Passenger Re-accommodation based on Airlines Schedule Change

Team-59

December 16, 2023

Abstract

In the aviation industry, flight cancellations are inevitable, causing disruptions and inconveniences for passengers. This report addresses the challenge of efficiently searching for feasible flight alternatives for impacted passengers and subsequently reallocating them to minimize the overall disruption. Traditional methods often fall short in handling the complex optimization constraints involved in this process. To overcome these limitations, we propose using quantum annealing to explore vast solution spaces and optimize under stringent constraints. Quantum computing, with its inherent parallelism and superposition capabilities, offers the potential to explore multiple solutions simultaneously and efficiently navigate the complex constraints. We delve into the application of quantum annealing to formulate and solve the flight search and passenger reallocation problem.

1 Introduction

Quantum computing is a rapidly-emerging technology that harnesses the laws of quantum mechanics to solve problems too complex for classical computers. An explosion of benefits from quantum research and development is taking shape on the horizon. As quantum hardware scales and quantum algorithms advance, many big, important problems like combinatorial optimization should find better and faster solutions. To utilize the existing resources in an efficient manner, specific algorithms have been proposed for various disciplines, e.g., quantum chemistry, combinatorial optimization, and machine learning. A promising strategy to cope with the existent resource limitations is represented by hybrid quantum-classical approaches, which comprise quantum as well as classical computation.

The normal operation of aircraft and flights can be affected by various unpredictable factors, such as seasonal demands, severe weather, airport closure, and corrective maintenance, leading to disruption of the planned schedule. When a disruption occurs, the airline operation control center performs various operations to reassign resources (e.g., flights, aircraft, and crews) and redistribute passengers to restore the schedule while minimizing costs and taking into account that maximum number of passengers get re-accommodated to the alternate flights [SXW+21].

To identify optimal alternate flight solutions for all the impacted passengers (impacted due to planned schedule change) based on the provided rule sets, this problem falls into the set of constraint optimization problems. Generally these types of problems have a very high time complexity when solved through available classical algorithms.

Quantum algorithms take a new approach to these sorts of complex problems — creating multidimensional computational spaces. This turns out to be a much more efficient way of solving complex problems like combinatorial optimization. Applying QC in optimization problems has been one of the main fields of study in recent years especially using the Quadratic Unconstrained Binary Optimization (QUBO) model, which companies such as IBM, D-wave and Microsoft have built quantum computers designed for solving this type of problems. In our approach we aim to solve this problem using Binary Quadratic Model (BQM) and Constraint Quadratic Model (CQM).

2 Literature Review

Petersen et al. [PSC+12] defined the aviation recovery problem into four modules, which were divided into flight schedule recovery, crew recovery, passenger flow recovery and aircraft recovery, etc. By using column generation and Benders branch-and-bound algorithm, the overall optimal solution was obtained through iterative calculation of problem decomposition and optimization. J Sun et al. proposes [SL23] that by upgrading the passenger recovery engine, the purpose is to provide the optimal recovery scheme for passengers, so as to reduce the risk of transferring overseas flights, and thus reduce the economic loss of airlines. In this paper, the optimization model and algorithm based on network flow, combined with actual business requirements, comprehensively consider multiple optimization objectives to quickly generate passenger recovery solutions, and at the same time achieve the optimal income of airlines and the acceptance rate of passenger recovery, so as to balance the two. The practicability and effectiveness of the proposed model and algorithm are proved by some concrete examples.

3 Problem Formulation

To solve this problem we have divided it into 2 sub problems with first one being finding the feasible paths to which passengers can be re-accommodated and the other one being reallocation of impacted passengers to these paths. For both the problems required objectives are being formulated taking into account flight schedules, PNR ranking and constraints.

3.1 Path Searching

Our aim is to minimize departure time delay and arrival time delay while preserving our constraints.

3.1.1 Objective Function

$$min \sum_{i}^{S_1} \sum_{k}^{A_1} \sum_{l}^{D_T} |T_{Sikl} - \alpha| q_{Sikl} \tag{1}$$

$$min \sum_{e}^{E} \sum_{j}^{S_2} \sum_{k}^{A_2} \sum_{l}^{D_T} |T'_{jekl} - \beta| q_{jekl} \tag{2}$$

S: Starting Airport
E: Set of all Ending Airports (including city pairs)
q_{Sikl}: Binary decision variable for selection of flight going from starting airport S to airport i with aircraft ID k with departure time index l
T_{Sikl}: Departure time of departure time index l of flight k going to airport i from S
T'_{jekl}: Arrival time of arrival time index l of flight k going to airport e from airport j

