United Nations Humanitarian Air Service: Network Optimization

A Tabu Search Approach

Beihong Yang

TU Delft

United Nations Humanitarian Air Service: Network Optimization

A Tabu Search Approach

by

Beihong Yang

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday 17th of February, 2022 at 13:00.

Student number:	4456858
Project duration:	August 2020 – February 2022
Thesis committee:	Dr. B.F. Santos
	Ir. P.C. Roling
	Dr. F. Oliviero

An electronic version of this thesis is available at http://repository.tudelft.nl/.
Cover image credits taken from [1]

TU Delft

Acknowledgements

At the beginning of 2020, I was attracted by this project topic on the board of ATO group on the fourth floor of the faculty. Now the year 2022 has begun, I finally finished this thesis through numerous obstacles and interruptions, but our journey with the pandemic has not ended yet. These two years flew much faster than any former years in my life. After the whole world broke down because of the COVID-19, every day is almost the same day, with little vitality. I could not live with my own rhythm anymore and I felt lost in the forest. I have been struggling with depression and procrastination for almost a year and thankfully, I finally overcame these difficulties steadily and accomplished my work.

First I would like to thank my family for supporting me financially and spiritually. From the day I decided to apply for the bachelor programme of aerospace engineering at TU Delft, they have been encouraging me to study what I am interested in and to pursue my goal. Thank you for your understanding when I feel depressed and frustrated

Secondly, I would like to thank my daily supervisor Paul and Alessandro for your guidance and comments on my work. I apologise for this long-time duration to accomplish the thesis and I really appreciate your patience. I have learnt a lot during this project. Not only my programming skills have improved, but also I comprehend how to write a scientific report in a professional way. It is also nice to hear your advice and suggestion on my questions about doing research in the future. I would also like to thank Dr. Qichen Deng for his help on some problems I encountered when developing the model. He also helped me a lot with my questions and worries about PhD application.

Finally, I would like to thank all of my friends, no matter graduated or not, in Delft or anywhere in the world. Thank you for your accompany during this tough time. I would like to especially thank my Roland BrotherSisterhood, thank you all for cheering me up when I am down and it is so nice to have you guys around almost every weekend. I believe our friendship could last forever and wish you all the best for your study and future career!

This marks the end of my master programme and probably the end of my 5-year study life at TU Delft. As a young graduate, there is still a lot to explore and I hope to find my position in this society soon and shine.

Beihong Yang
Delft, January 2022

Contents

List of Figures

List of Tables

Introduction

The United Nations Humanitarian Air Service (UNHAS) is a transportation service from The World Food Program's (WFP)[6], whose aim is to deliver assistance and supply to needed places. Aviation provides the possibility to reach the rural and isolated areas that are not accessible by other way of transportation. However, due to the high expenses of air operation and the fact that UNHAS is a non-profit operator, it is essential to conduct optimisation of the flight schedule to save the operational cost while satisfying maximal demand.

This thesis is a continuous study of the UNHAS network optimisation topic done by previous master graduates. The previous studies focused on forming the optimisation model, which focused on solving the problem in the real application. However, many assumptions have been made in the previous studies to simplify the scenario for the research. In this thesis, the main focus is put on solving the UNHAS optimisation problem with metaheuristic method, and the model is designed to satisfy more requirements and considers a more general scenario. Moreover, the project provides an assisting as well as a decision-making tool for the planner to generate the optimal flight plan or verify the possibility of improving an existing flight plan.

This thesis report is organised as follows : In Part I, the scientific paper is presented. Part II contains the relevant Literature Study that supports the research. Finally, in Part III, the fundamental design logic of the constraints are presented.

I

Scientific Paper

United Nation Humanitarian Air Service: Network Optimisation

Beihong Yang*, P.C. Roling †

Delft University of Technology, Delft, The Netherlands

Abstract

Network scheduling and fleet assignment are essential tasks for airline operation. In order to generate an optimal flight plan, the flight route and the flight schedule of each aircraft in the fleet requires deliberated consideration and planning. Compared with commercial airlines, which are in pursuit of maximal benefit during the operations, the humanitarian air services have different goals and therefore different strategies are designed. The humanitarian air service dedicates to fulfilling maximal passenger requests by reacting in a relatively short time frame, and the overall cost efficiency needs to be maximised. In this research, the United Nations Humanitarian Services (UNHAS) South Sudan mission is taken as the case to study and a metaheuristic method on top of the multi-integer linear programming (MILP) model is designed to solve the optimisation problem. The optimisation process consists of two stages: the tabu search process to assign the flight routes among the fleet, and a variation of a Fleet Size and Mix Vehicle Routing Problem (FSMVRP) model to finally determine the time schedule of each aircraft. The model is able to unlimitedly split the passenger requests and recaptures passenger spillage. It considers much fewer assumptions during the solving process and it provides large flexibility for the planner to manually modify the model based on their purpose. A dynamic balance of aircraft utilisation time regarding the Minimum Guaranteed Hours (MGH) within the fleet is also discussed. The result of this method is compared with the previous study of S.P. Niemansburg, which shows 1% to 11% of cost saved on a single day's operation regarding different levels of passenger spillage.

1 Introduction

This thesis report presents the design of optimisation model for the UNHAS network problem and the evaluation of its performance.

The United Nation Humanitarian Air Service (UNHAS) is the world's main transporter of humanitarian personnel and aid, which provides aviation logistics to places of natural disasters and emergencies around the world. It is a non-commercial operator, which provides aviation service to not only the staff of the World Food Program (WFP)[wfp, 2021], but also other UN agencies and non-governmental organisations (NGOs) to deliver supplies to where it is needed[Dorn, A. W., 2014]. The project aims to provide air transport service for delivery of life-saving assistance based on the request and to transport humanitarian workers to where they are needed the most.

The UNHAS network optimisation problem is one type of vehicle routing problems. To be more specific, it is a capacitated VRP (CVRP), which is NP-hard and time-consuming to solve (Garey and Johnson[Garey, M. R. and Johnson, D. S., 1990]). Contrary to the general strategy of a commercial aviation company, which focuses on cost minimisation, profit maximisation and optimal utilisation of a certain fleet type (Abara[Abara, J., 1989]), the UNHAS emphasises more on cost minimisation of aircraft lease and efficiency maximisation when planning the flight schedule. Similar research and discussion have been made in other humanitarian programmes: Eftekar et al.[Eftekhar, M., Masini, A., Robotis, A. and Van Wassenhove, L. N., 2013] researched the real-life operations of the International Committee of the Red Cross (ICRC)[ICR, 2021] in order to seek an optimal policy that can be implemented for the operations. Liu et al.[Liu, M., Cao, J., Liang, J. and Chen, M., 2020] focused on the epidemic logistics and developed the medical resource allocation models under different level of supplies with the help of epidemic dynamic method.

Due to this difference in functionality, the problem cannot be directly solved by the regular aircraft assignment model, but few adjustments are needed in order to add relevant criteria into the model. For example, the total cost needs to be constrained by the budget, and due to the speciality of this mission, there is no revenue part but the number of transported passengers and the proportion of served passengers need to be maximised.

The report is structured as follows: An overview of popular VRP variants, the existing heuristics and an additional emphasis of the tabu search method are introduced in section 2. After applying the tabu search

*Msc Student, Air Transport and Operations, Faculty of Aerospace Engineering, Delft University of Technology

†Aerospace Engineering Faculty, Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands; **p.c.roling@tudelft.nl**

strategy to the model, the resultant metaheuristic method model is introduced in section 3. The verification of the model and validation of the result is present in section 4. Finally, the conclusion is drawn in section 5 to discuss the overall performance with recommendation for future research.

2 Literature Review

2.1 Vehicle routing problem

The vehicle routing problem (VRP) is a set of problems for optimal routing and scheduling. All VRPs are NP-hard, and the problem can be solved by forming linear programming model. Dantzig introduced the simplex method to solve the linear programming problem in a mathematically tractable way[Nash, S.G., 1990]. Based on this method, every VRP model can be formed mathematically as follow:

$$\text{Minimise/Maximise} \qquad \mathbf{c'x} \tag{1}$$

$$\begin{aligned} \textit{subject to} \qquad \mathbf{A_1x} &\leq \mathbf{b_1} && (2)\\ \mathbf{A_2x} &= \mathbf{b_2} && (3)\\ \mathbf{A_3x} &\geq \mathbf{b_3} && (4)\\ \mathbf{x} &\geq \mathbf{0} && (5) \end{aligned}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_1} \\ \mathbf{A_2} \\ \mathbf{A_3} \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} \mathbf{b_1} \\ \mathbf{b_2} \\ \mathbf{b_3} \end{bmatrix} \tag{6}$$

and $\mathbf{c'}$ is the cost coefficient matrix of all the decision variables $\mathbf{x}$[Nash, J. C., 2000].

In the context of airline operation, the common VRPs are network scheduling and planning problems. The necessity of both integer and continuous decision variables lead most of the problems to be formed as mixed-integer linear programming (MILP) model. Due to the characteristic of humanitarian airline mission, it does not seek a sole objective of maximal profit from the operation. The UNHAS is an operator that transports passengers and delivers supply with no profit, and these requests should be fulfilled to the utmost. The operator is funded by the United Nations and the operational budget is limited, therefore the operational cost of the mission should be constrained.

The objective function is the most essential element in the optimisation model, it provides the ultimate goal the problem is pursuing. In contrast to the majority of VRPs that only have one objective function to consider, these goals of UNHAS mission lead the problem to be solved by considering two objectives: maximisation of passenger requests and minimisation of operational cost. The weights of two sides are hard to be quantified with lack of information. The conflict between these two aspects makes the problem impossible to be solved with a mono-objective model. In order to resolve this dilemma, a MILP model with hierarchical objective is selected. The hierarchical objective function is commonly applied with the heuristics, while the exact algorithms do not take account the number of vehicles in the objective function[Bräysy, O. and Gendreau, M., 2005a][Bräysy, O. and Gendreau, M., 2005b]. A common hierarchical approach is to split the optimisation process into multiple steps to avoid more than one objective function for the model. For example, in the Vehicle Routing Problem with Time Windows (VRPTW), the minimisation of the route length can not be achieved when the minimisation of the deployed vehicles is considered as well, since the utilisation of the vehicles and drivers leads to high fix costs with respect to the routing plan. One can first optimise the number of vehicles and then the second objective can be optimised with the fixed result from the first optimisation problem.

In the UNHAS scenario, the passenger delivery is considered as the leading objective. When the same amount of passenger is delivered, the following step is searching for minimal operational cost.

2.2 Metaheuristics

Since 1956, the first study of VRP is done by Flood[Flood, M. M., 1956] to solve the famous travelling salesman problem (TSP), the methodology has been continuously developing for a variety of VRPs for decades. Along with the research of more complex VRPs, the mathematical algorithm is improved as well to obtain faster and better results. Especially with the assistance of computer and programming languages, the solving process becomes faster and relatively larger scale LP problem are solvable.

Dantzig introduced the simplex method to solve the TSP in a mathematically tractable way and provided the general procedure to solve linear programming problem. The simplex method is a pure algebraic procedure, which is beneficial to transform and solve the problem on the computer. Geometrically, the constraints form a polytope in the n-dimensional coordinate system to represents the constraints in the problem and the objective function is applied to the polytope to find the optimal solution[Nash, S.G., 1990]. The famous CPLEX Optimiser developed by IBM[cpl, 2021] is also developed based on the simplex method in the C language.

The development in the transportation industry leads to more complicated and large problems to solve, which reveals the disadvantages of the exact method. Exact algorithm consumes an enormous amount of computational time to seek the global optimum when applied to an LP problem with a large network. Based on the study of Toth and Vigo, the exact algorithm is not applicable to consistently solve a VRP problem with more than 50 customers[Toth, P. and Vigo, D., 2015].

A heuristic technique is a way to solve the problems in a reasonable time. The result may not be optimal, but is close enough to the global optimum which the exact method can possibly obtain. The simplest heuristics include trial and error and rule of thumb, but these methods are quite inefficient when solving large scale problem. The heuristic function approximates the exact method solution by searching the branching steps to follow the branch with the best result[Pearl, J, 1984]. The main heuristic methods are constructive heuristics, improvement heuristics and metaheuristics.

The main task of this thesis is to seek the possibility of applying metaheustic method to solve the UNHAS network optimisation problem. Metaheuristic algorithms are high-level procedures that select various lower-level heuristics to perform a partial search in the solving process. Metaheuristics can provide sufficiently good solution for optimisation problem with incomplete information and limited computation capacity[Vikhar, P. A., 2016]. Based on the searching process, metaheuristics are categorised into two groups: local search algorithms and population-based algorithms.

The local search algorithm searches the best result in the domain from a single solution, where the population-based algorithm evolves multiple solutions to generate the optimal result. For model design of the UNHAS scenario, not only the optimal solution is pursued, but also the solving process should not be too complicated and time-consuming. The solving process is considered to be initialised from a single solution, as multiple starting points are not necessary for the UNHAS case. Therefore, a local search algorithm is chosen. Several existing local search algorithms that have been used for VRPs are simulated annealing (SA)[Osman, I. H., 1993], tabu search (TS)[Glover, F., 1986] and iterated local search (ILS)[Chen, P., Huang, H-K. and Dong, X-Y., 2010].

The tabu search method is chosen in this case, as the evolution of the flight schedule is the key in the searching process. Tabu search algorithm was initially invented by Fred W. Glover[Glover, F., 1986] in 1986 and formally introduced in 1989. It is an algorithm that tries to enable the search process to escape from a local optimum, and it continues to search the neighbourhood to find the global optimum[Coello, C. C., Lamont, G. B. and Van Veldhuizen, D. A., 2002][Hillier, F.S. and Lieberman, G.J., 2014]. It uses the tabu list to generate the areas that have been searched during the previous iterations, and the list is used as a reference to discourage the search from coming back to the previously-visited solutions and therefore avoid cycling.

2.3 Research gap

The first stage of the problem has been established by S.P. Niemansburg during his master thesis project in 2019[Niemansburg, S.P., 2019], which solved the humanitarian flight optimisation in South Sudan on a daily timescale. In the following years, the topic has been expanded and researched by other students as well[Mekking, Y.C., 2020][Billet, T., 2021], focusing on the humanitarian application. There are few limitations in his model:

1. The request division is determined manually.
2. The planning horizon of the model is short.
3. The recapture of spilt passengers are not considered in the planning.
4. No anticipation of the possible future demands.

In addition, the model needs to be improved further with new requirements: decision support tool development for the tasking officers, and the consideration of minimum guaranteed hours (MGH) for the contracts.

In conclusion, the master thesis project this year focuses on the following fields:

1. Optimisation by considering past and future demands.
2. Optimal dividing of requests over flights.
3. Analysis in network effects and route dependencies
4. Decision support tool development.
5. Implementation of minimum guaranteed hours (MGH) requirements.

The points listed above brought new challenges and possibilities to explore alternatives to solve the problem. Scheduling over a planning horizon and consideration of the MGH requirements lead to VRP with time windows (VRPTW). The heuristic method is one of the many options to discover relatively good feasible solution within the required time limit. Coello et al.[Coello, C. C., Lamont, G. B. and Van Veldhuizen, D. A., 2002] introduced multiple metaheuristic methods to solve multi-objective problems. In their study, the tabu search algorithm is selected as the most suitable algorithm for the vehicle routing problems.

Therefore, the main research objective of this thesis is:

"To achieve a cost-efficient flight scheduling of the UNHAS concerning the operational and safety constraints to the non-commercial humanitarian setting by means of a decision support tool that improves the demand satisfaction with the help of metaheuristic method"

To achieve the design objective, a theoretical research question needs to be reformulated to provide a clear specification of the aspects that need to be considered during the project:

"How to develop a decision-making model that provides a cost-effective flight schedule by considering both the past and future demand?"

In order to answer the research question, the following sub-questions are framed:

1. Which type of VRP is this problem?
2. Which aspects to be included in the objective function?
3. What are the shortcomings of the existed algorithm?
4. Which metaheuristic is chosen for the model?
5. How to make a fast decision when choosing the most cost-effective aircraft?
6. How to sufficiently anticipate future demand?
7. How many days the time window should be?
8. How much margin should be reserved per day for possible spilt passengers?
9. How to meet the requirements of minimum guaranteed hours (MGH) from the contracts?

3 Methodology

The UNHAS network optimisation problem is a sub-problem of Vehicle Routing Problem (VRP), which is a NP-hard problem. When dealing with large-scale problem with enormously large decision variables and constraints, it is sometimes time-consuming as well as difficult to solve the problem with the exact method. Tabu search algorithm is one of the possible solutions for it, where the solution space is examined from an initial point to find the optimal result. As multiple criteria are considered in this project, and they are mutually relative to each other. A pure tabu search algorithm has a complicated structure and it is inconvenient to form it in this project. Therefore, a tabu search logic with the assist of the CPLEX is implemented.

The main goal of the UNHAS mission is to deploy aircraft and deliver passengers successfully based on their request, so the main issue is to assign aircraft to deliver the passengers from the requests. However, due to the fact that flight transfer is possible during the daily operation, it is not guaranteed that the passenger will stay in the same aircraft along the way. Therefore, distribution of aircraft's capacity to different requests during the operation is also another aspect to be considered.

3.1 Layout

The scenario of 2015 UNHAS operation in South Sudan is considered in this project, which contains 44 airports among the country and a fleet of 15 aircraft for the operation. All the operating airports in South Sudan is shown in Table 8 of Appendix C and the characteristics of the fleet for the operation is listed in Table 1.

The UNHAS network problem focuses on the flight leasing plan in South Sudan, which aims to satisfy maximum passenger delivery as well as constraining the total operational cost. Passenger delivery and the operational cost are correlate to each other, where less passenger delivery can possibly reduce the operational cost and vise versa. However, due to the fact that the UNHAS mission depends on the funding from the United Nations, governments and NGOs worldwide, the passengers spend little to none when taking the service.

