A Large Neighbourhood Search Approach to Airline Schedule Disruption Recovery Problem

Kam K. H. NG [1], K. L. KEUNG [2], C. K. M. LEE [2], Y.T. CHOW [3]

[1] Interdisciplinary Division of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China

[2]Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China

[3] Division of Business and Hospitality Management, College of Professional and Continuing Education, The Hong Kong Polytechnic University, Hong Kong SAR, China

(kamhung.ng@ntu.edu.sg, dicky-kin-lok.keung@connect.polyu.hk, ckm.lee@polyu.edu.hk, hester.chow@cpce-polyu.edu.hk)

Abstract - **The occurrence of unplanned aircraft shortages and disruption of flight schedules during the day-to-day operations of airlines is inevitable. When equipment failure causes unsafe flight, the aircraft will be grounded or temporarily delayed when the weather shuts down the airport or the required flight crew is unavailable. Real-time decisions must be made to reduce revenue loss, passenger inconvenience and operating costs by reallocating available aircraft and cancelling or delaying flights. A large neighbourhood search algorithm is used in this research to construct a feasible and efficient solution to the airline schedule disruption recovery problem. We aim to reduce the aircraft turn-around times, including total delay time, the number of flight adjustments and the number of flights delayed for more than one hour, as an objective function. Ten real-life cases are solved, and the proposed approach yields an approximate 50% improvement in solution quality.**

Keywords – **Airline recovery, passenger itineraries, large neighbourhood search, fleet assignment**

I. INTRODUCTION

The airline schedules can be disrupted by several factors, because as the air routes are interlinked and interconnected network. Airline schedule disruptions are expected to be increasingly frequent in the near future [1-3]. It is important to develop a disruption recovery approach via optimisation to resolve the impact of delay aggregation [4-9]. More importantly, adjustment plans have to be made more dynamically and efficiently to avoid the extra costs caused by airline scheduling [10-13].

Airlines usually develop desirable flight schedules in advance and fully utilise their fleets [1]. A flight schedule consists of information such as the originating city, departure time, destination, and arrival time. Flight schedules are usually created in advance based on the seasonality of trends and other factors. However, specific plans for the assignment of aircraft to flights are based on a much shorter time span [5, 14]. These plans minimise the idle time of aircraft fleets to maximise utilisation to cover the high purchase and maintenance costs of aircraft [15, 16]. Flight schedule is disrupted when an aircraft cannot go on a scheduled flight due to various reasons, such as extreme weather or emergency maintenance. [13, 17, 18].

To recover from airline schedule disruptions, one of the solutions is to swap aircraft to execute the flights. However, this entails the flight crews to be adjusted. Also, the swapped aircraft may require a change in the maintenance schedule for flying the additional flights. Another possible adjustment in this category is the reassignment of aircraft. The airline can choose to use a spare aircraft or ferry another aircraft from another airport in place of the aircraft that is being replaced. [19]. While using spare aircraft can reduce the ripple effect caused by a single disruptive incident, the airlines cannot afford the high cost of keeping spare aircraft. Using ferrying aircraft also runs to high cost for the airlines. In unforeseen circumstances, the terminal manoeuvring area of an airport may be affected by inclement weather, preventing the aircraft from taking off or landing at the affected airport. This resultant cancellation of a large number of flights can affect the entire scheduling, leading to missed connections and other disruptions at the upstream and downstream airports. Therefore, the airline must be able to modify its route layout to minimise the impact of such unforeseen events as bad weather [20]. At the end of the unforeseen event and the airlines must be able to re-establish the normal schedule without delay [21].

This paper aims to develop solutions, namely, schedule and rotation modifications to resume normal operations as quickly as possible after the disrupting event. In real-life situations, modifications to the entire schedule are very costly. The costs will continue to rise as long as the effects of schedule disruption persist. Therefore, a quick meta-heuristic algorithm that aims at modifying rotations and schedules within short a recovery period and with a reasonable decision-making time is required [22]. The aim is to resume the pre-planned schedule soon after the recovery period, thus minimising the incurred costs for the airlines. Furthermore, the objective of this paper is also to minimise the costs and other potential impacts on passengers during the recovery period.

