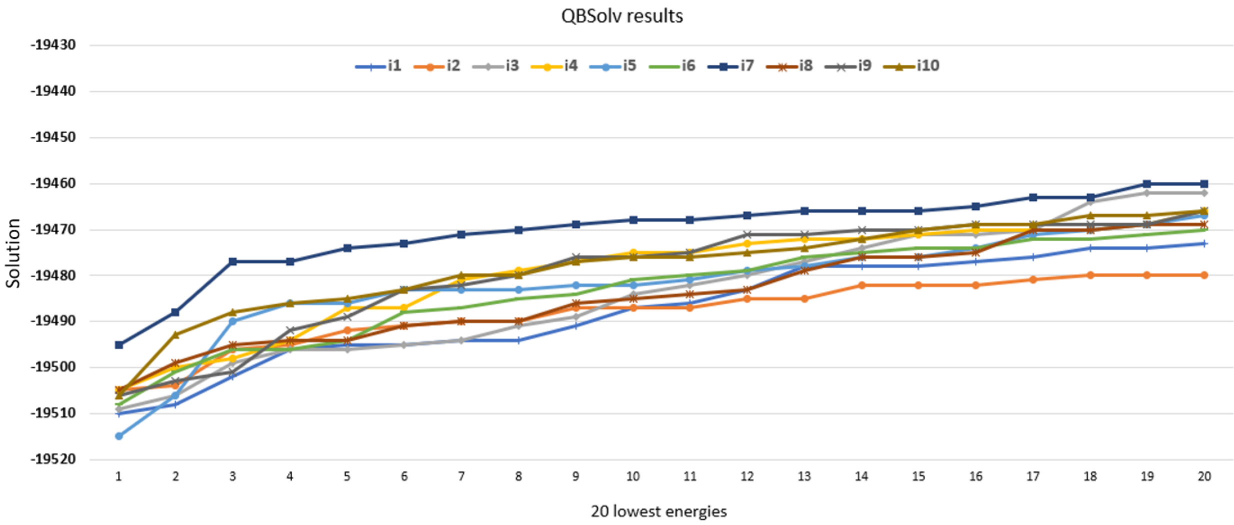


Figure 3. Results of the QBSolv solver for the lowest energies in 10 iterations. The horizontal axis shows the 20 lowest energies, and the vertical axis shows the level of energy—the lower the energy.

Algorithm 1 Q construction Pseudo Code

$G \leftarrow NetworkXGraph$
$Q \leftarrow [emptyMatrix]$
$dataset \leftarrow csvfile$
for i in cars:
- for j in routes:
- for k in segment:
- $node1 \leftarrow k.getNode(0)$
- $node2 \leftarrow k.getNode(1)$
- riskCovid19 $\leftarrow mapCovid19ToEdges(node1, node2)$
- weight $\leftarrow$ weight + riskCovid19
- G.addEdge(k,weight)
- weight(j).update
- Q(i,j).update
- $if(Q(i,j) > Q(i,j-1))\&j > 0:$
- maxValue.update(Q(i,j)
- Q(i,j).update $\leftarrow$ maxValue

3.2 Dataset

Microsoft Taxi data, called T-drive trajectory data, are used in this study to present the proof of concept. This dataset contains one-week trajectory data of 10,357 taxis in Beijing. This dataset contains about 15 millions records [1]. Due to the lack of a real trajectory dataset, the T-Drive dataset is used to simulate road users' movement. The COVID-19 dataset is acquired from the official website of the Beijing government [2].

4. IMPLEMENTATION

4.1 Road Network Construction

The first part of implementation is to construct the road network of the specific area. The road network is needed because it should be converted to a graph, and all other information should be mapped to the nodes and edges of the graph. In this paper, the road network is exported from Open Street Map (OSM). In order to get access to the data structure of the network, OSMnx[3] is used, which is an open-source python package. OSMnx helps to programmatically get access to the geospatial data structure of the OSM.

The next part is data cleansing as not all the imported information from OSM are required in this study. The derived road network has different attribute types for a single line (edge). For example, it can be drivable, walkable, etc. In this study, only walkable roads are used [4].

Once the data is converted as a shapefile format, it should be converted to GeoJSON format. All attributes and geospatial information such as spatial reference and geometry of features must be kept unchanged during the conversion. In this paper, one of the geoprocessing tools, feature to JSON conversion, is used to convert shapefile to GeoJSON [5]. The output of this step is the input of the next step. As in the next step, a geo-data-frame will be generated, GeoJSON format is needed.