Passenger spillage is considered to have almost no extra cost or penalty from the operator's side. Therefore, the passenger spillage cannot be quantified in the same scale of the operational cost, which leads the merging of two objectives in a reasonable way almost impossible.

The dual objectives define the UNHAS network problem as a bi-objective mixed-integer linear programming (MILP) problem. The bi-objective optimisation problem can be destructed as an single-objective optimisation problem if the value of the other objective can be defined. All the combinations of the amount of delivered passenger and the corresponding minimal operational cost construct the Pareto front, which is essential for the further analysis and decision-making of the trade-off.

In order to construct the MILP model for the solver, the mission goals and requirements need to be interpreted in the form of objective functions and constraints with varies sets of decision variables.

Aircraft	**Aircraft type**	**Cruising Speed [nm/hr]**	**Cost [-/nm]**	**Seats**	**Range [nm]**	**Runway required [m]**	**Hub**
Fokker 50	Fokker 50	230	20	50	1080	1500	Juba
Dash 8_1	DHC8-106	200	18	37	1020	2000	Juba
Dash 8_2	DHC8-202	200	17	37	1020	2000	Juba
Dornier 228	Dornier 228	220	11	15	1000	1000	Juba
Cessna 208_1	Cessna 208B	180	10	10	1070	1000	Juba
Cessna 208_2	Cessna 208B	180	9.2	10	1070	1000	Juba
Cessna 208_3	Cessna 208B	180	9.5	10	1070	1000	Juba
Cessna 208_4	Cessna 208B	180	10.2	10	1070	1000	Juba
Cessna 208_5	Cessna 208B	180	9.8	10	1070	1000	Juba
MilMi8_1	Mi8-T	120	32	17	355	50	Juba
MilMi8_2	Mi8-T	120	33	17	355	50	Juba
Cessna 208_1R	Cessna 208B	180	11	10	1070	1000	Rumbek
Cessna 208_2R	Cessna 208B	180	10.5	10	1070	1000	Rumbek
MilMi8_1R	Mi8-T	120	32	17	355	50	Rumbek
MilMi8_2R	Mi8-T	120	31	17	355	50	Rumbek

Table 1: Fleet used by UNHAS in South Sudan

3.2 Assumption

When premeditating the daily aircraft operation, unforeseen circumstances are always present. Thus, it is essential to maintain the practicality of the model in the real-life scenario while simplifying its complexity when designing the MILP model. The following assumptions are defined based on the objective requirements, technical limitations and operation strategy:

- **Aircraft refuelling**: Only designated airports are considered as aircraft refuelling stations, while each aircraft can always refuel at its own hub. In the UNHAS scenario, 5 airports are designated to provide the refuelling service: Juba, Rumbek, Wau, Bor and Malakal.
- **Aircraft speed**: Each aircraft is assumed to fly from one airport to another with a steady speed.
- **Time block**: A distributed timeline is considered for the flight plan, where the time unit is 1 minute.
- **Max. number of flight legs**: Every aircraft is limited to have maximum six flight legs per day.
- **Minimum daily passenger delivery**: It is restricted to satisfy minimal 75% of the total passenger delivery during the daily operation.
- **Aircraft revisit**: Besides its own hub, every aircraft can only visit the same airport twice during the day, while the hub can be visited once in the middle of the route before returning back to the hub.
- **Flight transfer**: Maximal one transfer is considered when generating the model for the request. However, more than one time transfer is allowed when solving the model by CPLEX.
- **Aircraft deployment**: Each aircraft always deploys from its own hub at the beginning of the daily operation, and they are all required to return back to their hubs at the end of the daily operation.
- **Passenger delivery**: If the request is (partially) considered, all passengers in question should be transported successfully during the daily operation. No passenger should be spilt halfway during the journey.

- **Passenger spillage & recapture**: All passenger requests can be spilt if necessary, while each spilt passenger must be recaptured on the next day.
- **Transfer & turnaround time (TAT)**: In all cases, the reserved passenger transfer time and aircraft TAT equal to 1 hour.

3.3 Nomenclatures

The following parameters are defined to be used in the mathematical model in section 3 and Appendix B.

Sets:	
A	Set of arcs (i,j) within the map.
A^k	Set of arcs (i,j) compatible for vehicle k: $A^k =\backslash\{(i,j)\in A \mid i\in V^k$ and $j\in V^k\backslash\}\cup\backslash\{(i,h'^k)\in A \mid i\in V^k\backslash\}\cup\backslash\{(h'^k,j)\in A \mid j\in V^k\}$.
H	Set of hub airports: $H=\backslash\{h^k\in V^k \mid k\in K\ \}$
K	Set of vehicles.
R	Set of requests.
V	Set of airport nodes, each node i represents the location of airport i on the map.
V'	Set of airport nodes on the map by considering the revisit possibility: $V'=\backslash\{i_1,i_2 \mid i\in V\}$
V^k	Set of nodes compatible with vehicle k: $V^k=\backslash\{i\in V \mid rwy^k\le rwy_i\}$.
V'^k	Set of revisiting nodes compatible with vehicle k: $V'^k=\{i_1,i_2\in V' \mid rwy^k\le rwy_i, i\ne h^k\}\cup\{h^k,h'^k\}$.
V_{fuel}	Set of nodes and revisiting nodes with refuelling possibilities, including hubs.

Decision Variables	
q^{rk}_{ij}	Amount of passengers from requests r that are travelling from i to j on vehicle k.
$s^{k_1k_2}_{a_{i_{ab}}}$	Binary, 1 if the time vehicle k_2 arrives at node i_a is later than the time vehicle k_1 arrives at node i_b, $\forall a,b\in\{1,2\}$, 0 otherwise.
$s^{k_1k_2}_{d_{i_{ab}}}$	Binary, 1 if the time vehicle k_2 departures from node i_a is later than the time vehicle k_1 departures from node i_b, $\forall a,b\in\{1,2\}$, 0 otherwise.
$t^k_{i_a^-}$	Binary, 1 if passengers get on board aircraft k at node i_a via transfer, 0 otherwise.
$t^k_{i_a^+}$	Binary, 1 if passengers on aircraft k get off the aircraft at node i_a for transfer, 0 otherwise.
$u^k_{i_aj_b}$	Binary, $\forall a,b\in\{1,2\}$, 1 if arc (i_a,j_b) is traversed by vehicle k, 0 otherwise.
$v^k_{i_a}$	Distance travelled by vehicle k (since last refuelling) when arriving at node i for the a^{th} time.
$w^k_{a_{i_a}}$	Time of a^{th} arrival at node i by vehicle k.
$w^k_{d_{i_a}}$	Time of a^{th} departure at node i by vehicle k.
x^k_{ij}	Binary, 1 if arc (i,j) is traversed by vehicle k, 0 otherwise.
y^k	Binary, 1 if vehicle k is used in the final plan, 0 otherwise.
$z^{rk}_{p_a}$	Binary, 1 if vehicle k picks up part of the request r at its origin during its a^{th} visit, 0 otherwise.
$z^{rk}_{d_a}$	Binary, 1 if vehicle k delivers part of the request r at its destination during its a^{th} visit, 0 otherwise.

Parameters	
c^k	Cost for utilising vehicle k for an extra hour.
c^k_{ij}	Cost for vehicle k traversing arc (i,j).
d_{ij}	Distance between node i and node j.
D^r	Destination node of request r: $D^r\in V$.
h^k	Hub airport of vehicle k, $h^k\in V^k$.
h'^k	Hub airport of vehicle k as a starting and ending node.
mgh	Minimum guaranteed hours.
n	Number of airports.
O^r	Origin node of request r: $O^r\in V$.
q^r_n	Amount of passengers from request r on day n.
Q^k	Capacity of vehicle k.
ra^k	Range of vehicle k.
rwy^k	Minimum runway requirement for successful take-off or landing for vehicle k.
rwy_i	Runway length at node i.
s_i	Service time at node i.
$[t^r_{p_1},t^r_{p_2}]$	Time window for the pickup of request r.
$[t^r_{d_1},t^r_{d_2}]$	Time window for the delivery of request r.
t^k	Hours travelled of vehicle k until the previous day.
T^k_{ij}	Travel time of vehicle k on arc (i,j).
t^k_n	Accumulated time budget for vehicle k to operate at day n.
$t^k_{n_{tot}}$	Accumulated operational time of vehicle k at the beginning of day n.
$T^i_{transfer}$	Minimal passenger transfer time at airport i.
Δt	Required minimal time difference between landings or departures.
π_r	Penalty cost of spilling one passenger from request r.

3.4 Map structure

Traditionally, when solving the network scheduling problem, each airport are regarded as a node based on its location or its function (pick-up/delivery points). In most cases, the time dimension is not considered when the aircraft is visiting the airport. Due to this limitation, each aircraft can only visit the same airport at most once and the problem may miss a possible better solution.

One possible solution to consider the time dimension at each airport is to implement the time-space network, where each airport has its own time line. In this case, flight arcs and ground arcs are used to represent the aircraft movement and time flow when the aircraft is on ground. However, the time-space network is ideal for fleet assignment problem where the flights and time schedule have been determined.

For the UNHAS case, the flight schedule need to be decided along with the fleet assignment. Due to the adequate time slots at almost all airports considered, it is unnecessary to consider all the time point at each airport. The map structure for this model is considered to be a bit different than the time-space network model. As shown in the decision variables descriptions in section 3.3, every decision variable regarding to the node always consider which aircraft is operating at the same time. The network is observed from the aircraft's perspective and for every aircraft, the network is slightly different.

In order to form the mathematical model systematically, three nodes are assigned for every airport on the map. For an airport i on the map, node i represents the geographical location of the airport, regardless of the amount of times an aircraft k visits. When the time dimension is considered, node i_1 indicates the airport i if vehicle k arrives for the first time, and another node i_2 indicates the same airport when the same vehicle k revisits. The node i_2 is considered as a redundant node, an aircraft k could only visit i_2 if and only if it has visited the node i_1 before. This distinction is considered in order to distinguish the different states in the time dimension of the same airport. Each aircraft is able to visit the same airport at most twice and therefore the model can generate more complicated routes.

Based on this definition, when considering the flight arc between two airport A and B, four different combinations need to be considered. Each airport has two nodes to represent the same location but at different time: nodes A_1 and A_2 for airport A, and similarly, nodes B_1 and B_2 for airport B. Therefore the directional flight arc (A,B) can be represented as (A_1, B_1), (A_2, B_1), (A_2, B_1) and (A_2, B_2) when the time dimension is considered. However, when the flight arc from airport A to B is generally denoted without considering the revisit, the expression (A, B) is used. This difference can be noticed between decision variables $u^k_{i_a j_b}$ and x^k_{ij}.

In conclusion, to clarify the definition more clearly, for an airport A:

- A: node representing the geographical location of airport A.
- A_1: node representing the location of airport A, when an aircraft k visit airport A for the first time.
- A_2: node representing the location of airport A, when an aircraft k visit airport A for the second time.

However, if airport A is the hub of aircraft k, then h^k and h'^k are used to represent airport A. h^k and h'^k denote the same airport geographically, namely the hub of aircraft k. However, node h^k and h'^k are used to differentiate the same location at different time. It is defined that h'^k is regarded as the starting and ending node of the daily route of aircraft k. Therefore node h'^k is always visited first by aircraft k than node h^k in the flight route. Node A, A_1 and A_2 are therefore eliminated when considering the decision variables or constraints relevant to aircraft k in the time dimension at its hub.

3.5 Passenger Delivery Method

In order to stimulate the actual scenario of passenger delivery, three main forms are considered and implemented in the model:

- **Direct flight**: the passenger takes a non-stop flight from the origin to the destination.
- **Transit flight**: the passenger stays in the same aircraft when flying from the origin to the destination via a number of intermediate stops.
- **Transfer flight**: the passenger shifts to one or more connecting flights when flying from the origin to the destination.

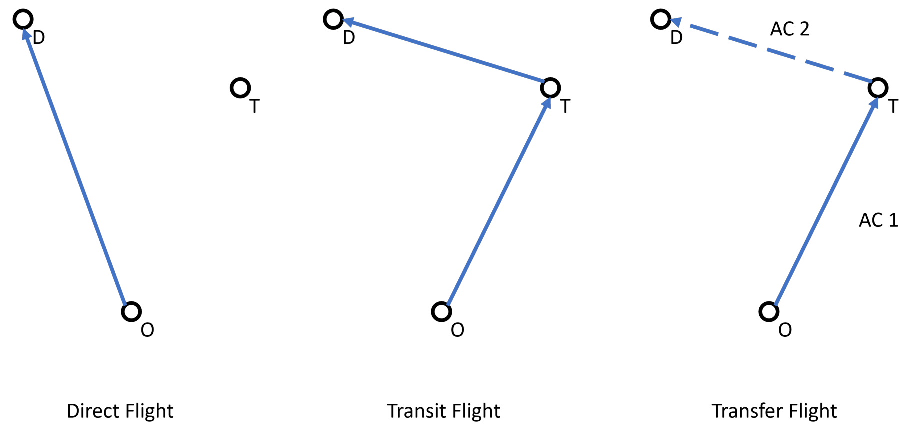


Figure 1: three ways of passenger delivery

When considering the transit flight option, maximal one transition is considered during the journey in order to simplify the planning strategy when forming the model as well as to decrease the computing time. However, more than one transition in the journey may occur in the actual result when the model is solved.

3.6 Relationship among decision variables

The general connection between all decision variables are illustrated in Figure 2, where the arrow shows the direction of the determination. As can be seen from the graph, y^k determines if the aircraft k is deployed or not in the daily operation. Therefore y^k is the base of the system and decides the existance of all other decision variables related to aircraft k.

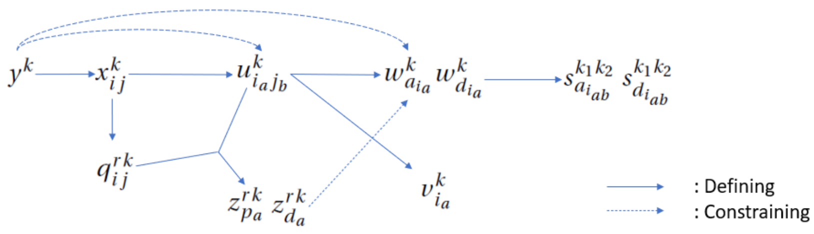


Figure 2: General connection between all decision variables.

Among all the decision variables, only two outputs are necessary for the plan: the value of all the x^k_{ij} for network scheduling and all the $w^k_{a_{i_a}}$ and $w^k_{d_{i_a}}$ that determine all the departure/arrival time of the vehicles at each airport. In order to have a deeper insight of all decision variables, the function of them are introduced as follows:

The most dominant decision variables above all is the deployment decision variable y^k, which determines if the aircraft k is deployed ($y^k = 1$) or not ($y^k = 0$) during the daily operation. It directly constrains the value of all the x^k_{ij}, which form the flight path during the daily operation.

In order to specify the flight path with more detail, the x^k_{ij} is extended to two more set of decision variables. By considering the aircraft capacity and number of passengers to be transported for a certain request, the number of passengers on each flight leg is determined by q^{rk}_{ij}. From the aircraft's perspective, when concerning the number of visit at an specific airport, a more explicit flight path can be described by $u^k_{i_a j_b}$ set. Furthermore, the pick-up/delivery occasions are presented by $z^{rk}_{p_a}$ and $z^{rk}_{d_a}$ set.

Moreover, to prevent violation of the aircraft range and maximum fuel capacity, decision variable set $v^k_{i_a}$ is created and refuel opportunities are also premeditated. Timetable of every aircraft arrival and departure can be generated by $w^k_{a_{i_a}}$ and $w^k_{d_{i_a}}$ set. The arrival and departure sequence can therefore be arranged by $s^{k_1k_2}_{a_{i_{ab}}}$ and $s^{k_1k_2}_{d_{i_{ab}}}$ set to space two arrival or departure with sufficient time based on the safety regulation.

3.7 Model Formation

To form the MILP model for the tabu search, the exact method model is first introduced in Appendix B and the tabu search strategy is implemented on top of it.

Usually, the number of daily requests is larger than the number of aircraft available during the day, which means that issuing requests to aircraft is more convenient. However, randomly matching the pick-up and delivery aircraft of the passengers for the same request results in massive combinations and numerous futile attempts.

In general, two sets of paths are essential for the problem-solving procedure: the paths of all aircraft and the paths of all individual cluster of passengers from the requests during the day. Therefore, in the preparation stage, the aircraft flying plan is considered as the base, and then the passenger flow is scheduled on top of it. All possible combinations of the flight paths form a smaller searching space with higher density of feasible solutions.

With this process, decision variable set $\{y^k\}$, $\{x^k_{ij}\}$, $\{u^k_{i_a j_b}\}$ and $\{v^k_{i_a}\}$ can all be determined based on the determination network from Figure 2. The remaining decision variable set $\{q^{rk}_{ij}\}$, $\{z^{rk}_{p_a}\}$, $\{z^{rk}_{d_a}\}$, $\{w^k_{a_{i_a}}\}$, $\{w^k_{d_{i_a}}\}$, $\{s^{k_1k_2}_{a_{i_{ab}}}\}$ and $\{s^{k_1k_2}_{d_{i_{ab}}}\}$ can preserve less elements, which can be determined in the following solving process. Consequently, The model scale shrinks significantly compared with the exact method approach. Moreover, in order to avoid the timeline error mentioned in section G.6.1, an additional set of decision variables $\{t^k_{i_a^-}\}$ and $\{t^k_{i_a^+}\}$ is introduced in the model.

3.8 Process Flow

The process and decision flow of the programme is illustrated in Appendix A, which provides an overview of the mathematical logic within the whole model.