II. PROBLEM FORMULATION

The joint passenger and aircraft recovery problem can be developed as a time-space network $G = (N, A)$. Each node is in the set $N = \{1, \dots, n\}$. A node represents an airport at a specific time. Each arc is in the set $A = \{(i, j),$ $i, j,\ i \neq j \in N$. Arc represents a connection between two flight legs at the same airport. p represents the maximum number of arrivals and departures allowed during each 60-minute period. $I_i(p)$. $O_i(p)$ represents all acrs (i,j) departing from node i during period p and all acrs (i,j) arriving at node j during period p. $a_{ip.}$ and b_{ip} represents the arrival capacity at airport i in period p and the departure capacity at airport i in period p. Aircraft, $f \in F = F_c \cup F_d \in F,\ f = \{1, \ldots n\}$, where flights can be classified as normal, delayed and cancelled [23, 24].

Operation costs are represented by $c_{ijf.}$ The algorithm developed in this paper calculates operating costs according to a function of the aircraft model and flight time. These costs are independent of the number of passengers on board and of the possible delay. Furthermore, if a planned flight is cancelled, the operating costs of the flight will be taken out of consideration. If the trip is one-way, the total travel duration for the passenger only consists of a single flight. If the trip involves of transition between different flights, the duration of the trip is calculated as the sum of the durations of these flights excluding transit time. The calculation of the cost of delay for the delayed passengers is based on the level of effect mentioned previously is denoted by r_{ij}. The cost of cancelled trips for passenger p, which includes the full ticket price refund and financial compensation, which is related to the duration of the trip is denoted by e_{ij}. The perceived delay cost that depends on distance type indicator, delay duration, and the property of the trip is denoted by h_{ij}. The perceived cancellation cost that depends on distance type indicator, delay duration, and property of the trip is denoted by f_{ij}. The downgrading cost g_{ijf} includes the costs of each leg of flight in the passenger's itinerary. For every flight, downgrading cost is calculated based on distance type indicator and the downgrading degree, which is decided by the cabin class difference. L^f_{fam}, L^f_{mod} and L^f_{conf} represents the return to normal costs incurred due to the position penalty of aircraft, regarding the degrees of differences respectively. The decision variable x_{ijf} is equal to 1 if, and only if, aircraft f uses arc (i,j). The notation and decision variables are shown in Table I.

The objective function from Equation (1) of the algorithm includes additional costs or benefits-related parameters that result from modifying the flight plan (modified flight operating costs, deducting operating costs for cancelled flights, delay and cancellation costs, as well as measuring the negative impact on passengers and their perceived loss. A return to normal cost is also enforced to simulate a realistic environment. The goal is to minimize the sum of all the costs calculated for these three categories. Constraints (2) and (3) are the flow conservation constraints. Constraints (4) and (5) are the airport capacity constraints. Constraint (6) is the assignment constraint, which is that only one aircraft will be assigned to each flight leg.

TABLE I
NOTATIONS AND DECISION VARIABLES

Notations	Explanation
$G = (N, A)$	Time Space Network, N = Node; A = Arc
N	Node $N = \{1, \ldots, n\}$
A	Arc $A = \{(i,j),\ i,j,\ i \neq j \in N\}$
p	A set of 60-minute periods within a recovery period
$I_i(p)$	All acrs (i,j) departing from node i during period p
$O_j(p)$	All acrs (i,j) arriving at node j during period p
a_{ip}	Arrival capacity at airport i in period p
b_{ip}	Departure capacity at airport i in period p
f	Aircraft, $f \in F = F_c \cup F_d \in F,\ f = \{1, \ldots n\}$
k	Itinerary which consists of flight legs characterized by a cabin class, $k \in K$
c_{ijf}	Operational Costs, including fuel, maintenance and crew of aircraft f on arc (i,j)
r_{ij}	Delay cost of arc (i,j)
e_{ij}	Cancelation cost of arc (i,j)
h_{ij}	Legal delay cost of arc (i,j)
f_{ij}	Legal cancelation cost of arc (i,j)
g_{ijf}	Downgrading cost of aircraft f of arc (i,j)
L^f_{fam}	An aircraft f of the same family at the end of the recovery period
L^f_{mod}	An aircraft f of the same model at the end of the recovery period
L^f_{conf}	An aircraft f of the same configuration at the end of the recovery period
Auxiliary or decision variables	**Explanation**
x_{ijf}	1, if and only if aircraft f uses arc (i,j); 0, otherwise

$$\min \sum_{i,j\in N}\sum_{f_d\in F}(r_{ij}+h_{ij})x_{ijf} + \sum_{i,j\in N}\sum_{f_c\in F}(e_{ij}+f_{ij})x_{ijf} + \sum_{i,j\in N}\sum_{f\in F}c_{ijf}x_{ijf} + \sum_{i,j\in N}\sum_{f_c,f_d\in F}g_{ijf}x_{ijf} + \sum_{f\in F}(L^f_{fam}+L^f_{mod} + L^f_{conf})x_{ijf} \quad (1)$$