α: Departure time of cancelled flight
β: Arrival time of cancelled flight
S_1: Set of all airports visited by flights going from starting airport
S_2: Set of all airports visited by flights going from starting airport
A_1: Set of all Aircraft IDs of flights leaving S_1
A_2: Set of all Aircraft IDs of flights leaving S_2
D_T: Departure time indexes of the flights

3.1.2 Constraints

$$\sum_{i}^{S_1} \sum_{k}^{A_1} \sum_{l}^{D_T} q_{Sikl} = 1 \tag{3}$$

$$\sum_{e}^{E} \sum_{j}^{S_2} \sum_{k}^{A_2} \sum_{l}^{D_T} q_{jekl} = 1 \tag{4}$$

$$\sum_{o} \sum_{k} \sum_{l} q_{ojkl} - \sum_{j} \sum_{k} \sum_{l} q_{jqrs} = 0 \quad \forall j \notin S, E \tag{5}$$

Constraint 3 ensures first flight of every path departs from starting airport S while constraint 4 ensures that last flight of every path is the destination airport E.

Constraint 5 ensures that on an intermediate node if a flight arrives then a flight departs from that node as well.

$$\sum_{k'} \sum_{m'} \sum_{l'} T_{jk'l'm'} q_{jk'l'm'} - \sum_{i} \sum_{k} \sum_{l} q_{ijkl} \left(T'_{ijkl} + \lambda_1 \sum_{k'} \sum_{m'} \sum_{l'} q_{jk'l'm'} \right) \geq 0 \quad \forall j \notin S, E \tag{6}$$

$$\sum_{k'} \sum_{m'} \sum_{l'} T_{jk'l'm'} q_{jk'l'm'} - \sum_{i} \sum_{k} \sum_{l} q_{ijkl} \left(T'_{ijkl} + \lambda_2 \sum_{k'} \sum_{m'} \sum_{l'} q_{jk'l'm'} \right) \leq 0 \quad \forall j \notin S, E \tag{7}$$

λ_1: Minimum time between connecting flights
λ_2: Maximum time between connecting flights

Last two constraints ensure that connection time between 2 flights is greater than a certain value and smaller than maximum time to stay on an airport.

3.2 Passenger Reallocation

Our aim is to maximize total PNR scoring while preserving our constraints.

3.2.1 Objective Function

$$max \sum_{c \in S_p} \sum_{i \in c} \alpha_i \sum_{j}^{N} \sum_{k}^{K} p_{jk} x_{cijk} \tag{8}$$

α_i: Value of flight based on path score
p_{jk}: PNR score if PNR j is allocated to class k
x_{ijk}: Binary decision variable for allocation of PNR j in class k of flight i
S_p: Set of all paths
N: Total Number of Impacted PNR
K: Total Number of Classes

3.2.2 Constraints

$$\sum_{j}^{N} \sum_{c \in S_f} x_{cijk}.w_j \leq C_{ik} \quad \forall k \in K, \ \ \forall i \in c \tag{9}$$

$$\sum_{i \in S_1} \sum_{k}^{K} x_{cijk} \leq 1 \quad \forall j \in (1, 2, ..., N), \ \ \forall c \in S_p \tag{10}$$

$$\sum_{i \in S_2} \sum_{k}^{K} x_{cijk} \leq 1 \quad \forall j \in (1, 2, ..., N), \ \ \forall c \in S_p \tag{11}$$

w_j: Number of passengers in PNR j
S_f: Set of all flights in all feasible paths
S_1: Set of all flights coming from the starting airport in the feasible paths
S_2: Set of all flights going to the destination airport in the feasible paths

Constraint in equation 9 is the capacity constraint for the class k of flight i, while constraints in 10 and 11 says that a passenger at best can be accommodated to one of the flights leaving from the departure airport and flights reaching the destination airport respectively for any feasible.

$$\sum_{c \in S_p} \left(\sum_{i \in F_I} \sum_{k}^{K} x_{cijk} - \sum_{i \in F_O} \sum_{k}^{K} x_{cijk} \right) = 0 \quad \forall j \in 1, 2, ..., N \tag{12}$$

F_I: Set of incoming flights at airport c
F_O: Set of outgoing flights from airport c

$$\sum_{c \in S_p} \left(\sum_{i \in F_c} \sum_{k}^{K} x_{cijk} - N_c. \sum_{k}^{K} x_{cfjk} \right) = 0 \quad \forall j \in 1, 2, ..., N \tag{13}$$

N_c: Number of flights in path c

F_c: Flights of path c
f: First flight of path c

Constraints 12 is about if a PNR reaches an intermediate airport then it should leave it as well and 13 deal with preserving the path of the PNR on which it travels.