The next step is to construct the geo-data-frame by using Pandas and GeoPandas. The GeoJSON format should be converted to the geo-data-frame because a road network graph can be generated from a geo-data-frame. Pandas is one of the robust data analysis and management frameworks in Python. Pandas stores data as a Data-Frame that is similar to tables with fields and records. GeoPandas, however, has more functionalities for geospatial analysis. GeoPandas has all the functionalities of Pandas as it is built on top of Pandas. GeoPandas contains many powerful spatial analysis functions, such as spatial aggregation methods and spatial joins that are useful to map trajectory data and COVID-19 data on the dataset.

Finally, the road graph has been constructed from the geo-data-frame by using NetworkX. The generated graph contains intersections of the road network as nodes and connections between

[1] https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/
[2] http://wjw.beijing.gov.cn/wjwh/ztzl/xxgzbd/gzbdyqtb/index.html
[3] https://osmnx.readthedocs.io/en/stable/
[4] https://github.com/gboeing/osmnx
[5] https://pro.arcgis.com/en/pro-app/latest/tool-reference/conversion/features-to-json.htm

intersections as edges. This graph is ready to get trajectory data and COVID-19 data mapped onto it.

4.2 Trajectory Data Reconstruction

After constructing the road network, next step is to reconstruct trajectory of of road users using origin and destination information. To do so, first, a geodatabase should be created from the Comma-Separated Values (CSV) file format origin-destination information. The CSV file is imported into ArcGIS, and a feature class is created for trajectory data. The generated feature class is then converted to the GoeJSON. Finally, the GeoJSON, for the same reason in the previous section, is converted to a geo-data-frame. Now, two geo-data-frame are created; one for road networks and the other one for trajectory data. The next step is to map the trajectory data on the geo-data-frame of the road network. The following section describes the process of mapping data.

4.3 Find Nearest Node To Each Trajectory Point

To map trajectory data on the road network data, we need to find the nearest node in the road network to one point of the trajectory data as presented in Fig 2. To do so, the Dijkstra algorithm has been used for shortest path project identifying alternative paths.

4.4 Construct Q matrix of QUBO problem

This is the essential part of the implementation, where the Q matrix should be defined based on the linear and quadratic coefficients of the QUBO problem. The Q matrix is upper triangular. Assume that the matrix is an empty matrix on the first step. The linear coefficients will be added to the diagonal of the matrix, and the quadratic coefficients will be added to the off-diagonal entries.

5. RESULTS AND DISCUSSION

The results of the paper show that Quantum Processing Unit (QPU) is faster than every other solver. QPU is only applicable for small problems, and this limitation is caused by QPU architectures. Mapping the logical layer (QUBO) into the physical layer (QPU) requires embedding on chimera or pegasus architecture. Before sending the problem to the QPU, the availability of such mapping should be checked. This process can be done by a minorminer[6]. It is essential to mention that if the mapping does not exist, it does not mean that the problem cannot be solved. The obstacle to mapping might be a couple of inactive qubits on the physical layer. It is possible to divide the problem into small batches, map them individually, and put them back together using classical computing. This approach is also known as hybrid quantum computing.

In terms of scalability, hybrid quantum computing is the fastest solution. Different hybrid solvers are available in D-wave quantum computers once the problem is formulated as a QUBO. Three quantum solvers have been investigated in this paper; QBSolv, D-wave hybrid, and Leap Hybrid solvers.

QBSolv is one of the decomposing solvers that find the minimum energy of a QUBO problem by breaking the problem into sub-problems. The sub-problems are solved on a classical computer by running a tabu algorithm. We first run the problem on a

[6] https://github.com/dwavesystems/minorminer

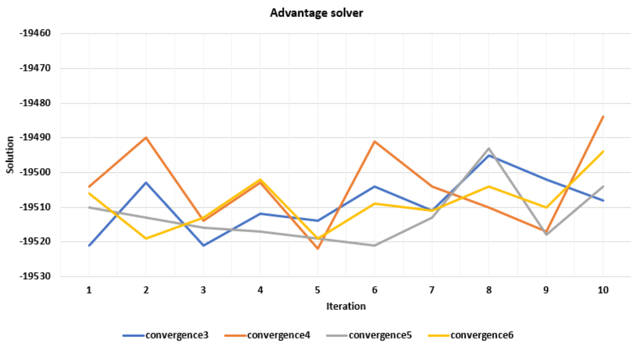


Figure 4. Results of running the proposed method on Hybrid-Advantage4.1 quantum solver.