$s.t.$

$$\sum_{i,j\in N}x_{ijf} - \sum_{i,j\in N}x_{jif} = 0, \forall f \in F, \forall i,j, i \neq j \in N \backslash \{o_f, d_f\} \quad (2)$$

$$\sum_{j\in N}x_{o_f jf} = 1, \forall f \in F, \forall j \in N \quad (3)$$

$$\sum_{(i,j)\in I_j(p)}\sum_{f\in F}x_{ijf} \leq a_{jp}, \forall i,j, i \neq j \in N, \forall p \in P \quad (4)$$

$$\sum_{(i,j)\in O_i(p)}\sum_{f\in F}x_{ijf} \leq b_{ip}, \forall i,j, i \neq j \in N, \forall p \in P \quad (5)$$

$$\sum_{f\in F}x_{ijf} \leq 1, \forall i,j, i \neq j \in N \quad (6)$$

A large neighbourhood search heuristic was developed to solve the airline schedule disruption recovery problem. Large neighbourhood search was introduced by Shaw and originally served the purpose of solving vehicle routing problems. It is based on the idea that an initial solution can be improved by constantly destroying parts of the solution and repairing them afterwards. The heuristic applied in this paper consists of three phases, namely the construction phase, repair phase, and improvement phase. The first two phases are constructed complying with the constraints set in the previous part of this paper and serve the purpose of producing a feasible initial solution. The improvement phase will make random changes to the solution one at a time and compare the results. Solutions are compared and replaced based on the total cost while the feasibility of the schedule is strictly maintained.

The construction and repair phases will be executed repeatedly to yield several initial solutions until the computing time limit is reached or that a certain number of consecutive runs fail to produce an improvement. The initial solution with the lowest cost will then be used in the third phase for further improvement, and the improvement phase will be executed until the computing time limit runs out. The pseudo-code of the large neighbourhood search algorithm can be found in Sinclair, et al. [25] and Guimarans, et al. [26].

A. Construction Phase

In this phase, the first step is to shuffle all aircraft in the given set and start working on each one of their rotations one by one. This gives the solutions a feature of randomness and ensures that various initial solutions can be found.

For each one of these aircraft, the algorithm checks its flights and find those that become infeasible strictly due to delays of one of its previous flights. For these flights, we try delaying them by increments of 30 minutes to see if that can resolve the infeasibility. An upper limit of 16 hours is set so that flights do not get infinitely delayed in search of a solution. For flights that are cancelled, we try to create similar flights first. This is based on the aircraft models that have the same configuration. The step will slot the newly created flight into normal intervals between two flights. However, if no interval is feasible for execution of the newly created flight, the shortest loop containing the cancelled flight will be sought out and removed from the rotation of the aircraft. If infeasibility persists, all the loops after the cancelled flight will be cancelled within the aircraft's rotation.

Next, maintenance constraints will be taken into account in the algorithm. For flights that become infeasible because of maintenance conflicts, we try removing the shortest loop from the rotation again to make the rotation feasible. Else, we will cancel all the loops from the flight in conflict to the end of the rotation.

The last step in this phase will consider airport capacity constraints. Flights that have infeasibility caused by insufficient airport capacities will be delayed by increments of 30 minutes until the feasibility is reached or the maximum delay time of 16 hours is reached. If feasibility cannot be reached by executing the previous steps, the algorithm will try to remove the shortest loop in the rotation or cancel all the loops till the end of the rotation.

B. Repair Phase

This phase consists of three major steps, focusing on flight rotation insertion, itinerary cancellation, and itinerary repair, respectively. The airport capacity constraint is checked again, the first in-case-violations were introduced during the last steps in the construction phase. Immediately thereafter, we try to reinsert the loops that were cancelled in the previous steps. Itineraries that have become infeasible would be cancelled in this phase, and compensation costs of these cancelled itineraries will be calculated. These itineraries will be sorted in decreasing order of compensation costs since we want to tackle the ones with the highest costs first. The shortest path algorithm from the C++ libraries is used to assign passengers with cancelled itineraries into flights.