3.9 Preparation

In the preparation stage, it is important to determine the domain where the tabu search algorithm is implemented. Unlike the traditional domain, which is continuous or distributed within a range of the n-dimensional space, the domain in question is formed by all possible flight plans the fleet can operate. Each flight plan in the domain is constructed by the flight schedule of all aircraft based on the assumption stated in section 3.2.

In order to define a minimal searching space, only the reasonable aircraft deployment plans are selected as candidates in the domain. Within each combination of aircraft deployment plan, all possible passenger travelling routes are generated based on the origins and destinations of the current requests. The preparation stage eliminates most of the combinations that are absolutely useless for the solution. This process saves a large proportion of storage as well as the solving time of the tabu search procedure.

3.10 Neighbourhood determination

In this method, a neighbour is defined as a flight plan that can be derived from the current flight plan by changing one (pair of) element within the current plan.

One of the most important step for the tabu search procedure is the determination of the neighbouring nodes for the next step. Unlike the node of the traditional n-dimensional space, which can easily determine the nearby node by shifting the node for one unit along one of the axes. The node in question is a flight plan that is formed by multiple aircraft's flight paths, therefore another way of determination is introduced.

Similar to the classical inter-route improvement heuristics, such as Relocate, Swap and 2-opt[Laporte, G., Ropke, S. and Vidal, T., 2014]. With the current aircraft flight plan, it is possible to generate a neighbouring plan by swapping, deleting, adding and replacing the intermediate stop or stops between the starting and final hub locations of one or more aircraft's route. When only one aircraft's route varies within the original flight plan, an inner-AC neighbour is generated. On the other hand, it is considered as an inter-AC neighbour when two aircraft's routes have been mutually modified.

The formation of inter-AC neighbours can be learnt from section 3.10.1 to section 3.10.3, while section 3.10.4 to section 3.10.7 illustrate the formation of inner-AC neighbours. All the inter-AC neighbours and inner-AC neighbours follow the requirements and assumptions for the aircraft's route formation.

3.10.1 Inter-AC Node Swap

Inter-AC swap is done by changing one stop of an aircraft's route into a different stop within another aircraft's route. It is only eligible if both aircraft are operating in the original flight plan.

$$\begin{matrix} AC\ 1: ...\ A\ B\ C\ D\ ... \\ AC\ 2: ...\ A'\ B'\ C'\ D'\ ... \end{matrix} \longrightarrow \begin{matrix} AC\ 1: ...\ A\ B'\ C\ D\ ... \\ AC\ 2: ...\ A'\ B\ C'\ D'\ ... \end{matrix} \tag{7}$$

3.10.2 Inter-AC Node Relocation

Inter-AC relocation extracts one stop of an aircraft's route and reissues it to another aircraft for operation, which can be inserted at any location in the route.

$$\begin{matrix} AC\ 1: ...\ A\ B\ C\ D\ ... \\ AC\ 2: ...\ A'\ B'\ C'\ ... \end{matrix} \longrightarrow \begin{matrix} AC\ 1: ...\ A\ B\ C\ ... \\ AC\ 2: ...\ A'\ B'\ C'\ D\ ... \end{matrix} \tag{8}$$

When an aircraft is only visiting one airport other than its hub, it is no more deployed after the relocation action has been done. On the other hand, if the receiving aircraft was not deployed beforehand, the starting and ending hubs are automatically supplemented to create a complete aircraft route.

$$\begin{matrix} AC\ 1: h^1\ A\ h^1 \\ AC\ 2: \text{(no deployment)} \end{matrix} \longrightarrow \begin{matrix} AC\ 1: \text{(no deployment)} \\ AC\ 2: h^2\ A\ h^2 \end{matrix} \tag{9}$$

3.10.3 Inter-AC Flight Swap

Inter-AC flight swap is done by switching the flight routes of two aircraft to form a new flight plan. However, this exchange of routes is only applicable for the following situation:

- Both aircraft are deploying from the same hub.
- At least one aircraft is deployed in the original flight plan.

The swap can be illustrated as follows ($h^1 = h^2$):

$$\begin{matrix} AC\ 1: h^1\ A\ ...\ h^1 \\ AC\ 2: h^2\ B\ ...\ h^2 \end{matrix} \longrightarrow \begin{matrix} AC\ 1: h^1\ B\ ...\ h^1 \\ AC\ 2: h^2\ A\ ...\ h^2 \end{matrix} \quad or \quad \begin{matrix} AC\ 1: h^1\ A\ ...\ h^1 \\ AC\ 2: \text{(no deployment)} \end{matrix} \longrightarrow \begin{matrix} AC\ 1: \text{(no deployment)} \\ AC\ 2: h^2\ A\ ...\ h^2 \end{matrix} \tag{10}$$

3.10.4 Inner-AC Swap

Inner-AC swap is done by swapping the sequence of two nearby stops at a random location within an aircraft's route, which can be illustrated as follows:

$$... \ A \ B \ C \ D \ ... \quad \longrightarrow \quad ... \ A \ B \leftrightarrow C \ D \ ... \quad \longrightarrow \quad ... \ A \ C \ B \ D \ ... \tag{11}$$

However, due to the fact that at least two stops are required between the hubs at two ends, therefore swapping method is not applicable for a route where the aircraft is not deployed or only visits one airport other than the hub during the operation.

3.10.5 Inner-AC Elimination

Elimination is done by removing a stop at a random place of an aircraft's route, which is illustrated as follows:

$$... \ A \ B \ C \ D \ ... \quad \longrightarrow \quad ... \ A \ B \ D \ ... \tag{12}$$

Similar to the applicable condition of swapping, deleting method is not considered for a route where the aircraft is not deployed. For an aircraft that only visits one airport other than the hub during the operation, deleting method results in cancellation of the aircraft deployment.

$$Hub \ A \ Hub \quad \longrightarrow \quad \text{(no deployment)} \tag{13}$$

3.10.6 Inner-AC Addition

Addition is done by inserting an additional stop at a random place of an aircraft's route, between the hubs at two ends, as shown below:

$$... \ A \ B \ C \ D \ ... \quad \longrightarrow \quad ... \ A \ B \ \_\_ \ C \ D \ ... \quad \longrightarrow \quad ... \ A \ B \ E \ C \ D \ ... \tag{14}$$

Exception occurs when the aircraft is initially not deployed, then not only a random stop is inserted, but also two hubs at both ends of the daily operation are attached simultaneously. This is exactly the opposite procedure as illustration shown in illustration 13:

$$\text{(no deployment)} \quad \longrightarrow \quad Hub \ A \ Hub \tag{15}$$

On the other hand, in order to improve efficiency and reduce computing unnecessary situations, when there is already an existing flight plan that can fulfil all the requests with no passenger spillage, the inner-AC addition method is not further considered in the following steps. This is because that in this circumstance, visiting an additional airport in the flight plan would only raise the total operational cost with no extra passenger delivery.

3.10.7 Inner-AC Substitution

Substitution is done by replacing an intermediate stop to another airport, at a random place of an aircraft's route between the hubs at two ends, as shown below:

$$... \ A \ B \ C \ D \ ... \quad \longrightarrow \quad ... \ A \ E \ C \ D \ ... \tag{16}$$

Replacement method is not considered if the aircraft in question is not deployed.

3.11 Evaluation

After obtaining a neighbour from the variation method, the next step is to determine roughly if the new node is reasonable and valuable to be processed for the following steps based on few requirements and constraints for the operation. This step filtered out many useless candidates and preserve less nodes to be processed in the following steps and save time.

The rough evaluation of the new flight plan consists two parts: Feasibility of the plan and the possibility of passenger delivery.

3.11.1 Feasibility check

The feasibility check assesses if the new flight plan is feasible for the aircraft operation in reality. Since that only the routes of one or two aircraft are rearranged compared with the original flight plan, it is sufficient to only examine the feasibility of these aircraft's route in the flight plan in question. The following aspects are considered:

- **Aircraft range limitation**: This check is done by calculating the aircraft accumulated distance since the last refuelling location by following the flight route, which is the same strategy as constraint 85u to 85x. If the accumulated distance exceeds the maximum range of the aircraft before reaching another refuelling location, then this route is eliminated. Otherwise, it is kept for other assessments.
- **Assumption on aircraft operation**: Regarding the assumption listed in section 3.2, each aircraft is allowed to visit any airport besides its own hub maximum twice, and passes its hub at most once in the middle of the operation. It is prohibited for the aircraft travelling the same flight arc more than once during the day, except that an arc connects the hub with another airport. Moreover, the condition where two identical stops are visited sequentially is considered impossible. If the new flight plan violates one of these rules, then this neighbour is eliminated. Otherwise, the flight plan is kept for further assessments.

3.11.2 Passenger delivery possibility

Normally, when an aircraft is deployed, it should be utilised functionally. The flight plan should achieve delivery of at least one passenger from any request, otherwise no aircraft deployment is necessary. Therefore, it is recommended to analyse whether the current flight plan is possible to form a delivery route for a request.

The underlying method is to first split or intercept part of an aircraft's flight path respectively, and then recombine different parts of the path from different aircraft together to simulate passenger transfer in reality. If there is at least one recombination or original flight path segment that can deliver one or more passengers from any requests, then this new neighbour is considered to be valuable for further procedure.

3.12 Neighbour selection strategy

When the neighbourhood of the current solution has been generated, it is essential to design the determination strategy when evaluating these candidates. The most common strategy is to examine all the candidates in the neighbourhood and determine the next incumbent solution based on a certain criteria, which ensures that the next incumbent solution is considered to be the best solution among all neighbours. However, this method consumes a lot of computing time, which results in a slow evaluating process and a lot of redundant situations have been processed.

During the searching process, there is always a trade-off between the accuracy and the computational time. In order to accelerate the process, another possible strategy is to stop the neighbourhood evaluation when a good enough candidate has been found. This process may result in a hastier and rougher assessment in every step, but the overall Pareto front shape can still be generated with less computational time.

On the other hand, an accurate and complete Pareto front is not necessary, as only the segment near zero passenger spillage is valuable for further analysis. In general, the part that has high passenger spillage is not considered at all and more focus and efforts should be put on the part that has low passenger spillage. Therefore, rough evaluation occurs when there is low passenger delivered and a more detailed and careful searching is taken place when large portion of the passenger requests have been satisfied.

During the creation of the Pareto front, not all solutions on the front are generated from the MILP model. Ideally, the Pareto front is assembled by two sets of solutions:

- **Decisive solutions**: Pareto optimal solutions which have been generated from the MILP model results of the neighbours in every step. They form the overall shape of the Pareto front.
- **Supplemental solutions**: Pareto optimal solutions which have been derived from the nearby decisive solutions. They complement the blanks or update the previous Pareto optimal solutions between adjacent decisive solutions.

3.12.1 Pareto Front Formation

The Pareto Front is formed with two axes: number of passenger spillage on the horizontal axis (x) and the operational cost on the vertical axis (y). Considering all the requests of a particular day, the total number of passengers to be delivered is defined as $nPAX_{tot}$. When there is no aircraft deployed, all the passengers are spilt. Therefore the point $(nPAX_{tot}, 0)$ is always on the Pareto front, and it is also considered as the starting point for the tabu search process.

After a good enough solution has been calculated in the n^{th} step, denoted as $(nPAX_n, OpsCost_n)$, it is added on the Pareto front curve and the relevant supplemental solutions are filled automatically with respect to the solutions from the previous steps.

The formation process is designed to have two main parts:

- **Stem formation**: The overall shape of the Pareto front is formed with rapid selection strategy. The solution that has higher passenger delivery than all the existing solutions or the one that has lower operational cost of the maximum passenger delivery solution recorded are selected immediately.
- **Branch growth**: After the stem has been formed, each decisive solution is considered as the starting point and a more detailed and smaller-sized tabu search is taken place respectively. The individual tabu search ends when there is no more better solution than the existing ones on Pareto front can be generated.

These two steps are designed for an ideal formation process of the Pareto front. However, for the UNHAS scenario, it is time-consuming and unnecessary to form the Pareto front with full accuracy. To pursue less computing time while reserving maximum solution quality, the step of branch growth is only applied for the solution that has zero passenger spillage.

3.12.2 Neighbour minimisation

In order to examine as few neighbours as possible and save computational time, the ones that are obviously invaluable are eliminated immediately before the model formation. This applies to the aircraft's flight plan where the combination of its maximal expectation of passenger delivery and its operational cost is no better than the existing solutions on the Pareto front.

The operational cost can be calculated directly with the current flight plan, while the number of maximal deliverable passengers is related to the aircraft route in the flight plan. During the rough estimation, if both the origin and the destination of a request are presented in the flight plan, then this request can be potentially fulfilled. In this way, the actual operational capacity is exaggerated and the maximum passengers to be delivered is also overestimated. Therefore, If the combination of the operational cost and the overestimated maximal passenger delivery is not better than any existing solutions on the Pareto front, the slackness in question ensures that the actual solution from the MILP model will be no better than the estimation and can be directly eliminated from the neighbourhood.

3.13 Daily Operation Time Budget

The main goal of the operation budget is to limit and minimise the accumulated aircraft operational time during the month with respect to the minimal guaranteed hours (MGH).

The minimal guaranteed hours (MGH) is the minimal amount of time each aircraft can be utilised in total in the contract. It is considered as a sink cost and therefore all aircraft are preferably operating within the amount of time issued by the MGH during the whole month. Extra aircraft utilisation time is possible, but it will result in extra cost.

Balancing of the aircraft utilisation time is a dynamic process, which aims to encourage the aircraft that have been utilised less during the previous days to be deployed longer in the coming days compared to other aircraft, and vice versa.

Ideally, all aircraft are utilised evenly during the days and all aircraft utilisation time does not exceed the MGH at the end of the month. Based on this requirement, the following characteristics need to be considered when designing the weight of operational time in the objective function:

- On the first day of the month, all aircraft should be considered equally with respect to the operational time in the flight schedule.
- If two aircraft have utilised the same proportion of their own MGH, they should be considered equally with respect to the operational time in the flight schedule.
- The larger proportion of the MGH an aircraft has utilised, it should be more discouraged to be utilised in the future.

3.13.1 Methodology

The daily operational time budget is considered in the definition of the corresponding weights when evaluating the total time cost of all aircraft. It is considered as a soft constraint and aircraft are not forced to operate within their corresponding budgets. In order to arrange the operation time budget systematically, a grading method is introduced to assess the performance of current flight schedule, as shown in definition 17, where C_t

is defined as the time budget cost of this flight plan. The time budget cost equals to the weighted sum of the operational cost of all aircraft, which is a reference to decide which plan to choose in the neighbourhood. A lower grade represents a flight plan that utilises the aircraft more evenly based on the accumulated aircraft utilisation time $t^k_{n_{acc}}$ with respect to their MGHs in general.

$$C_t = \sum_{k \in K} \sum_{(i,j) \in A^k} C^k_{mgh} c^k_{ij} x^k_{ij} \tag{17}$$

C^k_{mgh} is the weight of each aircraft, which is determined by the remained utilisation time within the MGH limit. When the total utilisation time of the aircraft approaches the MGH limit, the aircraft should operate less, which results in the fact that the weight C^k_{mgh} is positively correlated with the accumulated utilisation time $t^k_{n_{acc}}$ of aircraft k during the previous days, as can be shown in expression 18.

$$\left(C^k_{mgh}\right)_n \propto t^k_{n_{acc}} \tag{18}$$

And the definition of the remained aircraft utilisation time $t^k_{n_{rem}}$ is:

$$t^k_{n_{rem}} = MGH^k - t^k_{n_{acc}} \tag{19}$$

It is not desired if the MGH limit of an aircraft is exceeded while another aircraft still have some time remained under its MGH. The value of weight C^k_{mgh} is small when the aircraft just start its operation to encourage its utilisation, and the weight should increase exponentially when its total utilisation time is approaching the MGH. Therefore, a exponential function is chosen. Regarding to a single aircraft, the remaining time $t^k_{n_{rem}}$ within the MGH should be therefore negatively proportional to C^k_{mgh}. The weight C^k_{mgh} is considered to be a function of $t^k_{n_{rem}}$, which results in the definition of C^k_{mgh} in Equation 20:

$$(C^k_{mgh})_n = f_k\left(t^k_{n_{rem}}\right) \tag{20}$$

On the other hand, the distribution strategy of the MGH among the days is also considered. It is preferred that the remaining utilisation time $t^k_{n_{rem}}$ is evenly distributed in the following days of the month. Due to the fact that the MGH may vary among different aircraft, a MGH normalisation factor $f^k{}_{norm}(t^k_{n_{tot}})$ is introduced on the remaining utilisation time of the aircraft. The definition of C^k_{mgh} is therefore defined as Equation 21:

$$\left(C^k_{mgh}\right)_n = f\left(f^k{}_{norm}\left(t^k_{n_{rem}}\right)\right) \tag{21}$$

The normalisation factor is designed to eliminate the effect of different MGHs for different aircraft. Ideally, the most convenient budget distribution is to split the MGH uniformly throughout the days, and each aircraft is treated equally among the days when they are utilised steadily towards their individual MGHs. If two aircraft utilised the same percentage of their own MGH at the beginning of the day, they should be weighted equally. Therefore, the normalised remaining utilisation time should be defined as the ratio of the remaining time with the corresponding MGH, as shown in Equation 22:

$$f^k{}_{norm}\left(t^k_{n_{rem}}\right) = \frac{t^k_{n_{rem}}}{MGH^k} \tag{22}$$

On the other hand, regarding two aircraft to be utilised under the ideal situation, if the time budget is distributed uniformly and therefore the accumulated utilisation time increases linearly among the days, the corresponding ratio of the weights C^k_{mgh} from different aircraft should stay the same to ensure both aircraft are weighted in the same way in each day. Considering a constant budget B^k for aircraft k, the ratio between the two aircraft k_1 and k_2 should be the same on two different days, as shown in Equation 23:

$$\begin{aligned} \frac{(C^{k_1}_{mgh})_n}{(C^{k_2}_{mgh})_n} &= \frac{(C^{k_1}_{mgh})_{n+1}}{(C^{k_2}_{mgh})_{n+1}} \\ \Rightarrow \frac{(C^{k_1}_{mgh})_n}{(C^{k_1}_{mgh})_{n+1}} &= \frac{(C^{k_2}_{mgh})_n}{(C^{k_2}_{mgh})_{n+1}} = constant, \quad \forall k_1, k_2 \in K \end{aligned} \tag{23}$$

This characteristic leads to an exponential function for the definition of $C^{k_1}_{mgh}$.