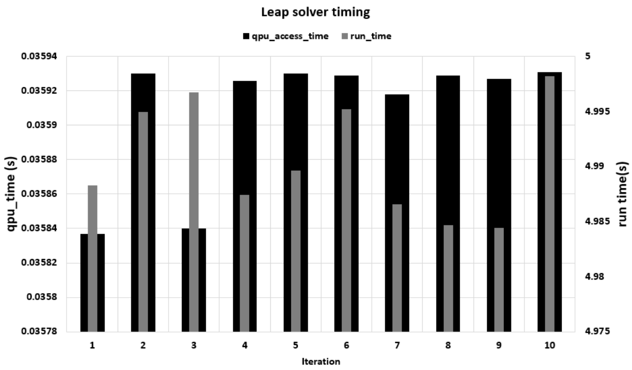


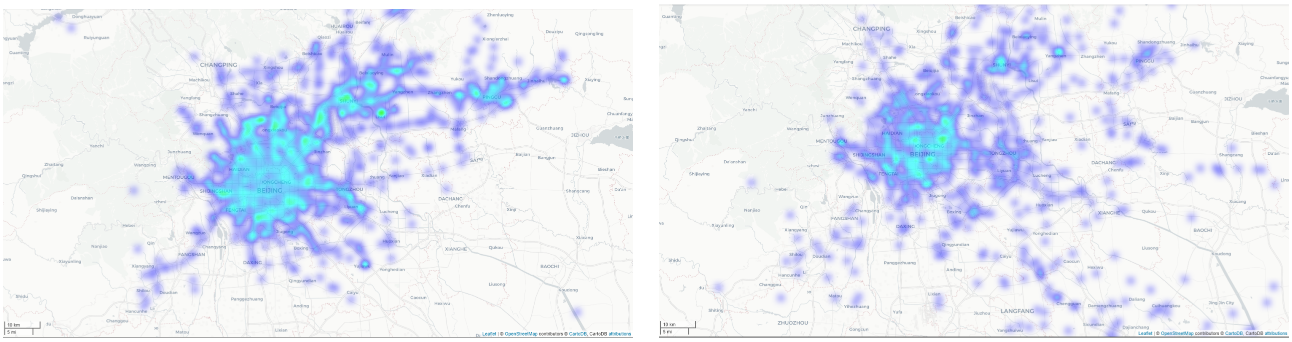
Figure 5. Run time, and QPU access time for ten iterations on Leap hybrid solver. The left vertical axis shows the run time values and the right vertical axis shows the QPU access time.

QBSolv because it helps us define the QUBO problem on a classical computer and debug it, and once everything works well, it will be passed to the quantum solvers. Figure 3 represents the results of the QBSolv solver for the lowest energies in 10 iterations. The horizontal axis shows the 20 lowest energies, and the vertical axis shows the level of energy—the lower the energy, the better answer. As it is shown in Figure 3, iteration 5 found the best answer, and iteration seven found the worst answer.

Figure4 depicts the result of running the proposed method on the Advantage-system-4.1 quantum computer for four convergence values in 10 iterations. Convergence in this solver means the number of iterations with no improvement that terminates sampling. The vertical axis shows the energy level of the results, and the horizontal axis shows the iteration number. The results show that convergence equals to 3 found the best answers on iterations one and two. Advantage-system-4.1 solver has 5760 qubits, from which 5619 qubits were active at the time of our run.

Figure5 shows the result of running the problem on the Leap Hybrid quantum solver for ten iterations. The left vertical axis represents qpu-time, which is the time it takes to run the problem on QPU, and the right vertical axis shows run-time, which is the time it takes to break the problem into sub-problems the classical computer. As it is shown, qpu-time is always less than run-time.

the key lesson learned from the results are as follows. Since the hybrid solvers give the sub-optimal solution, the final solution is not the same in each iteration. However, the comparison



(a) Current state of trajectory data. Heatmap without using the proposed model.

(b) Heatmap of the cars trajectory by using the proposed model on a hybrid solver.

Figure 6. A comparison between two states. The expected results for whole datasets will be a heatmap with fewer hotspots for each time unit of the network. Figure 6a is used as ground truth to evaluate results.

between different solvers helps to better understand the results. The other important note is that the number of reads should be different for each problem, and it can be defined by try and error. This parameter forces QPU to run a problem based on its value. As the time scale for each read is microseconds, it is possible to increase the number of reads. Moreover, as the results of the QPU are probabilistic, more reads lead to more diverse solutions. The comparison between different solvers reveals that the solvers with Pegasus architecture are better than Chimera because of the topology and connections of the qubits. It is expected that once QPU can map more complex graphs, they will perform incredibly better than GPUs in these types of combinatorial problems.

In this research, two primary states were defined to evaluate results. The first state represents ROE by integrating the actual data layers; the second state, however, calculates ROE based on the proposed model. A comparison between two states is shown in Figure 6 for a subset of the datasets. A visualization of the datasets is available in Github [7].