C. Improvement Phase

In this phase, the best solution produced by the construction and repair phases will be destroyed and repaired constantly in the hope of finding further improvements. Flights will be delayed randomly to find out if the schedule can accommodate more passengers to reduce the eventual total cost. In this phase, itineraries that were formerly feasible may become infeasible due to the deliberately delayed flights that create intervals in the schedule. Passengers will be reassigned in the schedule. Any solution that yields an improvement will replace the previous solution. If no improvement arises within 5 minutes of computing time, the procedure will end.

III. COMPUTATIONAL EXPERIMENTS

For each instance, the number of aircraft, airports, and flights is specified. The number of itineraries is also factored in. Note that each itinerary may contain one or more than one passenger due to how the booking process works in reality. Each instance has a different amount of disruptions in respective legs of flights, aircraft, or airport. We can generally expect that instances with all three types of disruptions will yield higher recovery costs. Table II shows the description of test instances. The number of aircraft is represented by f; a represents the number of airports; fl represents the number of flights; k represents the number of itineraries; fl_d represents the number of flight disruptions; f_d represents the number of aircraft disruptions and a_d represents the number of airport disruptions.

The configuration of the computation unit consists of an Intel Core i7-4790 3.60GHz CPU and 16 RAM and Microsoft Windows 10 operating system. The program is written in C++ using the QT Framework with Object-Oriented Design approach. Ten sets of data provided by the 2009 ROADEF Challenge were used to test the performance, stability, and comprehensiveness of the algorithm.

TABLE I
DESCRIPTION OF THE TEST INSTANCES

Instance ID	f	a	fl	k	fl_d	f_d	a_d

1_1	255	45	1423	11214	230	0	0
1_2	256	45	1423	11214	255	0	0
1_3	256	45	1423	11214	229	1	0
1_4	256	45	1423	11214	230	0	1
1_5	256	44	1423	11214	0	0	34
1_6	256	45	1423	11565	230	0	0
1_7	256	45	1423	11565	255	0	0
1_8	256	45	1423	11565	229	1	0
1_9	256	45	1423	11565	230	0	1
1_10	256	44	1423	11565	77	0	34

IV. COMPUTATIONAL RESULTS

Tests of each instance were run 10 times, each run lasting 20 minutes. The average number of iterations and costs were calculated. Feasible schedules were produced for all 10 data instances, which proved the algorithm's comprehensiveness and its capability at handling all three types of disruptions. It is also observed that the solutions generated by the construction and repair phase can significantly improve after only a few runs of the improvement phase, which demonstrated the algorithm's efficiency in producing modified schedules that are more refined than the Initially produces schedules. Table III reveals the computation results. In the 10 data sets that were used, the improvement over the solutions produced during the construction and repair phase was often approximately 50%. The algorithm shows good potential for further improvements, should real-life situations that allow for longer computational time arise.

TABLE II
COMPUTATIONAL RESULTS

Instance ID	C&R	Average Cost	Imp.	N. Average Cost
1_1	556	2044374.77	9	865523.18
1_2	556	3433845.36	9	1597331.40
1_3	556	2329388.3	9	949639.40
1_4	602	2157158.07	9	882656.97
1_5	297	2404413.59	6	1535512.35
1_6	556	7564183.33	9	4060300.65
1_7	556	16482456.69	9	10878421.18
1_8	556	5357592.43	9	3814555.26
1_9	624	10354358.47	9	5375641.40
1_10	282	62514738.19	6	48253133.16

V. CONCLUSION

High disruption costs provide stronger motivation for improving the efficiency, speed, and stability in airline schedule disruption recovery. In this paper, a large neighbourhood search approach was developed to tackle the problem. The algorithm can run on all of the data sets and produce effective results without producing errors. Significant improvements can be observed in each phase, and modified schedules are obtained within 20 minutes. It also offers great built-in flexibility. The weights assigned to the variables in the mathematical formulae can be adjusted to meet the different needs of airlines of various sizes. The runtime can be adjusted, which allows the algorithm to produce satisfactory results in emergency scenarios or, given more time, better results can be obtained. The managerial insights are to reschedule in the case of disruptions that takes into considerations of various factors to simulate real-world environments and to develop an efficient, dynamic solution to airline schedule disruption. The algorithm can be further optimised for consistent, short delays by such various means as adjusting the neighbourhood sizes for conducting the neighbourhood search.