4 Methodology

We propose a two-stage algorithm to solve this problem. In the first stage we deploy quantum annealing using BQM and CQM to model the objective function for finding out the feasible paths according to the relevant constraints. For the passenger reallocation problem we have mapped it to a version of knapsack problem which is the class constrained multi-knapsack problem (CCMKP) [A+23]and relevant constraints have been added according to the given rule set.

These two problems are being modelled using Linear Programming (LP) [ABES21] and Quadratic Programming (QP). The problem is formulated using these as shown above and we have used both classical and Quantum-Annealing based approach to solve the problems [KM20] and in later we compare their time complexities as well.

4.1 Data Preprocessing

We have applied four steps in data preprocessing stage. First of all flights in next 72 hours from the departure time of cancelled flight, are identified. Then in the next three subsequent steps paths with direct flight, one connecting flight and two connecting flights from source to destination airport are being extracted.

4.2 Rules Used

We are neglecting the paths from source to destination with more than two stopovers. We have used the following scores for flight ranking and PNR ranking:

Rule	Score
Arrival<6hrs	70
Arrival<12hrs	50
Arrival<24hrs	40
Arrival<48hrs	30
Equipment	50
Same_Citypairs	40
Different_Same_Citypairs	30
Different_Citypairs	20
SPF<6hrs	70
SPF<12hrs	50
SPF<24hrs	40
SPF<48hrs	30
Stopover	-20

Table1: Flight Scoring

Rule	Score
SSR	200
Per_PAX	50
First Class	2000
Business Class	1850
PC Class	1650
Economy Class	1500
Prem. Platinum	2000
Platinum	1800
Gold	1600
Silver	1500

Table2: PNR Scoring

The business rules for PNR scoring can simply be toggled on and off in the code in any of the rules mentioned in the above table and also there is separate allowance for either allowing upgrade or downgrade of cabin or both. For the case of city pairs we have put the default value as 150KM but this is also subject to change according the airline preferences.

4.3 Classical Approach

For the classical approach, the system strategically leverages the efficient algorithms of the Pandas library for data handling. The exploration of alternative flight routes is facilitated through a recursive search mechanism. Despite the theoretical concern of poor time complexity associated with recursion, real-world constraints dictate that stopovers rarely exceed three flights. This limitation ensures a low recursion depth, contributing to the code's remarkable speed. Interestingly, this classical approach proves more efficient than leveraging quantum algorithms. The allocation of passengers is accomplished through a greedy approach, ensuring high-priority passengers are assigned to the best available flights. The integration of filters further refines the reallocation process, imposing constraints on the number of stopovers and ensuring a minimum time gap between two flights during a stopover.

4.4 Quantum Annealer-Based Approach

Quantum annealers, at their core, function as ISING machines designed to address combinatorial optimization challenges. The process of utilizing quantum annealers for solving optimization problems entails the conversion of real-world problems from various industries and academic fields into energy minimization problems.

Quantum annealers implement energy encoding to translate these problems into a format compatible with their hardware, and they adhere to a quantum optimization approach inspired by natural processes. This approach enables the quantum system to progress over time, all while maintaining the ability to control the pace of this evolution. Given sufficient time, the system will ultimately reach its lowest energy state.

We have used D-Wave's quantum annealers [Dwa] which take constraints and objectives in the form of BQM and CQM [Cod21]. Dimod has been utilized for BQM and CQM formulation. After this D-Wave's leap hybrid solvers are being utilized which employs advanced classical algorithms with strategic allocation of the quantum computer's capabilities to the portions of the problem where they offer the most significant benefits, to minimize the energy of the Ising Hamiltonian made from CQM and BQM to finally solve the problem. Dwave's leaphybridCQMSolver offers a choice to provide multiple solutions ranked by their energies, all satisfying the provided constraints. This will provide us with all possible paths in one go, to be further used for reallocation of passenger.

4.5 Reason for using Dwave Quantum Computers

Initially we explored the quantum walk algorithm which are quantum counterparts of Markov chains [Amb03]. We tried to use it in path searching part of our problem but due to limited qubits and noise in them did not offered reliable solutions. We also tried to use quantum genetic algorithm [HK00] for passenger reallocation but due to less number of qubits we could not simulate it for problems for big size. Qiskit utilised a gate based approach for quantum computation, by transforming qubit states to find an optimum solution. A major drawback in this is the computation power offered by Qiskit. Maximum number of qubits offered by qiskit is very less to construct the entirety of a computationally expensive task into a scenario suitable for quantum computation.