6. CONCLUSION

In this study, we sought to bring QC into the geospatial problems to take advantage of this meta-heuristic approach and open the door for future works. To do so, we briefly introduce QC's fundamentals, followed by adiabatic quantum computing and QA. The Covid-19 prevention problem has been converted to the QUBO. The results of the paper show that Quantum Processing Unit (QPU) is faster than every other solver. However, the QPU is only applicable for small problems, and this limitation is caused by QPU architectures. Therefor, hybrid quantum solvers have been investigated such as D-wave hybrid and Leap Hybrid solvers. The heat maps indicate the better distribution of the vehicles on road networks after applying the proposed model. The result shows a promising avenue for such complex problem solving using a quantum computer. Further research will expand on broader transportation systems leading to more complex optimization with additional constraints.

REFERENCES

Albash, T., Lidar, D. A., 2018. Adiabatic quantum computation. *Reviews of Modern Physics*, 90(1), 015002.

Briegel, H. J., Browne, D. E., Dür, W., Raussendorf, R., Van den Nest, M., 2009. Measurement-based quantum computation. *Nature Physics*, 5(1), 19–26.

Cartenì, A., Di Francesco, L., Martino, M., 2021. The role of transport accessibility within the spread of the Coronavirus pandemic in Italy. *Safety science*, 133, 104999.

Chen, D., Pan, S., Chen, Q., Liu, J., 2020. Vehicle routing problem of contactless joint distribution service during COVID-19 pandemic. *Transportation Research Interdisciplinary Perspectives*, 8, 100233. https://www.sciencedirect.com/science/article/pii/S2590198220301445.

Gibney, E., 2019. Hello quantum world Google publishes landmark quantum supremacy claim. *Nature*, 574(7779), 461–463.

Hogg, T., 2003. Adiabatic quantum computing for random satisfiability problems. *Physical Review A*, 67(2), 022314.

Kato, T., 1950. On the adiabatic theorem of quantum mechanics. *Journal of the Physical Society of Japan*, 5(6), 435–439.

Kochenberger, G., Hao, J.-K., Glover, F., Lewis, M., Lü, Z., Wang, H., Wang, Y., 2014. The unconstrained binary quadratic programming problem: a survey. *Journal of combinatorial optimization*, 28(1), 58–81.

Lenstra, J. K., Kan, A. R., 1981. Complexity of vehicle routing and scheduling problems. *Networks*, 11(2), 221–227.

Michielsen, K., Nocon, M., Willsch, D., Jin, F., Lippert, T., De Raedt, H., 2017. Benchmarking gate-based quantum computers. *Computer Physics Communications*, 220, 44–55.

Nagaj, D., 2012. Universal two-body-Hamiltonian quantum computing. *Physical Review A*, 85(3), 032330.

Neukart, F., Compostella, G., Seidel, C., Von Dollen, D., Yarkoni, S., Parney, B., 2017. Traffic flow optimization using a quantum annealer. *Frontiers in ICT*, 4, 29.

Roy, A., DiVincenzo, D. P., 2017. Topological quantum computing. *arXiv preprint arXiv:1701.05052*.

Scokaert, P. O., Rawlings, J. B., 1998. Constrained linear quadratic regulation. *IEEE Transactions on automatic control*, 43(8), 1163–1169.

[7] https://amirhossein-nourbakhsh.github.io/Quantum-geo-AI/index.html

Shi, P., Dong, Y., Yan, H., Zhao, C., Li, X., Liu, W., He, M., Tang, S., Xi, S., 2020. Impact of temperature on the dynamics of the COVID-19 outbreak in China. *Science of The Total Environment*, 728, 138890. https://www.sciencedirect.com/science/article/pii/S0048969720324074.

Somma, R. D., Nagaj, D., Kieferová, M., 2012. Quantum speedup by quantum annealing. *Physical review letters*, 109(5), 050501.

Tosepu, R., Gunawan, J., Effendy, D. S., Ahmad, L. O. A. I., Lestari, H., Bahar, H., Asfian, P., 2020. Correlation between weather and Covid-19 pandemic in Jakarta, Indonesia. *Science of The Total Environment*, 725, 138436. https://www.sciencedirect.com/science/article/pii/S0048969720319495.

WHO, 2020. Coronavirus disease 2019 (COVID-19): situation report, 72. *World Health Organization*. https://apps.who.int/iris/handle/10665/331685.

WHO, 2021. WHO Coronavirus (COVID-19). *World Health Organization*. https://covid19.who.int/info/.

Xie, J., Zhu, Y., 2020. Association between ambient temperature and COVID-19 infection in 122 cities from China. *Science of The Total Environment*, 724, 138201. https://www.sciencedirect.com/science/article/pii/S0048969720317149.