ACKNOWLEDGEMENT

The research is supported by *Interdisciplinary Division of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong SAR* and *Department of Industrial and Systems Engineering*, *The Hong Kong Polytechnic University*, Hong Kong SAR and *College of Professional and Continuing Education*, *The Hong Kong Polytechnic University*, Hong Kong SAR, China. Our gratitude is also extended to the Research Committee and the *Interdisciplinary Division of Aeronautical and Aviation Engineering*, *The Hong Kong Polytechnic University* for support of the project (BE3V).

REFERENCES

[1] K. K. H. Ng, C. K. M. Lee, F. T. S. Chan, and Y. Lv, "Review on meta-heuristics approaches for airside operation research," *Applied Soft Computing*, vol. 66, pp. 104-133, 2018/05/01/ 2018, doi: https://doi.org/10.1016/j.asoc.2018.02.013.

[2] K. L. Keung, C. K. M. Lee, P. Ji, and K. K. H. Ng, "Cloud-based Cyber-Physical Robotic Mobile Fulfillment Systems: A Case Study of Collision Avoidance," *IEEE Access*, pp. 1-1, 2020, doi: 10.1109/ACCESS.2020.2992475.

[3] K. L. Keung, C. K. M. Lee, K. K. H. Ng, and C. K. Yeung, "Smart City Application and Analysis: Real-time Urban Drainage Monitoring by IoT Sensors: A Case Study of Hong Kong," in *2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 16-19 Dec. 2018 2018, pp. 521-525, doi: 10.1109/IEEM.2018.8607303.

[4] N. Boysen, R. de Koster, and F. Weidinger, "Warehousing in the e-commerce era: A survey," *European Journal of Operational Research*, vol. 277, no. 2, pp. 396-411, 2019/09/01/ 2019, doi: https://doi.org/10.1016/j.ejor.2018.08.023.

[5] K. K. H. Ng, C. K. M. Lee, and F. T. S. Chan, "A robust optimisation approach to the aircraft sequencing and scheduling problem with runway configuration planning," in *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Singapore, Singapore, 2017: IEEE, pp. 40-44, doi: 10.1109/IEEM.2017.8289847.

[6] M. Samà, A. D'Ariano, F. Corman, and D. Pacciarelli, "Metaheuristics for efficient aircraft scheduling and re-routing at busy terminal control areas," *Transportation Research Part C: Emerging Technologies*, vol. 80, pp. 485-511, 7// 2017, doi: https://doi.org/10.1016/j.trc.2016.08.012.

[7] M. Samà, A. D'Ariano, P. D'Ariano, and D. Pacciarelli, "Scheduling models for optimal aircraft traffic control at busy airports: Tardiness, priorities, equity and violations considerations," *Omega*, vol. 67, pp. 81-98, 2017/03/01/ 2017, doi: https://doi.org/10.1016/j.omega.2016.04.003.

[8] M. Samà, A. D'Ariano, P. D'Ariano, and D. Pacciarelli, "Optimal aircraft scheduling and routing at a terminal control area during disturbances," *Transportation Research Part C: Emerging Technologies*, vol. 47, pp. 61-85, 2014/10/01/ 2014, doi: http://dx.doi.org/10.1016/j.trc.2014.08.005.

[9] J. Huo, F. T. S. Chan, C. K. M. Lee, J. O. Strandhagen, and B. Niu, "Smart control of the assembly process with a fuzzy control system in the context of Industry 4.0," *Advanced Engineering Informatics*, vol. 43, p. 101031, 2020/01/01/ 2020, doi: https://doi.org/10.1016/j.aei.2019.101031.

[10] K. K. H. Ng, C. K. M. Lee, F. T. S. Chan, and Y. Qin, "Robust aircraft sequencing and scheduling problem with arrival/departure delay using the min-max regret approach," *Transportation Research Part E: Logistics and Transportation Review*, vol. 106, pp. 115-136, 2017/10/01/ 2017, doi: https://doi.org/10.1016/j.tre.2017.08.006.

[11] A. E. E. Eltoukhy, F. T. S. Chan, and S. H. Chung, "Airline schedule planning: a review and future directions," *Industrial Management & Data Systems*, vol. 117, no. 6, pp. 1201-1243, 2017, doi: doi:10.1108/IMDS-09-2016-0358.

[12] A. Harada, T. Ezaki, T. Wakayama, and K. Oka, "Air traffic efficiency analysis of airliner scheduled flights using collaborative actions for renovation of air traffic systems open data," *Journal of Advanced Transportation*, vol. 2018, 2018.