In conclusion, based on the characteristic mentioned above, the definition of C^k_{mgh} is derived in Equation 24:

$$\begin{aligned} (C^k_{mgh})_n =& f\left(f^k{}_{norm}\left(t^k_{n_{rem}}\right)\right) \\ &= f\left(\frac{t^k_{n_{rem}}}{MGH^k}\right) \\ &= c \cdot a^{\left(-\frac{t^k_{n_{rem}}}{MGH^k}\right)} \end{aligned} \tag{24}$$

Considering the designated boundary value at $t^k_{n_{rem}} = 0$ and $t^k_{n_{rem}} = MGH^k$:

$$(C^k_{mgh})_n = \begin{cases} f_k(0) = 0.1 \\ f_k(MGH^k) = 1 \end{cases}$$

The definition of weight factor C^k_{mgh} is defined as:

$$\begin{aligned} (C^k_{mgh})_n &= f_k(t^k_{n_{rem}}) \\ &= 10^{\left(-\frac{t^k_{n_{rem}}}{MGH^k}\right)} \\ &= 10^{\left(\frac{t^k_{n_{acc}}}{MGH^k}-1\right)} \end{aligned} \tag{25}$$

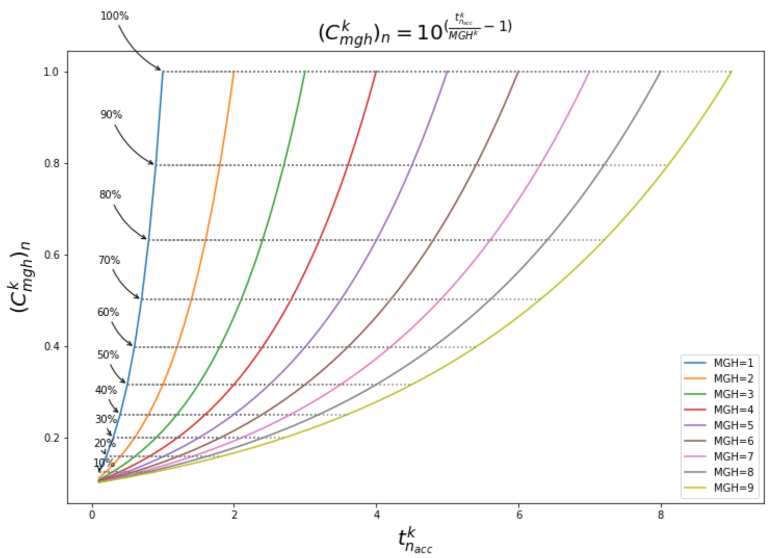


Figure 3: Graph of the C^k with different values of MGH

3.14 LP model formation

Regarding to a certain combination of aircraft flight plan, the first step is to "translate" the plan into the corresponding determined decision variables, which are regarded as constants. Then, corresponding undetermined decision variables, objective function and constraints are derived to construct the LP model for the CPLEX solver.

The size of the LP model for the tabu search method shrinks a lot compared with the exact method mentioned in Appendix B. There are less decision variables as well as constraints to consider, which can be found in section 3.15.

3.14.1 Formation of determined decision variables

Based on the flight plan, the values in decision variable set $\{y^k\}$, $\{x^k_{ij}\}$, $\{u^k_{i_a j_b}\}$ and $\{v^k_{i_a}\}$ can all be directly determined respectively. Among them, the decision variable set $\{v^k_{i_a}\}$ is not necessary to be determined, as the range limitation requirements is guaranteed as mentioned in section 3.11.1. These decision variables will not present in the LP model.

The determination process is simple, where the decision variables can be directly constructed from the aircraft route. When considering a sample aircraft route at a 3-airport map, the translation procedure can be seen from Table 2.

y_1	hub=1_2	2_1	3_1	1_1	3_2	2_2	1_2
x:		x^1_{42}	x^1_{23}	x^1_{31}	x^1_{13}	x^1_{32}	x^1_{24}
u:		$u^1_{1_2 2_1}$	$u^1_{2_1 3_1}$	$u^1_{3_1 1_1}$	$u^1_{1_1 3_2}$	$u^1_{3_2 2_2}$	$u^1_{2_2 1_2}$

Table 2: Example of translation process for determined DVs.

Consequently, the number of constraints as shown in Appendix B shrinks significantly. As a large number of DVs have been determined, many constraints that define these DVs are redundant. In addition, other constraints have less DVs to be determined by solving the model. In order to have a concise formation of the model, the sets of these determined DVs and parameters are defined for the convenience of model formation:

$$Y = \{y^k | y^k = 1, \forall k \in K\} \tag{26}$$
$$X = \{x_{ij}^k | x_{ij}^k = 1, \forall k \in K, \forall (i,j) \in A^k\} \tag{27}$$
$$U = \{u_{i_a j_b}^k | u_{i_a j_b}^k = 1, \forall k \in K, \forall (i,j) \in A^k, \forall a, b \in \{1,2\}\} \tag{28}$$
$$V'^k = \{i | x_{ij}^k \in X, \forall i \in V^k\} \tag{29}$$
$$A'^k = \{(i,j) | x_{ij}^k \in X, \forall (i,j) \in A^k\} \tag{30}$$
$$K' = \{k | y^k \in Y\} \tag{31}$$
$$J_X = \{(k,i,j) | x_{ij}^k \in X\} \tag{32}$$
$$J_U = \{(k, i_a, j_b) | u_{i_a j_b}^k \in U\} \tag{33}$$

3.14.2 Formation of undetermined decision variables

For decision variable set $\{q_{ij}^{rk}\}$, $\{w_{a_{i_a}}^k\}$, $\{w_{d_{i_a}}^k\}$, $\{s_{a_{i_{ab}}}^{k_1 k_2}\}$, $\{s_{d_{i_{ab}}}^{k_1 k_2}\}$, $\{z_{p_a}^{rk}\}$ and $\{z_{d_a}^{rk}\}$, individual volume also decrease accordingly. As shown in the relation map between DVs in Figure 2, only the decision variables that are corresponding to the determined decision variables are considered. Similarly, with the determined DVs example in Table 2 and two requests labelled as request 1 and 2, the generation of undetermined DVs is shown as Table 3:

y_1	**hub=** 1_2	2_1	3_1	1_1	3_2	2_2	1_2
x		x_{42}^1	x_{23}^1	x_{31}^1	x_{13}^1	x_{32}^1	x_{24}^1
u		$u_{1_2 2_1}^1$	$u_{2_1 3_1}^1$	$u_{3_1 1_1}^1$	$u_{1_1 3_2}^1$	$u_{3_2 2_2}^1$	$u_{2_2 1_2}^1$
q	r=1	$q_{42}^{1,1}$	$q_{23}^{1,1}$	$q_{31}^{1,1}$	$q_{13}^{1,1}$	$q_{32}^{1,1}$	$q_{24}^{1,1}$
	r=2	$q_{42}^{2,1}$	$q_{23}^{2,1}$	$q_{31}^{2,1}$	$q_{13}^{2,1}$	$q_{32}^{2,1}$	$q_{24}^{2,1}$

Table 3: Example of translation process for determined DVs.

The corresponding sets of undetermined DVs are defined as follows:

$$W_d = \{w_{d_{i_a}}^k | \forall u_{i_a j_b}^k \in U\} \tag{34}$$
$$W_a = \{w_{a_{j_b}}^k | \forall u_{i_a j_b}^k \in U\} \tag{35}$$
$$T^- = \{t_{i_a^-}^k | \forall u_{i_a j_b}^k \in U\} \tag{36}$$
$$T^+ = \{t_{i_a^+}^k | \forall u_{i_a j_b}^k \in U\} \tag{37}$$
$$Q^r = \{q_{ij}^{rk} | \forall (k,i,j) \in \{(k,i,j) | x_{ij}^k \in X\}\} \tag{38}$$
$$Q = \bigcup_{r \in R} Q^r \tag{39}$$
$$= \{q_{ij}^{rk} | \forall q_{ij}^{rk} \in Q^r, \forall r \in R\} \tag{40}$$
$$S_d = \{s_{d_{i_{ab}}}^{k_1 k_2}, s_{d_{i_{ba}}}^{k_2 k_1} | u_{i_a j_{a'}}^{k_1}, u_{i_b j_{b'}}^{k_2} \in U, k_1 \neq k_2\} \tag{41}$$
$$S_a = \{s_{a_{i_{ab}}}^{k_1 k_2}, s_{a_{i_{ba}}}^{k_2 k_1} | u_{j_{a'} i_a}^{k_1}, u_{j_{b'} i_b}^{k_2} \in U, k_1 \neq k_2\} \tag{42}$$
$$Z_P = \{z_{p_a}^{rk} | \forall r \in R, \forall u_{O_a^r j_b}^k \in U\} \tag{43}$$
$$Z_d = \{z_{d_b}^{rk} | \forall r \in R, \forall u_{i_a D_b^r}^k \in U\} \tag{44}$$

3.14.3 Mapping

Based on the definition of x_{ij}^k, $u_{i_a j_b}^k$ and q_{ij}^{rk}, they all represent characteristics related to the aircraft flight paths. Therefore, the elements in set Y, X and U are corresponding to each other with a certain type of mapping, which can be represented as:

$$f_{xu} : X \to U, \quad f_{xq}^r : X \to Q^r, \quad f_{uq}^r : U \to Q^r \tag{45}$$
$$f_{xq}^r = f_{uq}^r \circ f_{xu} \tag{46}$$

Only the pairing between X and U is bijection, while f^r_{xq} and f^r_{uq} are both injective non-surjective functions. All functions shown in Equation 45 are defined as:

$$u^k_{i_a j_b} = f_{xu}(x^k_{ij}) \quad \forall (k,i,j) \in J_X, \exists a,b \in \{1,2\} \tag{47}$$

$$q^{rk}_{ij} = f^r_{xq}(x^k_{ij}) \quad \forall (k,i,j) \in J_X, \forall r \in R \tag{48}$$

$$q^{rk}_{ij} = f^r_{uq}(u^k_{i_a j_b}) \quad \forall (k,i_a,j_b) \in J_U, \forall r \in R \tag{49}$$

3.15 LP Model with Tabu Search

Besides the number of DVs decreases, the main advantage of tabu search algorithm is that the scale of the model shrinks enormously. Most of the constraints related to flight route formation from Appendix B are removed and the rest are simplified with the determined DVs. Moreover, in order to avoid the timeline error mentioned in section G.6.1 and integrate the additional timeline checking method into the model, an additional set of decision variables $\{t^k_{i^-_a}\}$ and $\{t^k_{i^+_a}\}$ are introduced in the model.

$$min: \quad \sum_{k \in K'} c^k w^k_{a_{h'^k}} - \sum_{r \in R} \sum_{(k,i,j) \in J_X} q^{rk}_{ij} \tag{50}$$

$$s.t. \quad \begin{aligned} &\sum_{k \in K'} \sum_{j:(j,i) \in A'^k} q^{rk}_{ji} + \sum_{k \in K': i = h^k} \sum_{j:(j,i) \in A'^k} q^{rk}_{jh'^k} \\ &- \sum_{k \in K'} \sum_{j:(i,j) \in A'^k} q^{rk}_{ij} - \sum_{k \in K': i = h^k} \sum_{j:(i,j) \in A'^k} q^{rk}_{h'^k j} = 0 \end{aligned} \quad \begin{aligned} &\forall r \in R, i \neq \{O^r, D^r\}, \\ &\forall (k,i,j) \in J_X \end{aligned} \tag{51}$$

$$\sum_{\forall r \in R} q^{rk}_{ij} \leq Q^k \quad \forall (k,i,j) \in J_X \tag{52}$$

$$\sum_{k \in K'} \sum_{j:(O^r,j) \in A'^k} q^{rk}_{O^r j} + \sum_{k \in K': O^r = h^k} \sum_{j:(O^r,j) \in A'^k} q^{rk}_{h'^k j} \leq q^r_n \quad \forall r \in R, \forall (k,O^r,j) \in J_X \tag{53}$$

$$\sum_{k \in K'} \sum_{j:(j,O^r) \in A'^k} q^{rk}_{jO^r} + \sum_{k \in K': O^r = h^k} \sum_{j:(j,O^r) \in A'^k} q^{rk}_{jh'^k} = 0 \quad \forall r \in R, \forall (k,O^r,j) \in J_X \tag{54}$$

$$\begin{aligned} &\sum_{k \in K'} \sum_{j:(O^r,j) \in A'^k} q^{rk}_{O^r j} + \sum_{k \in K': O^r = h^k} \sum_{j:(O^r,j) \in A'^k} q^{rk}_{h'^k j} \\ &- \sum_{k \in K'} \sum_{i:(i,D^r) \in A'^k} q^{rk}_{iD^r} + \sum_{k \in K': D^r = h^k} \sum_{i:(i,D^r) \in A'^k} q^{rk}_{ih'^k} = 0 \end{aligned} \quad \forall r \in R, (k,i,D^r), (k,O^r,j) \in J_X \tag{55}$$

$$w^k_{a_{j_b}} - w^k_{d_{i_a}} \geq T^k_{ij} u^k_{i_a j_b} \quad \forall (k,i_a,j_b) \in J_U \tag{56}$$

$$w^k_{d_{i_a}} - w^k_{a_{i_a}} \geq s_i \quad \forall (k,i_a,j_b) \in J_U, i_a \neq h'^k \tag{57}$$

$$w^k_{d_{h'^k}} - w^k_{a_{h'^k}} \leq 0 \quad \forall k \in K' \tag{58}$$

$$\sum_{a \in \{1,2\}} z^{rk}_{p_a} - \sum_{j:(O^r,j) \in A'^k} q^{rk}_{O^r j} \leq 0 \quad \forall r \in R, \forall k \in K, \forall (O^r,j) \in A'^k \tag{59}$$

$$\sum_{a \in \{1,2\}} z^{rk}_{d_b} - \sum_{i:(i,D^r) \in A'^k} q^{rk}_{iD^r} \leq 0 \quad \forall r \in R, \forall k \in K, \forall (i,D^r) \in A'^k \tag{60}$$

$$z^{rk}_{p_a} - \frac{1}{Q^k} q^{rk}_{O^r j} u^k_{O^r_a j_b} \geq 0 \quad \begin{aligned} &\forall r \in R, \forall k \in K, \\ &\forall (O^r,j) \in A'^k, \forall a \in \{1,2\} \end{aligned} \tag{61}$$

$$z^{rk}_{d_b} - \frac{1}{Q^k} q^{rk}_{iD^r} u^k_{i_a D^r_b} \geq 0 \quad \begin{aligned} &\forall r \in R, \forall k \in K, \\ &\forall (i,D^r) \in A'^k, \forall b \in \{1,2\} \end{aligned} \tag{62}$$

$$w^k_{d_{O^r_a}} - t^r_{p_1} z^{rk}_{p_a} \geq 0 \quad \forall r \in R, \forall k \in K, \forall a \in \{1,2\} \tag{63}$$

$$w^k_{d_{O^r_a}} - (t^r_{p_2} - M) z^{rk}_{p_a} \leq M \quad \forall r \in R, \forall k \in K, \forall a \in \{1,2\} \tag{64}$$

$$w^k_{a_{D^r_b}} - t^r_{d_1} z^{rk}_{d_b} \geq 0 \quad \forall r \in R, \forall k \in K, \forall b \in \{1,2\} \tag{65}$$

$$w^k_{a_{D^r_b}} - (t^r_{d_2} - M) z^{rk}_{d_b} \leq M \quad \forall r \in R, \forall k \in K, \forall b \in \{1,2\} \tag{66}$$

$$s^{k_1 k_2}_{a_{i_{ab}}} + s^{k_2 k_1}_{a_{i_{ba}}} = 1 \quad \begin{aligned} &\forall k_1, k_2 \in K', \forall i \in V'^{k_1} \cap V'^{k_2}, \\ &\forall a,b \in \{1,2\} \end{aligned} \tag{67}$$

$$s^{k_1k_2}_{d_{i_{ab}}} + s^{k_2k_1}_{d_{i_{ba}}} = 1 \qquad \forall k_1, k_2 \in K', \forall i \in V'^{k_1} \cap V'^{k_2}, \forall a, b \in \{1, 2\} \tag{68}$$

$$w^{k_1}_{a_{i_a}} - w^{k_2}_{a_{i_b}} + M s^{k_1k_2}_{a_{i_{ab}}} \geq 0 \qquad \forall k_1, k_2 \in K', k_1 \neq k_2, \forall i \in V'^{k_1} \cap V'^{k_2}, \forall a, b \in \{1, 2\} \tag{69}$$

$$w^{k_1}_{d_{i_a}} - w^{k_2}_{d_{i_b}} + M s^{k_1k_2}_{d_{i_{ab}}} \geq 0 \qquad \forall k_1, k_2 \in K', k_1 \neq k_2, \forall i \in V'^{k_1} \cap V'^{k_2}, \forall a, b \in \{1, 2\} \tag{70}$$