Moreover Qiskit only offers a QUBO, as a combinatorial optimization model, which consumes a lot of time in constructing the model. Unconstrained optimization involves conversion of our constraints into penalties to be directly added to the objective function.

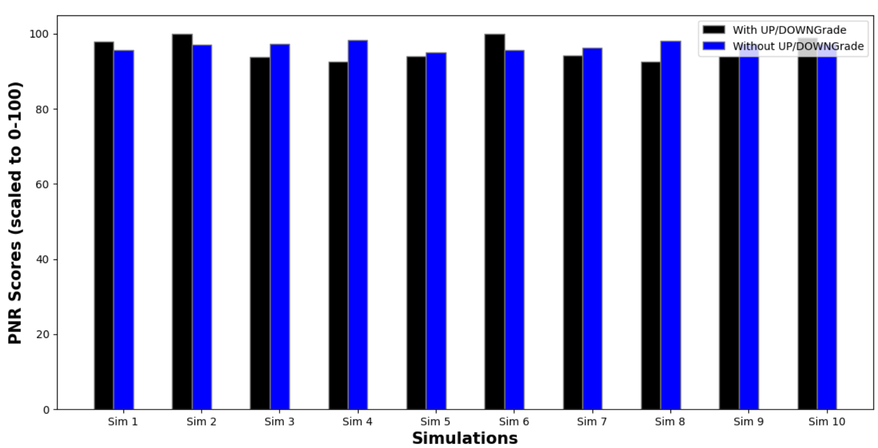
This brings us to our choice of optimization model, Constrained Quadratic Model (CQM). Dwave's libraries offers a wide range of inbuilt functions to properly model a task into CQM. Major benefit of CQM is that it does not require creation of penalties from constraints in the objective function, but penalises a result after an anneal step, if it violates any constraints. This saves a lot of time which was consumed in creation of penalties in the QUBO model.

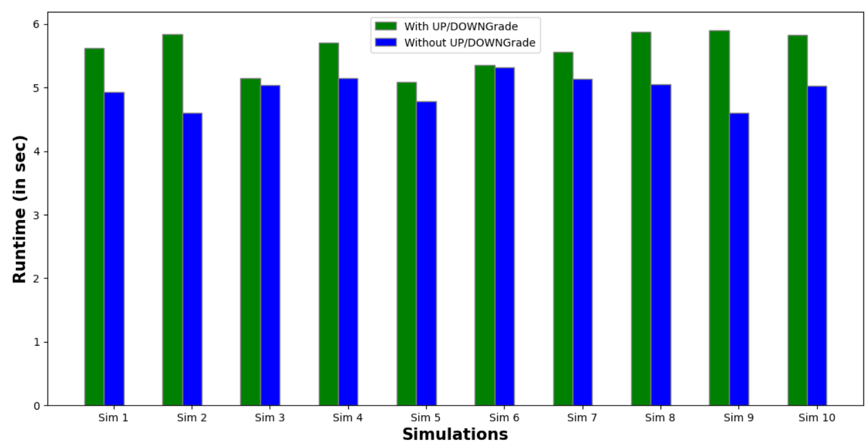
Furthermore, the feasible solutions provided by CQM, follow all constraints given, which is not a guarantee in QUBO. Moreover, by a hybrid of Quantum and classical solver, the maximum qubits offered by Dwave is much higher than that of Qiskit, and proves sufficient to solve our problem.