[13] D. T. Sanchez, B. Boyacı, and K. G. Zografos, "An optimisation framework for airline fleet maintenance scheduling with tail assignment considerations," *Transportation Research Part B: Methodological*, vol. 133, pp. 142-164, 2020/03/01/ 2020, doi: https://doi.org/10.1016/j.trb.2019.12.008.

[14] K. K. H. Ng and C. K. M. Lee, "Makespan minimization in aircraft landing problem under congested traffic situation using modified artificial bee colony algorithm," in *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Bali, Indonesia, 4-7 Dec. 2016 2016: IEEE, pp. 750-754, doi: 10.1109/IEEM.2016.7797976.

[15] Y. Qin, F. T. S. Chan, S. H. Chung, T. Qu, and B. Niu, "Aircraft parking stand allocation problem with safety consideration for independent hangar maintenance service providers," *Computers & Operations Research*, vol. 91, pp. 225-236, 2018/03/01/ 2018, doi: https://doi.org/10.1016/j.cor.2017.10.001.

[16] K. K. H. Ng, M. H. M. Tang, and C. K. M. Lee, "Design and development of a performance evaluation system for the aircraft maintenance industry," in *2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 6-9 Dec. 2015 2015, pp. 564-568, doi: 10.1109/IEEM.2015.7385710.

[17] Y. Qin, Z. X. Wang, F. T. S. Chan, S. H. Chung, and T. Qu, "A mathematical model and algorithms for the aircraft hangar maintenance scheduling problem," *Applied Mathematical Modelling*, vol. 67, pp. 491-509, 2019/03/01/ 2019, doi: https://doi.org/10.1016/j.apm.2018.11.008.

[18] Y. Qin, Z. X. Wang, F. T. S. Chan, S. H. Chung, and T. Qu, "A Family of Heuristic-Based Inequalities for Maximizing Overall Safety Margins in Aircraft Parking Stands Arrangement Problems," *Mathematical Problems in Engineering*, vol. 2018, p. 3525384, 2018/07/26 2018, doi: 10.1155/2018/3525384.

[19] Ö. Şafak, A. Atamtürk, and M. Selim Aktürk, "Accommodating new flights into an existing airline flight schedule," *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 265-286, 2019/07/01/ 2019, doi: https://doi.org/10.1016/j.trc.2019.05.010.

[20] C. K. Lee, S. Zhang, and K. K. Ng, "Design of An Integration Model for Air Cargo Transportation Network Design and Flight Route Selection," *Sustainability*, vol. 11, no. 19, p. 5197, 2019.

[21] K. K. H. Ng, C. K. M. Lee, F. T. S. Chan, C.-H. Chen, and Y. Qin, "A two-stage robust optimisation for terminal traffic flow problem," *Applied Soft Computing*, vol. 89, p. 106048, 2020/04/01/ 2020, doi: https://doi.org/10.1016/j.asoc.2019.106048.

[22] K. K. H. Ng, C. K. M. Lee, S. Z. Zhang, K. Wu, and W. Ho, "A multiple colonies artificial bee colony algorithm for a capacitated vehicle routing problem and re-routing strategies under time-dependent traffic congestion," *Computers & Industrial Engineering*, vol. 109, pp. 151-168, 2017/07/01/ 2017, doi: https://doi.org/10.1016/j.cie.2017.05.004.

[23] S. Bisaillon, J.-F. Cordeau, G. Laporte, and F. Pasin, "A large neighbourhood search heuristic for the aircraft and passenger recovery problem," *4OR*, vol. 9, no. 2, pp. 139-157, 2011/06/01 2011, doi: 10.1007/s10288-010-0145-5.

[24] R. Britto, M. Dresner, and A. Voltes, "The impact of flight delays on passenger demand and societal welfare," *Transportation Research Part E: Logistics and Transportation Review*, vol. 48, no. 2, pp. 460-469, 2012/03/01/ 2012, doi: https://doi.org/10.1016/j.tre.2011.10.009.

[25] K. Sinclair, J.-F. Cordeau, and G. Laporte, "Improvements to a large neighborhood search heuristic for an integrated aircraft and passenger recovery problem," *European Journal of Operational Research*, vol. 233, no. 1, pp. 234-245, 2014/02/16/ 2014, doi: https://doi.org/10.1016/j.ejor.2013.08.034.

[26] D. Guimarans, P. Arias, and M. M. Mota, "Large neighbourhood search and simulation for disruption management in the airline industry," in *Applied Simulation and Optimization*: Springer, 2015, pp. 169-201.