$$w^{k_2}_{a_{i_b}} - w^{k_1}_{a_{i_a}} - (M + \Delta t) s^{k_1k_2}_{a_{i_{ab}}} \geq -M \qquad \forall k_1, k_2 \in K', \forall i \in V'^{k_1} \cap V'^{k_2} \forall a, b \in \{1, 2\} \tag{71}$$

$$w^{k_2}_{d_{i_b}} - w^{k_1}_{d_{i_a}} - (M + \Delta t) s^{k_1k_2}_{d_{i_{ab}}} \geq -M \qquad \forall k_1, k_2 \in K', \forall i \in V'^{k_1} \cap V'^{k_2} \forall a, b \in \{1, 2\} \tag{72}$$

$$t^{rk}_{i^-_a} \geq \frac{1}{Q^k} q^{rk}_{ij} \qquad \forall r \in R, \forall k \in K', \forall (i, j) \in A'^k \tag{73}$$

$$t^{rk}_{i^-_a} \leq q^{rk}_{ij} \qquad \forall r \in R, \forall k \in K', \forall (i, j) \in A'^k \tag{74}$$

$$t^{rk}_{j^+_b} \geq \frac{1}{Q^k} q^{rk}_{ij} \qquad \forall r \in R, \forall k \in K', \forall (i, j) \in A'^k \tag{75}$$

$$t^{rk}_{j^+_b} \leq q^{rk}_{ij} \qquad \forall r \in R, \forall k \in K', \forall (i, j) \in A'^k \tag{76}$$

$$t^{rk}_{h'^{k+}} + t^{rk}_{h'^{k-}} \leq 1 \qquad \forall r \in R, \forall k \in K' \tag{77}$$

$$w^{k_2}_{d_{i_{a'}}} - w^{k_1}_{a_{i_a}} - (M + T^i_{transfer})(t^{rk_2}_{i^-_{a'}} + t^{rk_1}_{i^+_a}) \geq -2M - T^i_{transfer} \qquad \forall (k_1, j_b, i_a), (k_2, i_{a'}, j_{b'}), \in J_U \; \forall r \in R, k_1 \neq k_2 \tag{78}$$

$$s^{k_1k_2}_{a_{i_{ab}}}, s^{k_1k_2}_{d_{i_{ab}}} \in \{0, 1\} \qquad \forall k_1, k_2 \in K', \forall i \in V'^{k_1} \cap V'^{k_2}, \forall a, b \in \{1, 2\}, k_1 \neq k_2 \tag{79}$$

$$q^{rk}_{ij} \in [0, Q^k] \qquad \forall k \in K', \forall r \in R, \forall (i, j) \in A'^k \tag{80}$$

$$w^k_{a_{i_a}}, w^k_{d_{i_a}} \geq 0 \qquad \forall k \in K', \forall i \in V'^k, \forall a \in \{1, 2\} \tag{81}$$

$$z^{rk}_{p_a}, z^{rk}_{d_a} \in \{0, 1\} \qquad \forall k \in K', \forall r \in R, \forall a \in \{1, 2\} \tag{82}$$

3.16 Solution selection

In order to move to the next incumbent solution and find the global optimum, the movement direction is determined by the performance of each candidate in the neighbourhood. A grading standard is essential to evaluate the flight plan of each surrounding neighbour.

Different from the goal of commercial airline services, where passenger pays a certain amount of money for the transport service and the airline company pursues maximal profit from the operation. The UNHAS is non-profitable and charge none or little amount of expenses from the passengers. Passenger spillage is possible with almost no consequential penalty to the airline and no inconvenience to the passenger. On the other hand, the UNHAS is financially supported by patrons from the UN, countries and NGOs all over the world, therefore the funds should be spent responsibly.

The main goal of the UNHAS mission is to minimise the passenger spillage (maximise the passenger delivery) and minimise the general operational cost at the same time. The two aspects are correlated to each other and therefore a trade-off is needed when comparing different solutions on the Pareto front. A convenient way for this procedure is to grade the performance of all solutions with certain function or algorithm, and then the best solution is the one that has the highest or lowest score.

Comparing the operational cost per passenger is considered to be a solution of the dilemma between minimisation of both passenger spillage and the operational cost. A smaller value of operational cost per capita is certainly desired during the daily operation, which implies that with an unit amount of cost, more passengers can be transferred.

However, it is difficult when comparing two solutions with the same operational cost per passenger. Although it is possible to choose the point where more passengers are transferred, it is sometimes not convincing where more passengers can be delivered with a little bit more operational cost per passenger.

The strategy then evolves further. From Figure 4, it can be seen that a range of passenger spillage correspond to the same cost level. On every cost level, it is only valuable to consider the point that has the minimal passenger spillage (red points), which is defined as "optimal cost solution". It is sufficient to examine all these points on different cost levels. When comparing the adjacent two optimal cost solutions (x_1, y_1) and (x_2, y_2) on two

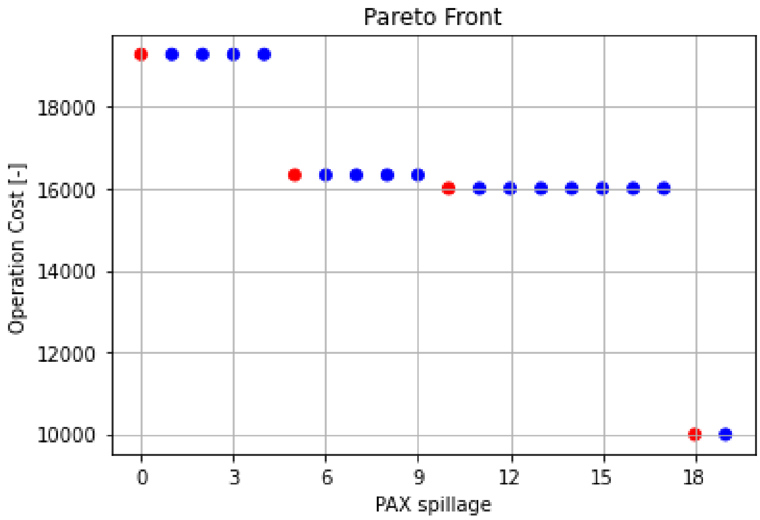


Figure 4: Example of a Pareto front

different cost levels, the slope represents the average cost saved per one more passenger spillage when shifting from one cost level to a lower one:

$$\text{slope} = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1} = \frac{\Delta OpsCost}{\Delta PAX} \tag{83}$$

Among all the segments of the Pareto front, the overall optimum is presented by the first steepest slope when viewing the Pareto front from zero-spillage solution. The increasing gradient shows that the particular cost increment could be used in the most efficient way for less passenger spillage. Therefore, the final optimal solution is determined to be the optimal cost solution on the lower cost level of the largest slope before its gradient drops along the direction of increasing spillage.

4 Verification and Validation

In this section, the verification and validation of the tabu search model are performed. The mathematical logic of the model as well as its result are checked during the verification, and the validation assesses if the result is reasonable and valuable. The verification of the model is done in section 4.1 and the model is further validated in section 4.2 by comparing the result with previous master student's work on the same topic.

The exact model has been verified and validated in Appendix G, which has been tested with multiple scenarios to check its feasibility and accuracy. This chapter will focus on the verification and validation of the tabu search model and algorithm.

4.1 Verification

The verification process focuses on the correctness of the logic throughout the model design, as well as the accuracy of the result from the optimisation model. A simplified scenario is created for the model to solve, and the verification is done by comparing the computational result from the model under this scenario with the manually computed or planned result. Therefore, the established scene should be simple enough while considering all the designated characteristics to ensure that the result can be obviously detected.

In Table 12, a simple request plan of 5 days has been created for the verification purpose. As can be noticed, the demand of each day stays the same in order to see the effect of the utilisation budget strategy by comparing the results of different days with the same requests.

The first day of the simplified scenario is solved based on the tabu search strategy and the designed model. The resultant Pareto front is shown in Figure 5 and the final flight plan chosen based on the largest gradient criteria in section 3.16 is shown in Table 4 and the remaining request spillage is summarised in Table 5.

The verification process consists three parts: verification of the algorithm, verification of the MILP model and verification of the utilisation budget distribution strategy.

4.1.1 Algorithm verification

Verification of the algorithm focuses on the logic of the solving process. After the decision variables (DVs) are generated, the following aspects are checked:

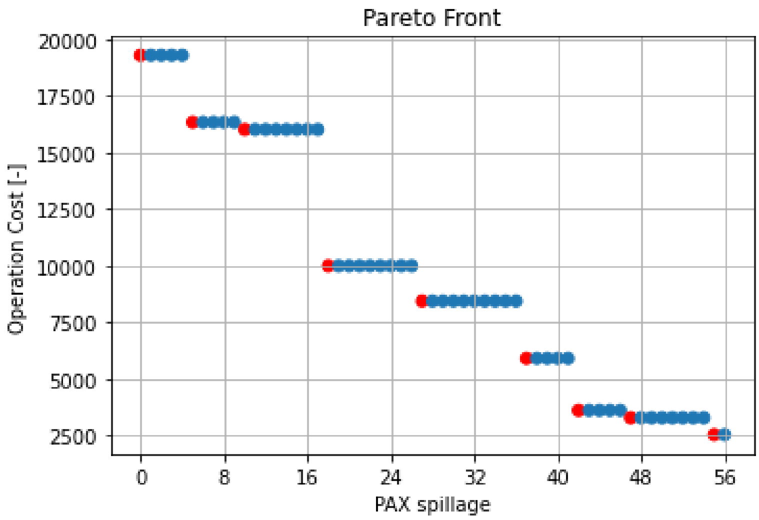


Figure 5: Pareto Front of simplified situation (day 1)

AC	From	Departure	To	Arrival	Request	PAX
Dash 8_1	JUB	00:00	WAU	00:44	0	0
Dash 8_1	WAU	01:44	RUB	02:07	1603	20
Dash 8_1	WAU	01:44	RUB	02:07	1624	2
Dash 8_1	RUB	03:07	JUB	03:54	1474	15
Dash 8_1	RUB	03:07	JUB	03:54	1603	20
Dash 8_1	RUB	03:07	JUB	03:54	1624	2
Dornier 228	JUB	04:54	RUM	05:18	1624	2
Dornier 228	RUM	06:18	JUB	06:42	1431	15

Table 4: Final flight plan of simplified situation (day 1)

Requst	From	To	Demand	Spillage
1474	RUB	JUB	15	0
1431	RUM	JUB	20	5
1603	WAU	JUB	20	0
1624	WAU	RUM	2	0

Table 5: Summary of the overall passenger spillage of simplified situation (day 1)

- **General code structure**: Check if the code has any grammar error or logical error.
- **Neighbourhood generation**: Check if all the neighbours are fully generated and each neighbour is derived from the flight plan from previous step based on the designated method. Moreover, check if all impossible, unnecessary and not valuable neighbours are filtered out.

4.1.2 Exact method model verification

The LP model developed with exact method in Appendix B is verified by creating few typical scenario tests. These scenario tests are collected in Appendix G, which proves that the exact method model is verified.

4.1.3 Tabu search model verification

After all the decision variables have been correctly generated, the MILP model can be formed with the decision variables. Regarding the formation and solution of the MILP model, following categories are checked:

- **Objective and constraints**: Check if the objective function and every constraint are constructed with the correct DVs and relations in the design.
- **Model formation**: Check if the objective function and all the constraints are correctly and fully programmed in the code based on the mathematical model.

- **Determined DVs**: Check if all the determined DVs are correct and fully generated based on the predetermined aircraft flight plan.
- **Undetermined DVs**: Check if the parameters of all undetermined DVs are correctly and fully assigned from the determined DVs. Check the decision variable types (e.g. continuous/integer/binary) are correctly issued.
- **LP solver result translation**: Check if the results from the LP solver are reasonable and are translated into literal schedule correctly.

4.1.4 Aircraft operation time budget strategy verification

The aircraft operation time budget strategy stated in section 3.13 is verified by solving the 5-day requests created in the simplified situation in Table 12 with the fleet stated in Table 13 of Appendix D. The utilisation time weight accumulation of each aircraft is shown in Figure 6.

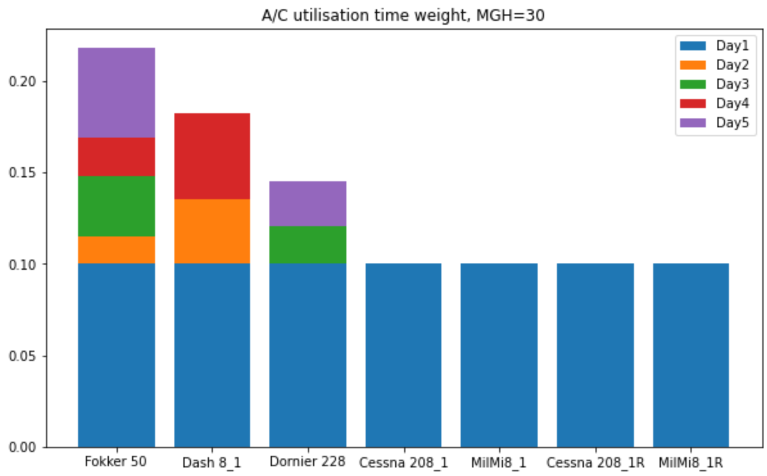


Figure 6: Aircraft utilisation time weight accumulation, simplified situation, allow spillage

It illustrates that only the first three aircraft are utilised during the period due to the small request volume and minimisation of aircraft operational cost. However, a rotational deployment can be noticed and the utilisation times of three aircraft are increasing steadily during the days. Therefore, the time budget strategy is functional to balance the operating time among aircraft.

4.2 Validation

In the tabu search method, due to the fact that the flight plan is pre-determined rather than generated by the MILP model solution, the validity of the aircraft deployment and flight paths are guaranteed. The flight paths should follow all the geometrical and technical assumptions stated in section 3.2, and the flight route of each aircraft forms a closed loop.

More focuses are put on the validation of the time schedule, the passenger flow and transfer strategy. Due to the fact that the UNHAS operation have been solved by previous graduates with different methods and validated by professional planners[Niemansburg, S.P., 2019], the validation of these aspects for this algorithm can be done by comparing the respective results with the ones from the previous master students' work. Based on the assumption and requirements designed for this method, the algorithm should provide a better result that saves operational cost. The points on the resultant Pareto front at low passenger spillage section should be not worse than those from previous student's design.

As can be seen from the result comparison in Figure 7, the tabu search model generates reasonable result. For the same passenger spillage, the tabu search model could find a solution with lower operational cost. When taking the final chosen flight plan from Niemansburg for request on 13/04/2015 (Table 44 of Appendix H) into the tabu search model, the same operational cost and passenger spillage is generated. Therefore the cost evaluation method the network scheduling of the tabu search method is validated.

On the other hand, as can be seen from Table 7, the results on Pareto front from the tabu search model save from 1.42% to 11.57% of the operational cost compared with the method from Niemansburg.

By solving the same problem with the same passenger spillage as Niemansburg's final flight plan in Appendix H, the tabu search model result is generated in Table 45. By comparing the two flight plans in Table 6 and the illustration in Figure 8, the following points can be noticed:

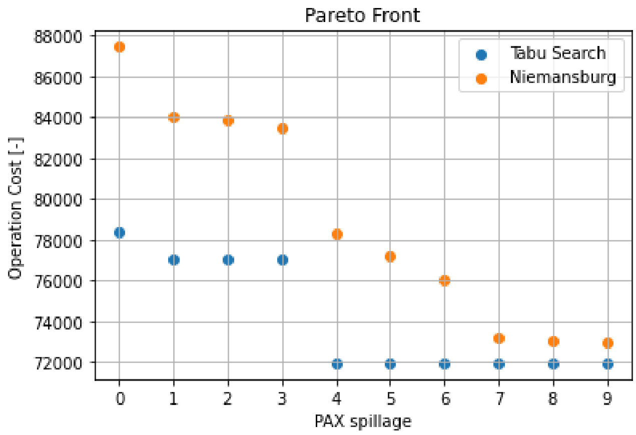


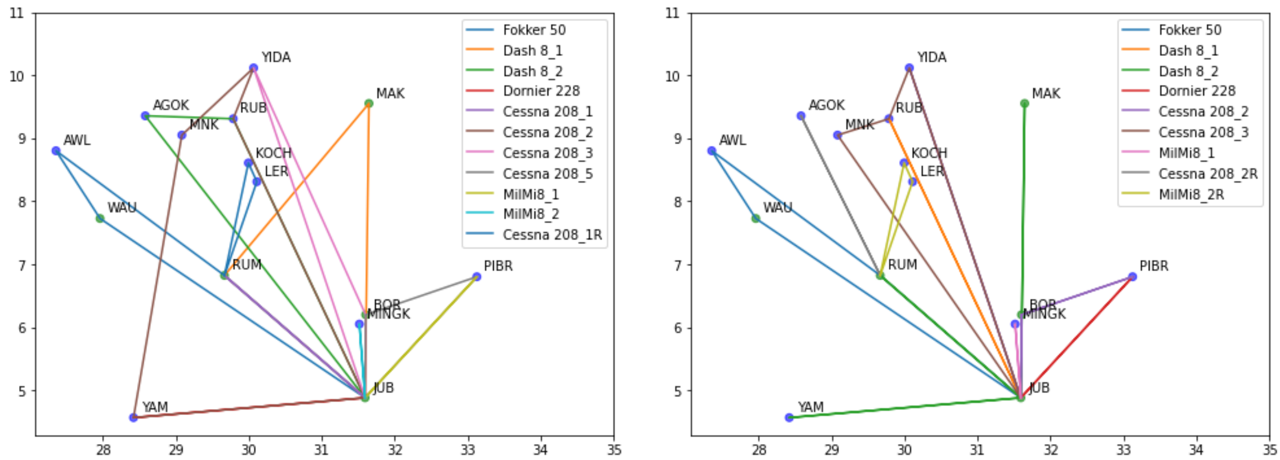
Figure 7: Pareto front comparison with result from Niemansburg, for requests on 13/04/2015

	Niemansburg	**TS Model**
Fokker 50	JUB, WAU, AWL, RUM, JUB	JUB, WAU, AWL, RUM, JUB
Dash 8_1	JUB, RUB, JUB	JUB, MAK, RUM, JUB
Dash 8_2	JUB, RUM, JUB, MAK, JUB, YAM, JUB	JUB, AGOK, RUB, JUB
Dornier 228	JUB, PIBR, JUB	JUB, YAM, JUB
Cessna 208_1		JUB, RUM, JUB
Cessna 208_2	JUB, BOR, PIBR, BOR, JUB, YIDA, JUB	JUB, RUB, YIDA, MNK, YAM, JUB
Cessna 208_3	JUB, YIDA, RUB, MNK, JUB	JUB, YIDA, BOR, JUB
Cessna 208_4		
Cessna 208_5		JUB, BOR, PIBR, JUB
MilMi8_1	JUB, MINGK, JUB	JUB, PIBR, JUB
MilMi8_2		JUB, MINGK, JUB
Cessna 208_1R		RUM, KOCH, LER, RUM
Cessna 208_2R	RUM, AGOK, RUM	
MilMi8_1R		
MilMi8_2R	RUM, KOCH, LER, RUM	

Table 6: Flight plan comparison with result from Niemansburg, for requests on 13/04/2015

- Fokker 50 is deployed with the same route in both plans.
- Cessna 208_4 and MilMi8_1R are not deployed in both solutions.
- The route operated by MilMi8_1 in Niemansburg's result is flown by Cessna MilMi8_2 in tabu search model result.
- The route operated by MilMi8_2R in Niemansburg's result is flown by Cessna 208_1R in tabu search model result.
- Less aircraft from RUM are deployed by tabu search model compared with Niemansburg's model.
- Passenger travelling from JUB to AGOK is transported directly in TS model, while this request is done by transferring at RUM in Niemansburg's model.
- The request from JUB to YAM has been split and operated by two smaller aircraft in TS model.
- An aircraft is particularly deployed for the request from JUB to MAK in Niemansburg's model, while the same request is fulfilled with other request by one aircraft by the tabu search model.
- Niemansburg's model depends on RUM and JUB heavily as hubs to transfer the passengers between different designated regions, while the aircraft in the tabu search model have more freedom to fly.