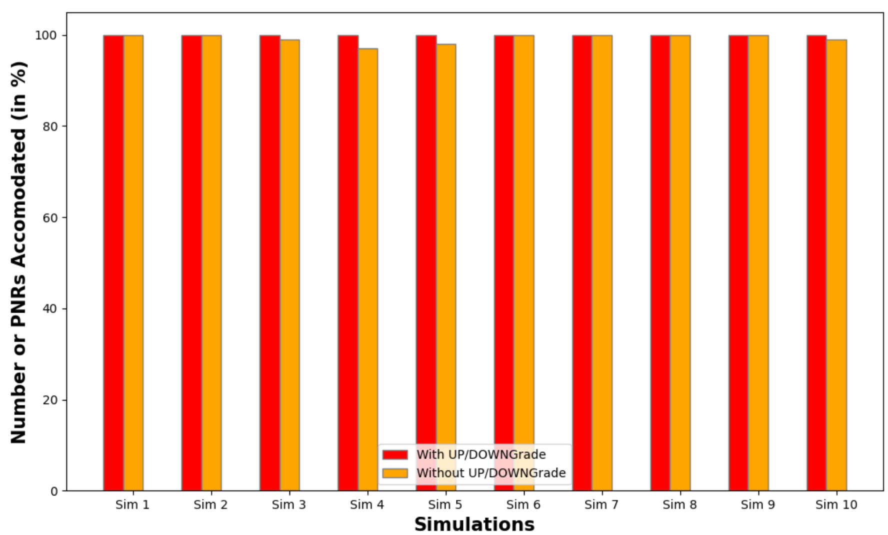
5 Results

We have used two approaches to solve this problem and analyzed their solutions through multiple simulations. The three approaches we have used are, 1.)Classical algorithm for path finding and quantum algorithm for passenger reallocation. 2.)Quantum Algorithms for both path finding as well as passenger reallocation. We did simulations for a single disrupted flight by changing utilizing different set of business rules in each simulation and ranked solutions on the basis of PNR scoring and number of passengers reaccommodated. For the first approach the runtime was more averaging around 5 seconds for each solution on the quantum annealer and the solutions obtained were optimal while for the second approach we obtained similar results but sometimes they were inaccurate compared to what we got for the first approach but the time complexity was much increased in this case averaging around 10 seconds.

One of our methods of simulation for testing our first approach was run it 10 times by randomly generating a single flight disruption in our data. This simulation is done for two business rules being upgrade and downgrade not allowed and they both being allowed. We compare the solutions based on runtime, number of passengers accommodated and total PNR score of allocated passengers.







In the results we observe that we have more PNR scoring and number of passengers reallocated when upgrade and downgrade of business rules are allowed but the time complexity of simulation in this case increases by a few milliseconds in comparison to when cabin upgrade and downgrade not allowed but with a trade of a few passengers not accommodated at times.

6 Conclusion

While quantum computing holds theoretical promise, its practical implementation is limited by the expansive search space of the possible flights. Consequently, the classical approach, complemented by Pandas, remains the more pragmatic and efficient choice given the current state of quantum computing technology. This is however not true when we talk about the allocation of passengers to the flights found. The greedy method, while fast, does not give the best solution in most cases. The Quantum approach turns out to be better in this situation. We also observed that for one out of twenty solutions while using quantum annealing are not feasible which might be due to noise in the qubits.

7 Future Scope

With the onset of better quantum computers we may be able to utilize other algorithms which are based on gates to solve combinatorial optimization problems such as these with a better time complexity. As the quality of qubits get better and they get less noisy quantum solutions will get more reliable and efficient.

References

[A+23] A Awasthi et al. Quantum computing techniques for multi-knapsack problems, 10.48550. *arXiv preprint ARXIV.2301.05750*, 2301, 2023.

[ABES21] M Alpuente, D Ballis, S Escobar, and J Sapiña. Journal of logical and algebraic methods in programming. 2021.

[Amb03] Andris Ambainis. Quantum walks and their algorithmic applications. *International Journal of Quantum Information*, 1(04):507–518, 2003.

[Cod21] Philippe Codognet. Constraint solving by quantum annealing. In *50th International Conference on Parallel Processing Workshop*, pages 1–10, 2021.

[Dwa] D-wave quantum computing. d-wave systems inc. [online]. available: https://www.dwavesys.com/.

[HK00] Kuk-Hyun Han and Jong-Hwan Kim. Genetic quantum algorithm and its application to combinatorial optimization problem. In *Proceedings of the 2000 congress on evolutionary computation. CEC00 (Cat. No. 00TH8512)*, volume 2, pages 1354–1360. IEEE, 2000.

[KM20] Thomas Krauss and Joey McCollum. Solving the network shortest path problem on a quantum annealer. *IEEE Transactions on Quantum Engineering*, 1:1–12, 2020.

[PSC+12] Jon D Petersen, Gustaf Sölveling, John-Paul Clarke, Ellis L Johnson, and Sergey Shebalov. An optimization approach to airline integrated recovery. *Transportation Science*, 46(4):482–500, 2012.

[SL23] Jiamin Sun and Haiming Li. Abnormal flight passenger recovery algorithm based on itinerary acceptance. *Journal of Computer and Communications*, 11(11):167–182, 2023.

[SXW+21] Yi Su, Kexin Xie, Hongjian Wang, Zhe Liang, Wanpracha Art Chaovalitwongse, and Panos M Pardalos. Airline disruption management: A review of models and solution methods. *Engineering*, 7(4):435–447, 2021.