Moreover, the utilisation time difference is summarised in Table 46 of Appendix H. As can be seen from the comparison, the tabu search result deploys more aircraft in the fleet with longer utilisation time in total. Comparing with Niemansburg's result, the average aircraft utilisation time of tabu search model result is shorter and the aircraft are used more evenly (lower standard deviation), which complies with the expectation of the utilisation time budget method in section 3.13.



(a) Tabu search model (b) Niemansburg's model

Figure 8: Final flight plan comparison on 13/04/2015

Passenger Spillage	**0**	**1**	**2**	**3**	**4**
Niemansburg	87470	84000	83916	83448	78309
Tabu Search	78401.2	77048.1	77048.1	77048.1	71954.1
Difference	-11.57%	-9.02%	-8.91%	-8.31%	-8.83%

Passenger Spillage	**5**	**6**	**7**	**8**	**9**
Niemansburg	77223	76047	73185	73034	72976
Tabu Search	71954.1	71954.1	71954.1	71954.1	71954.1
Difference	-7.32%	-5.69%	-1.71%	-1.50%	-1.42%

Table 7: Pareto front result comparison with Niemansburg, for requests on 13/04/2015

4.2.1 Validation of time schedule

Time schedule can be validated on the following aspects:

- **Flight time**: The duration of the flight on each leg is corresponding to the aircraft speed and the distance.
- **Turn around time**: Sufficient time is reserved between the last landing and the next departure for every aircraft.
- **Minimal time gap between departures/landings**: Certain length of time is reserved between every two departures or landings for safety requirements.
- **Transfer flight time**: Every connecting flight should not leave earlier than the arrival of the designated inbound flight.

4.2.2 Validation of passenger flow

- **Full/Partial completion of request**: No PAX left halfway, all passengers are either delivered to the destination or not delivered.
- **Aircraft capacity**: The passenger onboard should be no more than the number of seats on the aircraft.
- **Passenger conservation**: The passenger flow is continuous and conservative, where there is no passenger (dis)appear during the whole operation.

As can be seen from Appendix H, the tabu search method result in Table 45 is compared with previously validated result from Niemansburg's in Table 44 under the same daily problem with the same amount of overall passenger spillage, all the above-mentioned aspects are validated.

5 Conclusions

The tabu search approach of the UNHAS network optimisation model has been designed, verified and validated. The results have been generated and conclusions can be therefore yielded based on the performance. In this chapter, the applicability and insufficiency of the model will be described first, which answers the main research question. Recommendations for future development of the model will be also provided for further research based on the current experience and study.

5.1 Applicability

The main objective of this project is to introduce the tabu search method as the heuristic to solve the UNHAS optimisation problem, as well as to split the requests and consider past and future demands. Moreover, based on the comparison in section 4.2, the model is able to generate more cost-efficient solutions than the previously developed method.

On the other hand, the model is designed with high flexibility by considering the most general situation possible under all the requirements. Therefore, it is possible to modify the model to fit different situations by adding constraints or designating certain decision variables. Due to the high flexibility, the following aspects are realised or can be considered with the model.

5.1.1 Request Division

Due to the fact that a single request is not specifically assigned to a certain aircraft, large flexibility exists when splitting the request into a number of aircraft. Each aircraft can take any portion of the request and the solver is deciding how a single request is distributed among all aircraft to obtain an overall optimal solution.

Moreover, it is also possible to assign certain aircraft to delivery a certain amount of passengers from a specific request by adding corresponding constraints in the model. This characteristic offers the planner a lot of freedom to apply necessary manual interference to the solving process.

5.1.2 Passenger Transfer

The most important feature added to the model is the possibility of passenger transfers at any airport. It is the key that helps to maximise the utilisation of aircraft capacity and improve to decrease the overall operational cost with the same amount of passenger delivery.

Similarly, interference of the model is possible by assigning certain airports to be transfer hubs. It can be realised by changing a few constraints and decision variables in the model.

5.1.3 Consideration of Minimum Guaranteed Hours

The goal of considering the MGH is to utilise the aircraft within the limit sufficiently and to save the overall cost of the operation. With the weighting function, it is possible to change the deployment priorities of all aircraft in a dynamic way, therefore every aircraft can be used evenly based on their individual MGHs to avoid exceeded aircraft utilisation time.

5.1.4 Past and Future Demand Consideration

The model managed to consider the recapture of previous passenger spillage within one day, and the spillage itself is generated based on known or anticipated future requests from the next day. Due to the lack of demand frequency information, it is hard to anticipate the passenger requests in the coming days. However, the possibility of recapturing the spilt passenger within one day ensures last-minute requests can be considered in the plan the next day, and the current computing time of the model also guarantees that all requests that are submitted 24 hours before the deployment can be considered.

5.2 Model Limitations

Despite multiple new features and considerations of the real-life scenarios, there are also a few downsides of the model that cannot be ignored due to technical limitations and the model's own characteristics.

5.2.1 Computing time

One of the main difficulties of the current model is the large computing time, which is an unavoidable issue to be concerned. The tabu search approach simplifies the problem scale and shrinks the number of decision variables and constraints. It makes the problem easier for the solver to solve by scaling down the size of the matrix. To reach the same goal, the trade-off takes place between the complexity of each problem and the number of problems to be solved in general. As the size of a single problem is minimised in the design, this approach results in an enormously large number of neighbourhood assessments and multiple rounds of LP model formation.

5.2.2 Sub-problem division

The UNHAS network optimisation problem in question does not have optimal substructure[Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C., 2009]. The daily requests for the UNHAS network optimisation are considered integrally when forming the mathematical model. When considering a scenario that contains part of the requests, it is not guaranteed that the model could derive the optimal solution for the sub-problem from the overall optimal solution.

5.2.3 Demand anticipation

Due to the lack of passenger behaviour and more information about their schedule, the current design is impossible to make an accurate prediction of the passenger demand and plan the coming days beforehand.

5.2.4 Passenger spillage

As the compensation of the passenger spillage is not able to be quantified, it is not possible to evaluate if the passenger-spilling solution on the Pareto front is worthwhile. The current selecting strategy stated in section 3.16 compares the average marginal cost relative to decide the final solution. A reference should be introduced to compare the average marginal cost and the non-spillage solution can be considered as well.

5.3 Recommendations for Future Development and Research

The performance of the model has been discussed and the limitations have been reflected.

The following recommendations are generated for future research and development of the topic:

- The probability distribution of passenger demand between different O&Ds can be studied in order to design a sound strategy for request anticipation.
- The resultant compensation by passenger spillage can be studied. This could measure the penalty of every spilt passenger and the trade-off between operational cost and the spillage can be initiated in a more quantitative way.
- The model can be improved or redesigned to reduce the number of decision variables and constraints. A smaller problem size makes it easier for the solver to solve and accelerate the overall solving process.
- A machine learning approach can be implemented to anticipate future demands.

References

[cpl, 2021] (2021). ILOG CPLEX Optimization Studio.

[ICR, 2021] (2021). International Committee of the Red Cross.

[wfp, 2021] (2021). United Nations World Food Programme (WFP).

[Abara, J., 1989] Abara, J. (1989). Applying integer linear programming to the fleet assignment problem. *Interfaces*, 19(4):2028.

[Billet, T., 2021] Billet, T. (2021). Humanitarian Fleet Planning and Weekly Scheduling Optimisation - Thesis Report.

[Bräysy, O. and Gendreau, M., 2005a] Bräysy, O. and Gendreau, M. (2005a). Vehicle routing problem with time windows, part i: Route construction and local search algorithms. *Transportation Science*, 39(1):104118.

[Bräysy, O. and Gendreau, M., 2005b] Bräysy, O. and Gendreau, M. (2005b). Vehicle routing problem with time windows, part ii: Metaheuristics. *Transportation Science*, 39(1):119139.

[Chen, P., Huang, H-K. and Dong, X-Y., 2010] Chen, P., Huang, H-K. and Dong, X-Y. (2010). Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem. *Expert Systems with Applications*, 37(2):16201627.

[Coello, C. C., Lamont, G. B. and Van Veldhuizen, D. A., 2002] Coello, C. C., Lamont, G. B. and Van Veldhuizen, D. A. (2002). *Evolutionary algorithms for solving multi-objective problems*. Springer.

[Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C., 2009] Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C. (2009). *Introduction to algorithms.* Mit Press.

[Dorn, A. W., 2014] Dorn, A. W. (2014). *Air power in UN operations: wings for peace.* Ashgate.

[Eftekhar, M., Masini, A., Robotis, A. and Van Wassenhove, L. N., 2013] Eftekhar, M., Masini, A., Robotis, A. and Van Wassenhove, L. N. (2013). Vehicle procurement policy for humanitarian development programs. *Production and Operations Management*, 23(6):951964.

[Flood, M. M., 1956] Flood, M. M. (1956). The traveling-salesman problem. *Operations Research*, 4(1):6175.

[Garey, M. R. and Johnson, D. S., 1990] Garey, M. R. and Johnson, D. S. (1990). *Computers and Intractability; A Guide to the Theory of NP-Completeness.* W. H. Freeman Co., USA.

[Glover, F., 1986] Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers Operations Research*, 13(5):533549.

[Hillier, F.S. and Lieberman, G.J., 2014] Hillier, F.S. and Lieberman, G.J. (2014). *Introduction to Operations Research.* Mcgraw-Hill Education.

[Laporte, G., Ropke, S. and Vidal, T., 2014] Laporte, G., Ropke, S. and Vidal, T. (2014). The family of vehicle routing problems. In *Vehicle routing: problems, methods, and applications (2nd Edition)*, chapter 4, pages 1–33.

[Liu, M., Cao, J., Liang, J. and Chen, M., 2020] Liu, M., Cao, J., Liang, J. and Chen, M. (2020). *Epidemic-logistics modeling: a new perspective on operations research.* Science Press.

[Mekking, Y.C., 2020] Mekking, Y.C. (2020). A Novel Formulation of the Vehicle Routing Problem for Humanitarian Applications.

[Nash, J. C., 2000] Nash, J. C. (2000). The (dantzig) simplex method for linear programming. *Computing in Science Engineering*, 2(1):2931.

[Nash, S.G., 1990] Nash, S.G. (1990). *A History of Scientific Computing.* ACM.

[Niemansburg, S.P., 2019] Niemansburg, S.P. (2019). Humanitarian Flight Optimization - Thesis Report.

[Osman, I. H., 1993] Osman, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41(4):421451.

[Pearl, J, 1984] Pearl, J (1984). *Heuristics: Intelligent search strategies for computer problem solving.* Addison-Wesley Pub. Co., Inc.

[Toth, P. and Vigo, D., 2015] Toth, P. and Vigo, D. (2015). *Vehicle routing: problems, methods, and applications (2nd Edition).* Society for Industrial and Applied Mathematics.

[Vikhar, P. A., 2016] Vikhar, P. A. (2016). Evolutionary algorithms: A critical review and its future prospects. *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, page 261265.

Appendices

A Process Flow

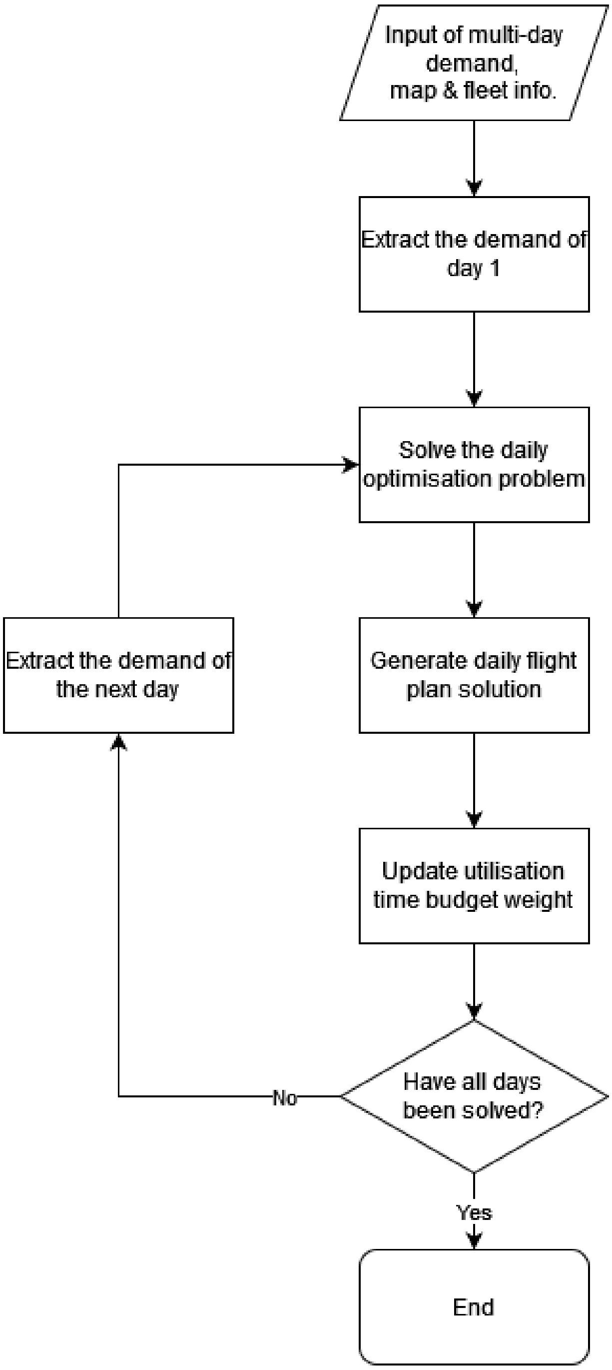


Figure 9: Process flow of multiple days' problem.

B MILP formation

In order to develop the MILP model for the tabu search method, the mathematical model for exact method is first designed as the base for modification.

Considering the requests of a single day, the objective function is given in polynomial 84, which consists three parts: the operational cost of all the flights during the day, the daily budget of aircraft utilisation, and the penalty for passenger spillage. This objective function intend to minimise the operational cost, balance the aircraft utilisation as well as maximise the demand implementation. The relevant constraints and boundaries of decision variables are shown from 85a to 85bm.

Constraint 85a ensures the continuity of vehicle flow at each node. Constraint 85b determines if the aircraft k is allowed to be deployed from its hub. Constraint 85c guarantees the continuity of passenger flow for request r at nodes other than the origin or the destination of the request. Constraint 85d reveals the capacity verification on each flight. Constraint 85e and 85f are the demand verification at pickup node of each request and avoids back flow of passengers. Constraint 85g ensures that all passengers in the request are delivered to the destination. Constraint 85h and 85i ensures the uniqueness of decision variable $u^k_{i_a j_b}$ on every flight arc. Constraint 85j

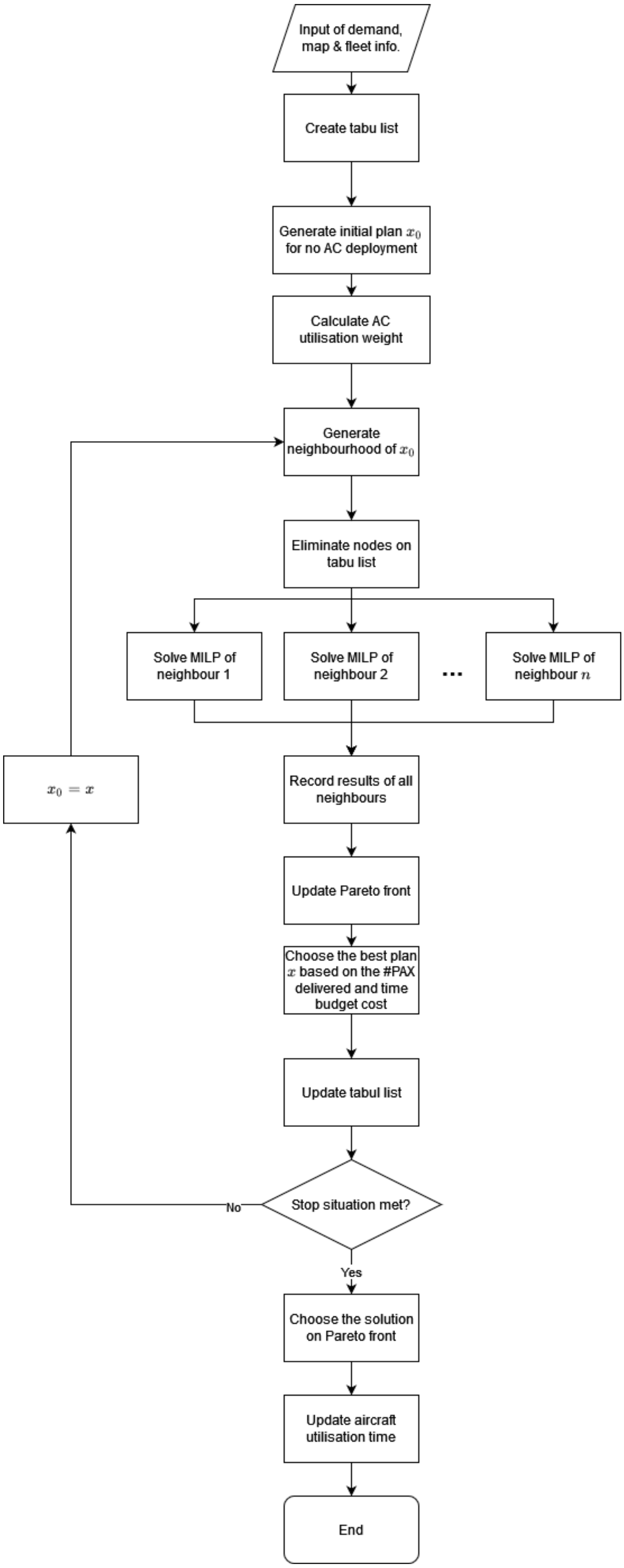


Figure 10: Process flow of the algorithm when solving the daily problem.

to 85n ensures the consistency between x_{ij}^k and $u_{i_a j_b}^k$ on every flight arc, where constraint 85l, 85n ensures the reasonable arrangement of the location-time nodes to decision variables x_{ij}^k and $u_{i_a j_b}^k$. 85o and 85p are the continuity constraints of inbound and outbound aircraft at the node in time dimension. Constraints 85q ensures aircraft to visit its hub halfway through the operation. Constraint 85r to 85s ensures the aircraft to always visit the first time node of the next stop after its deployment from the hub.

Range limitation of the aircraft and refuelling are discussed in constraint 85u to 85x. Arrival and departure time of the aircraft at all nodes are computed from constraint 85y to 85af. Constraint 85ag and 85ai exempt the aircraft arrival and departure time if the vehicle does not visit the airport in question or if it is not deployed.

Pick-ups and deliveries of the requests are modelled from constraint 85aj to 85as, where each pick-up or delivery of the request is matched with an aircraft and then the resultant departure or arrival time is bounded by the time interval.

Another important aspect to consider is the minimal time difference requirement between two departures or arrivals at the airport, which guarantees safe take-off and landing at the runway. The whole process is achieved by constraint 85at to 85bb.

In order to constrain the model further and to guarantee that every aircraft always arrives at the second node of a airport after the first node of the same airport has been visited, constraints 85bc is added for general airport and 85bd for their hub.

Lastly, the daily operational cost budget is considered in 85be. The bounds of all decision variables are shown from 85bf to 85bm.

$$\textbf{minimize} \quad \sum_{k\in K}\sum_{(i,j)\in A^k} c_{ij}^k x_{ij}^k + \sum_{k\in K} c^k\left(w_{a_{h'^k}}^k - w_{d_{h'^k}}^k - t_n^k\right) + \sum_{r\in R}\left(\pi^r\left(q_n^r - \sum_{k\in K}\sum_{j:(j,i)\in A^k} q_{ij}^{rk}\right)\right) \tag{84}$$

Example of multiple Equations:

$$\text{subject to :} \quad \sum_{j:(j,i)\in A^k} x_{ji}^k - \sum_{j:(i,j)\in A^k} x_{ij}^k = 0 \qquad \forall k\in K, \forall i\in V^k\cup\{h'^k\} \tag{85a}$$

$$\sum_{j:(h^k,j)\in A^k} x_{h'^k j}^k - y^k = 0 \qquad \forall k\in K \tag{85b}$$

$$\begin{aligned}&\sum_{k\in K}\sum_{j:(j,i)\in A^k} q_{ji}^{rk} + \sum_{k\in K: i=h^k}\sum_{j:(j,i)\in A^k} q_{jh'^k}^{rk}\\ &-\sum_{k\in K}\sum_{j:(i,j)\in A^k} q_{ij}^{rk} - \sum_{k\in K: i=h^k}\sum_{j:(i,j)\in A^k} q_{h'^k j}^{rk} = 0\end{aligned} \qquad \forall r\in R, i\neq\{O^r, D^r\} \tag{85c}$$

$$\sum_{\forall r\in R} q_{ij}^{rk} - Q^k x_{ij}^k \leq 0 \qquad \forall k\in K, \forall (i,j)\in A^k \tag{85d}$$

$$\sum_{k\in K}\sum_{j:(O^r,j)\in A^k} q_{O^r j}^{rk} + \sum_{k\in K: O^r=h^k}\sum_{j:(O^r,j)\in A^k} q_{h'^k j}^{rk} \leq q_n^r \quad \forall r\in R \tag{85e}$$

$$\sum_{k\in K}\sum_{j:(j,O^r)\in A^k} q_{jO^r}^{rk} + \sum_{k\in K: O^r=h^k}\sum_{j:(j,O^r)\in A^k} q_{jh'^k}^{rk} = 0 \quad \forall r\in R \tag{85f}$$

$$\begin{aligned}&\sum_{k\in K}\sum_{j:(O^r,j)\in A^k} q_{O^r j}^{rk} + \sum_{k\in K: O^r=h^k}\sum_{j:(O^r,j)\in A^k} q_{h'^k j}^{rk}\\ &-\sum_{k\in K}\sum_{i:(i,D^r)\in A^k} q_{iD^r}^{rk} + \sum_{k\in K: D^r=h^k}\sum_{i:(i,D^r)\in A^k} q_{ih'^k}^{rk} = 0\end{aligned} \qquad \forall r\in R \tag{85g}$$

$$\sum_{j:(i,j)\in A^k}\sum_{b\in\{1,2\}} u_{i_a j_b}^k \leq 1 \qquad \forall k\in K, \forall i\in V^k, \forall a\in\{1,2\} \tag{85h}$$

$$\sum_{i:(i,j)\in A^k}\sum_{a\in\{1,2\}} u_{i_a j_b}^k \leq 1 \qquad \forall k\in K, \forall i\in V^k, \forall b\in\{1,2\} \tag{85i}$$

$$\sum_{a\in\{1,2\}}\sum_{b\in\{1,2\}} u_{i_a j_b}^k - x_{ij}^k = 0 \qquad \forall k\in K, \forall (i,j)\in A^k \tag{85j}$$

$$\sum_{j:(i,j)\in A^k}\sum_{a\in\{1,2\}}\sum_{b\in\{1,2\}} u_{i_a j_b}^k - \sum_{j:(i,j)\in A^k} x_{ij}^k = 0 \qquad \forall k\in K, \forall i\in V^k\backslash\{h^k, h'^k\} \tag{85k}$$

$$\sum_{j:(i,j)\in A^k}\sum_{b\in\{1,2\}} u_{i_2 j_b}^k - \frac{1}{2}\sum_{j:(i,j)\in A^k} x_{ij}^k \leq 0 \qquad \forall k\in K, \forall i\in V^k \tag{85l}$$

$$\sum_{i:(i,j)\in A^k}\sum_{a\in\{1,2\}}\sum_{b\in\{1,2\}} u_{i_a j_b}^k - \sum_{i:(i,j)\in A^k} x_{ij}^k = 0 \qquad \forall k\in K, \forall j\in V^k\backslash\{h^k, h'^k\} \tag{85m}$$

$$\sum_{i:(i,j)\in A^k} \sum_{a\in\{1,2\}} u^k_{i_a j_2} - \frac{1}{2} \sum_{i:(i,j)\in A^k} x^k_{ij} \le 0 \qquad \forall k \in K, \forall j \in V^k \tag{85n}$$

$$\sum_{j:(i,j)\in A^k} \sum_{b\in\{1,2\}} u^k_{i_a j_b} - \sum_{j:(j,i)\in A^k} \sum_{b\in\{1,2\}} u^k_{j_b i_a} = 0 \qquad \forall k \in K, \forall i \in V^k, \forall a \in \{1,2\} \tag{85o}$$

$$\sum_{j:(h^k,j)\in A^k} \sum_{b\in\{1,2\}} u^k_{h^k j_b} - \sum_{j:(j,h^k)\in A^k} \sum_{b\in\{1,2\}} u^k_{j_b h^k} = 0 \qquad \forall k \in K \tag{85p}$$

$$\sum_{j:(h^k,j)\in A^k} \sum_{b\in\{1,2\}} u^k_{h^k j_b} - y^k \le 0 \qquad \forall k \in K \tag{85q}$$

$$\sum_{j:(h^k,j)\in A^k} u^k_{h'^k j_1} - y^k = 0 \qquad \forall k \in K \tag{85r}$$

$$\sum_{j:(h^k,j)\in A^k} u^k_{h'^k j_2} = 0 \qquad \forall k \in K \tag{85s}$$

$$\Big(\sum_{i:(i,h^k)\in A^k} \sum_{a\in\{1,2\}} u^k_{i_a h'^k} - y^k = 0 \Big) \qquad \forall k \in K \tag{85t}$$

$$v^k_{i_a} - v^k_{j_b} + (d_{ij} + M) u^k_{i_a j_b} \le M \qquad \forall k \in K, i \notin V_{fuel} \cup \{h'^k\}, \forall (i,j) \in A^k, \forall a,b \in \{1,2\} \tag{85u}$$

$$v^k_{j_b} - d_{ij} u^k_{i_a j_b} \ge 0 \qquad \forall k \in K, i \in V_{fuel}, \forall (i,j) \in A^k, \forall a,b \in \{1,2\} \tag{85v}$$

$$v^k_{j_b} - d_{h^k j} u^k_{h^k j_b} \ge 0 \qquad \forall k \in K, \forall (i,j) \in A^k, \forall a,b \in \{1,2\} \tag{85w}$$

$$v^k_{j_1} - d_{h^k j} u^k_{h'^k j_1} \ge 0 \qquad \forall k \in K, \forall (i,j) \in A^k \tag{85x}$$

$$w^k_{d_{i_a}} - w^k_{a_{j_b}} + (t^k_{ij} + M) u^k_{i_a j_b} \le M \qquad \forall k \in K, \forall (i,j) \in A^k, \forall a,b \in \{1,2\} \tag{85y}$$

$$w^k_{d_{h^k}} - w^k_{a_{j_b}} + (t^k_{h^k j} + M) u^k_{h^k j_b} \le M \qquad \forall k \in K, \forall (h'^k, j) \in A^k, \forall b \in \{1,2\} \tag{85z}$$

$$w^k_{d_{i_a}} - w^k_{a_{h^k}} + (t^k_{ih^k} + M) u^k_{i_a h^k} \le M \qquad \forall k \in K, \forall (i, h^k) \in A^k, \forall a \in \{1,2\} \tag{85aa}$$

$$w^k_{d_{h'^k}} - w^k_{a_{j_1}} + (t^k_{h^k j} + M) u^k_{h'^k j_1} \le M \qquad \forall k \in K, \forall (h'^k, j) \in A^k \tag{85ab}$$

$$w^k_{d_{i_a}} - w^k_{a_{h'^k}} + (t^k_{ih^k} + M) u^k_{i_a h'^k} \le M \qquad \forall k \in K, \forall (i, h'^k) \in A^k, \forall a \in \{1,2\} \tag{85ac}$$

$$w^k_{d_{i_a}} - w^k_{a_{i_a}} - s_i \sum_{j:(j,i)\in A^k} \sum_{b\in\{1,2\}} u^k_{j_b i_a} \ge 0 \qquad \forall k \in K, \forall i \in V^k, \forall a \in \{1,2\} \tag{85ad}$$

$$w^k_{d_{h^k}} - w^k_{a_{h^k}} - s_i \sum_{j:(j,h^k)\in A^k} \sum_{b\in\{1,2\}} u^k_{j_b h^k} \ge 0 \qquad \forall k \in K \tag{85ae}$$

$$w^k_{d_{h'^k}} \le w^k_{a_{h'^k}} \qquad \forall k \in K \tag{85af}$$

$$w^k_{d_{i_a}} + w^k_{a_{i_a}} - M \sum_{j:(j,i)\in A^k} \sum_{b\in\{1,2\}} u^k_{j_b i_a} \le 0 \qquad \forall k \in K, \forall i \in V^k, \forall a \in \{1,2\} \tag{85ag}$$

$$w^k_{d_{h^k}} + w^k_{a_{h^k}} - M \sum_{j:(j,h^k)\in A^k} \sum_{b\in\{1,2\}} u^k_{j_b h^k} \le 0 \qquad \forall k \in K \tag{85ah}$$

$$\sum_{i\in V^k} (w^k_{d_i} + w^k_{a_i}) + w^k_{d_{h'^k}} + w^k_{a_{h'^k}} - M y^k \le 0 \qquad \forall k \in K \tag{85ai}$$

$$\sum_{a\in\{1,2\}} z^{rk}_{p_a} \le 1 \qquad \forall r \in R, \forall k \in K \tag{85aj}$$

$$\sum_{a\in\{1,2\}} z^{rk}_{p_a} - \sum_{j:(O^r,j)\in A^k} q^{rk}_{O^r j} \le 0 \qquad \forall r \in R, \forall k \in K, \forall (O^r, j) \in A^k \tag{85ak}$$

$$\sum_{a\in\{1,2\}} z^{rk}_{d_a} \le 1 \qquad \forall r \in R, \forall k \in K \tag{85al}$$

$$\sum_{a\in\{1,2\}} z^{rk}_{d_b} - \sum_{i:(i,D^r)\in A^k} q^{rk}_{iD^r} \le 0 \qquad \forall r \in R, \forall k \in K, \forall (i, D^r) \in A^k \tag{85am}$$

$$z^{rk}_{p_a} - \sum_{b\in\{1,2\}} u^k_{O^r_a j_b} - \frac{1}{Q^k} q^{rk}_{O^r j} \ge -1 \qquad \forall r \in R, \forall k \in K, \forall (O^r, j) \in A^k, \forall a \in \{1,2\} \tag{85an}$$

$$z^{rk}_{d_b} - \sum_{a\in\{1,2\}} u^k_{i_a D^r_b} - \frac{1}{Q^k} q^{rk}_{iD^r} \geq -1 \qquad \forall r \in R, \forall k \in K, \forall (i, D^r) \in A^k, \forall b \in \{1,2\} \tag{85ao}$$

$$w^k_{d_{O^r_a}} - t^r_{p_1} z^{rk}_{p_a} \geq 0 \qquad \forall r \in R, \forall k \in K, \forall a \in \{1,2\} \tag{85ap}$$

$$w^k_{d_{O^r_a}} - (t^r_{p_2} - M) z^{rk}_{p_a} \leq M \qquad \forall r \in R, \forall k \in K, \forall a \in \{1,2\} \tag{85aq}$$

$$w^k_{a_{D^r_b}} - t^r_{d_1} z^{rk}_{d_b} \geq 0 \qquad \forall r \in R, \forall k \in K, \forall b \in \{1,2\} \tag{85ar}$$

$$w^k_{a_{D^r_b}} - (t^r_{d_2} - M) z^{rk}_{d_b} \leq M \qquad \forall r \in R, \forall k \in K, \forall b \in \{1,2\} \tag{85as}$$

$$\sum_{k_2\in K} \sum_{b\in\{1,2\}} (s^{k_1k_2}_{a_{i_{ab}}} + s^{k_2k_1}_{a_{i_{ba}}} + s^{k_1k_2}_{d_{i_{ab}}} + s^{k_2k_1}_{d_{i_{ba}}}) - M(w^{k_1}_{d_{i_a}} + w^{k_1}_{a_{i_a}}) \leq 0 \qquad \forall k_1 \in K, \forall a \in \{1,2\}, \forall i \in V^{k_1} \cup \{h'^k\} \tag{85at}$$

$$s^{k_1k_2}_{a_{i_{ab}}} + s^{k_2k_1}_{a_{i_{ba}}} \leq 1 \qquad \forall k_1, k_2 \in K, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{85au}$$

$$s^{k_1k_2}_{d_{i_{ab}}} + s^{k_2k_1}_{d_{i_{ba}}} \leq 1 \qquad \forall k_1, k_2 \in K, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{85av}$$

$$w^{k_1}_{a_{i_a}} - w^{k_2}_{a_{i_b}} + M s^{k_1k_2}_{a_{i_{ab}}} \geq 0 \qquad \forall k_1, k_2 \in K, k_1 \neq k_2, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{85aw}$$

$$w^{k_1}_{d_{i_a}} - w^{k_2}_{d_{i_b}} + M s^{k_1k_2}_{d_{i_{ab}}} \geq 0 \qquad \forall k_1, k_2 \in K, k_1 \neq k_2, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{85ax}$$

$$s^{k_1k_2}_{a_{i_{ab}}} + s^{k_2k_1}_{a_{i_{ba}}} - (w^{k_2}_{a_{i_b}} + w^{k_1}_{a_{i_a}}) \leq 0 \qquad \forall k_1, k_2 \in K, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{85ay}$$

$$s^{k_1k_2}_{d_{i_{ab}}} + s^{k_2k_1}_{d_{i_{ba}}} - (w^{k_2}_{d_{i_b}} + w^{k_1}_{d_{i_a}}) \leq 0 \qquad \forall k_1, k_2 \in K, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{85az}$$

$$w^{k_2}_{a_{i_b}} - w^{k_1}_{a_{i_a}} - (M + \Delta t) s^{k_1k_2}_{a_{i_{ab}}} \geq -M \qquad \forall k_1, k_2 \in K, \forall i \in V^{k_1} \cap V^{k_2} \ \forall a, b \in \{1,2\} \tag{85ba}$$

$$w^{k_2}_{d_{i_b}} - w^{k_1}_{d_{i_a}} - (M + \Delta t) s^{k_1k_2}_{d_{i_{ab}}} \geq -M \qquad \forall k_1, k_2 \in K, \forall i \in V^{k_1} \cap V^{k_2} \ \forall a, b \in \{1,2\} \tag{85bb}$$

$$w^k_{a_{i_2}} - w^k_{a_{i_1}} - M(\sum_{j:(j,i)\in A^k} \sum_{a\in\{1,2\}} u^k_{j_a i_2}) \geq -M \qquad \forall k \in K, i \in V^k \tag{85bc}$$

$$w^k_{a_{h'^k}} \geq w^k_{a_{h^k}} \qquad \forall k \in K \tag{85bd}$$

$$\sum_{k\in K} \sum_{(i,j)\in A^k} c^k_{ij} x^k_{ij} \leq C_{n_{budget}} \qquad \forall k \in K, \forall (i,j) \in A^k \tag{85be}$$

$$s^{k_1k_2}_{a_{i_{ab}}}, \, s^{k_1k_2}_{d_{i_{ab}}} \in \{0,1\} \qquad \forall k_1, k_2 \in K, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{85bf}$$

$$u^k_{i_a j_b} \in \{0,1\} \qquad \forall k \in K, \forall (i,j) \in A^k, \forall a, b \in \{1,2\} \tag{85bg}$$

$$x^k_{ij} \in \{0,1\} \qquad \forall k \in K, \forall (i,j) \in A^k \tag{85bh}$$

$$y^k \in \{0,1\} \qquad \forall k \in K \tag{85bi}$$

$$q^{rk}_{ij} \in [0, Q^k] \qquad \forall k \in K, \forall r \in R, \forall (i,j) \in A^k \tag{85bj}$$

$$v^k_{i_a} \in [0, ra^k] \qquad \forall k \in K, \forall r \in R, \forall i \in V^k, \forall a \in \{1,2\} \tag{85bk}$$

$$w^k_{a_{i_a}}, \, w^k_{d_{i_a}} \geq 0 \qquad \forall k \in K, \forall i \in V^k \cup \{h'^k\}, \forall a \in \{1,2\} \tag{85bl}$$

$$z^{rk}_{p_a}, \, z^{rk}_{d_a} \in \{0,1\} \qquad \forall k \in K, \forall r \in R, \forall a \in \{1,2\} \tag{85bm}$$

C UNHAS mission information

Code	Name	Latitude	Longitude	Runway [m]
AGOK	Agok	9.35622	28.5831	1000
AJUON	Ajuong Thok	9.95956	30.2775	1000
ALEK	Alek	8.66222	28.1523	1000
AWL	Aweil	8.80144	27.3602	3000
DOR	Dorein	6.53747	33.2831	1000
BOR	Bor	6.19739	31.6015	1000
GANY	Ganyiel	7.40536	30.4755	50
HSAK	Akobo	7.78128	33.0027	1000
HSPA	Pochalla	7.18219	34.0947	1000
HSRN	Renk	11.6364	32.8078	1000
HSTR	Torit	4.42239	32.5723	1000
JCH	Jiech	8.36781	31.1352	50
JUB	Juba	4.87881	31.5931	3000
KAP	Kapoeta	4.78428	33.5836	1000
KOCH	Koch	8.60806	29.9953	50
KUACH	Kuach	8.90081	30.0285	50
KURWA	Kurwai	9.24528	31.1586	50
LER	Leer	8.30975	30.1111	50
LKEN	Lankien	8.52475	32.0628	50
MABAN	Maban	9.98178	33.7474	2000
MAK	Malakal	9.55289	31.6461	3000
MBR	Mabior	7.15906	31.4064	1000
MENIM	Menime	8.60664	30.7173	50
MINGK	Mingkaman	6.04878	31.5153	50
MNK	Mankein	9.04989	29.084	1000
MOGOK	Mogok	8.41639	31.3307	1000
MOTO	Motot	8.16747	32.0543	1000
NYAL	Nyal	7.72583	30.248	50
OLDFG	Old Fangak	9.07253	30.8792	1000
PAGL	Pagil	8.71444	31.266	50
PCL	Paloich	10.5183	32.4975	2000
PGK	Pagak	8.47292	34.0252	1000
PIBR	Pibor	6.80078	33.1243	1000
RUM	Rumbek	6.82192	29.6673	3000
RUB	Rubkona	9.30933	29.7874	2000
WAI	Wai	8.24908	31.2903	50
WAT	Waat	8.19147	32.1534	1000
WAU	Wau	7.72581	27.9636	3000
YAM	Yambio	4.56364	28.4221	2000
YEI	Yei	4.13067	30.7262	2000
YIDA	Yida	10.1167	30.0667	1000
HAAT	Haat	8.51517	30.6332	50
LAB	Labrab	6.65189	33.9526	1000
GUM	Gum	8.78511	33.0415	1000

Table 8: Operating aircraft for UNHAS mission in South Sudan

	Agok	Ajuong Thok	Akobo	Alek	Aweil	Bor	Dorein	Ganyiel	Gum	Haat	Jiech	Juba	Kapoeta	Koch	Kuach	Kurwai	Labrab	Lankien	Leer	Maban	Mabior	Malakal
Agok	0	106.6297	278.9014	48.87509	79.79114	261.1578	326.7014	162.3378	266.5502	131.6603	162.6142	323.1246	405.0633	95.0335	89.9382	152.7465	358.1341	212.3321	110.295	307.9541	213.4001	181.7893
Ajuong Thok	106.6297	0	207.9363	148.0564	186.2752	239.198	272.2148	153.8052	178.275	89.24715	108.2515	314.9408	367.8022	82.84993	65.25656	67.52878	295.1242	136.4336	99.54553	205.1909	181.0073	84.58601
Akobo	278.9014	207.9363	0	293.0334	340.7816	126.5528	76.52428	152.0871	60.31484	147.5626	116.4636	193.5009	183.2504	185.487	189.0357	140.4129	88.31348	71.50657	174.8095	139.3075	102.1102	133.406
Alek	48.87509	148.0564	293.0334	0	47.74821	253.1081	330.9076	157.3835	290.2456	147.5443	177.9994	306.0547	398.8127	109.4453	112.2416	181.7019	365.6216	232.295	118.2244	340.8172	213.5195	213.9128
Aweil	79.79114	186.2752	340.7816	47.74821	0	296.9457	377.7153	203.2615	337.1009	195.0312	225.6228	345.1336	442.4728	156.8211	158.4107	226.8076	412.88	279.6179	165.9726	384.9179	259.999	258.0085
Bor	261.1578	239.198	126.5528	253.1081	296.9457	0	102.3966	98.82651	177.4428	150.6325	133.239	79.17006	145.7078	173.4759	187.3782	184.8841	142.9013	142.4093	154.8057	260.5598	58.90025	201.4835
Dorein	326.7014	272.2148	76.52428	330.9076	377.7153	102.3966	0	175.2485	135.7129	197.4254	168.6008	141.8129	106.7831	231.823	240.0519	205.894	40.51614	139.6862	216.7627	208.629	117.9371	205.5402
Ganyiel	162.3378	153.8052	152.0871	157.3835	203.2615	98.82651	175.2485	0	173.571	67.28961	69.84673	165.7166	243.2935	77.64995	93.63547	117.6873	212.0763	115.8645	58.46647	248.2549	57.37705	146.4822
Gum	266.5502	178.275	60.31484	290.2456	337.1009	177.4428	135.7129	173.571	0	143.87	115.9156	249.9202	242.3759	181.1074	178.8895	115.0219	139.0763	60.16183	176.3173	83.12849	137.7823	94.68901
Haat	131.6603	89.24715	147.5626	147.5443	195.0312	150.6325	197.4254	67.28961	143.87	0	31.1021	225.7058	284.8283	38.28083	42.70862	53.78811	227.0245	84.88928	33.3731	204.4734	93.51153	86.54164
Jiech	162.6142	108.2515	116.4636	177.9994	225.6228	133.239	168.6008	69.84673	115.9156	31.1021	0	211.2534	260.0232	69.21297	73.07621	52.70218	196.8164	55.88701	60.94117	182.6477	74.34485	77.33559
Juba	323.1246	314.9408	193.5009	306.0547	345.1336	79.17006	141.8129	165.7166	249.9202	225.7058	211.2534	0	119.2228	243.3253	258.8598	263.44	176.6284	220.6881	224.155	332.1296	137.3607	280.6522
Kapoeta	405.0633	367.8022	183.2504	398.8127	442.4728	145.7078	106.7831	243.2935	242.3759	284.8283	260.0232	119.2228	0	313.8092	325.5502	304.3203	114.278	242.1968	296.1348	312.2128	192.9544	308.6864
Koch	95.0335	82.84993	185.487	109.4453	156.8211	173.4759	231.823	77.64995	181.1074	38.28083	69.21297	243.3253	313.8092	0	17.6869	78.89783	263.1468	122.8507	19.18487	237.1194	120.8768	113.1252
Kuach	89.9382	65.25656	189.0357	112.2416	158.4107	187.3782	240.0519	93.63547	178.8895	42.70862	73.07621	258.8598	325.5502	17.6869	0	70.1244	269.6629	122.8241	35.82433	229.6182	132.8385	103.5531
Kurwai	152.7465	67.52878	140.4129	181.7019	226.8076	184.8841	205.894	117.6873	115.0219	53.78811	52.70218	263.44	304.3203	78.89783	70.1244	0	227.6895	68.90718	83.77764	159.4981	126.1201	34.27794
Labrab	358.1341	295.1242	88.31348	365.6216	412.88	142.9013	40.51614	212.0763	139.0763	227.0245	196.8164	176.6284	114.278	263.1468	269.6629	227.6895	0	159.0379	249.3999	200.2996	154.7915	221.6524
Lankien	212.3321	136.4336	71.50657	232.295	279.6179	142.4093	139.6862	115.8645	60.16183	84.88928	55.88701	220.6881	242.1968	122.8507	122.8241	68.90718	159.0379	0	116.6365	132.7318	90.81738	66.49042
Leer	110.295	99.54553	174.8095	118.2244	165.9726	154.8057	216.7627	58.46647	176.3173	33.3731	60.94117	224.155	296.1348	19.18487	35.82433	83.77764	249.3999	116.6365	0	237.773	103.4975	117.7299
Maban	307.9541	205.1909	139.3075	340.8172	384.9179	260.5598	208.629	248.2549	83.12849	204.4734	182.6477	332.1296	312.2128	237.1194	229.6182	159.4981	200.2996	132.7318	237.773	0	219.1698	126.9696
Mabior	213.4001	181.0073	102.1102	213.5195	259.999	58.90025	117.9371	57.37705	137.7823	93.51153	74.34485	137.3607	192.9544	120.8768	132.8385	126.1201	154.7915	90.81738	103.4975	219.1698	0	144.4308
Malakal	181.7893	84.58601	133.406	213.9128	258.0085	201.4835	205.5402	146.4822	94.68901	86.54164	77.33559	280.6522	308.6864	113.1252	103.5531	34.27794	221.6524	66.49042	117.7299	126.9696	144.4308	0
Mankein	34.92092	89.31861	244.8893	59.97028	103.3271	227.5364	291.7852	128.787	235.278	97.36865	128.441	291.6653	370.7911	60.22415	56.72458	123.5363	323.3709	179.5389	75.4379	281.7458	178.7255	154.7836
Menime	134.328	85.308	144.5736	152.2915	199.5837	153.9345	196.8549	73.54387	138.3604	7.422981	28.66354	229.8304	286.1335	42.86112	44.52714	46.42744	225.4733	80.03457	40.17321	197.6092	96.09067	79.12182
Mingkaman	264.3148	246.0647	136.6716	254.3922	297.4914	10.30093	109.5042	102.3621	187.7417	157.1253	141.0641	70.39952	145.0779	178.3355	192.7559	193.0894	149.8769	152.1918	159.4509	270.8597	66.97759	210.5325
Mogok	172.4773	111.72	106.4487	189.2923	236.8379	134.2024	161.9676	79.19226	103.9471	41.84764	11.97301	212.9739	256.135	80.12814	82.58723	50.80368	188.6158	43.95768	72.73307	171.3095	75.62494	70.75296
Motot	217.9894	150.581	60.97367	233.647	281.3362	121.3203	122.1953	104.4699	69.37096	86.96428	55.91608	199.3599	222.6727	125.1288	128.088	83.74477	145.0995	21.45709	115.7835	148.1271	71.77856	86.63371
Nyal	139.1165	134.1256	163.9148	136.641	183.3347	122.1814	194.3844	23.52803	177.7516	52.63189	65.32726	188.8435	266.1132	55.05735	71.74136	106.0496	229.9003	118.0501	35.99166	247.8662	76.90143	137.5514
Old Fangak	137.1407	64.07752	148.0401	163.6297	209.3519	177.895	208.832	102.9315	129.4098	36.50968	44.95734	255.3611	303.7431	59.39419	51.49354	19.54207	233.5181	77.54628	64.62062	178.3874	119.082	53.81931
Pagak	327.1184	239.3661	73.60555	348.8549	396.1177	198.7287	124.3164	220.5947	61.32452	201.439	171.759	259.9872	223.0273	239.4101	238.5964	176.264	109.4208	116.5705	232.6948	92.0766	174.6056	155.262
Pagil	163.6823	94.96552	117.4213	184.827	231.8319	152.4395	177.4579	91.57465	105.4483	39.42627	22.21371	231.1178	273.4403	75.69473	74.27327	32.50159	202.2041	48.65054	72.75672	165.5263	93.75863	55.15338
Paloich	241.7778	135.3915	167.0364	280.3297	321.0587	264.8294	243.5169	222.0572	108.9278	163.2496	152.2501	342.8428	350.2812	187.3491	175.4437	110.0618	247.6855	122.4293	193.8083	80.56268	211.821	76.76869
Pibor	310.4811	254.0769	59.31318	316.1977	363.2884	97.79482	18.43022	161.9319	119.242	180.4609	151.2096	147.2385	124.1423	215.4767	223.1537	187.6056	50.1937	121.2564	200.9233	194.5411	104.6138	187.1337
Pochalla	352.5116	281.3476	74.29359	364.366	412.0331	159.9965	61.95911	215.9554	114.8177	220.8882	189.8981	203.5505	147.1704	258.385	262.8295	213.9663	32.94724	145.2665	246.4723	169.3483	160.1524	203.5
Renk	284.4928	180.0117	231.7535	327.9797	364.0865	334.3034	307.4386	289.1313	171.7519	227.2452	219.7529	412.0934	413.9802	246.3478	232.2338	173.4704	306.8681	191.9465	255.5692	113.7554	281.3429	142.6495
Rubkona	71.40387	48.63825	211.8009	104.4586	147.1099	215.7667	266.2751	121.4031	195.4985	69.21244	97.92676	286.9335	353.4902	43.87315	28.38847	81.34295	294.5848	142.9478	63.01303	237.8467	161.0014	111.057
Rumbek	165.2453	191.8392	206.8133	142.5895	181.5435	121.3208	216.2942	59.54371	232.7546	116.7823	127.4599	163.8282	263.9799	109.0012	126.652	170.3798	255.7152	175.4072	93.15214	307.7416	105.5941	201.7718
Torit	379.8103	359.4401	203.2995	366.4403	407.0741	121.3478	133.9078	218.5254	263.4304	271.5905	251.9199	64.68725	64.30852	294.5794	308.7081	301.5757	157.2321	248.1756	275.7445	341.043	178.4589	312.9404
Waat	223.0908	153.7466	56.18518	239.3036	286.9708	124.156	119.9457	110.4045	63.65578	92.37244	61.41189	201.6816	221.6458	130.5996	133.1564	86.53483	141.4882	20.72104	121.5594	143.1238	76.27568	87.10276
Wai	173.8323	118.9605	105.6146	187.997	235.6999	124.5722	156.9694	70.10516	108.854	42.17321	11.64643	203.1578	248.97	79.87374	84.51463	60.32001	185.246	48.78062	70.15616	178.9932	65.80946	81.07716
Wau	104.5737	191.9091	299.8016	57.33096	73.86641	235.4279	324.841	150.7364	308.3473	165.6012	192.4502	275.9077	379.0421	131.8536	141.511	210.5193	362.5282	248.3213	132.4007	368.8652	207.7689	244.5644
Yambio	287.9093	342.2885	334.7609	246.6062	262.1979	213.8706	313.7214	210.1015	374.2892	271.4349	279.9256	190.6867	309.1567	260.3207	277.4552	324.9627	353.4332	322.0302	246.4529	454.2435	236.7441	355.8302
Yei	338.7518	350.9891	257.9067	312.4077	344.8875	134.6731	210.3383	197.1839	311.7119	263.3063	255.568	68.62929	175.4851	272.3382	289.4086	308.1639	245.1562	275.61	253.5855	394.7036	186.311	330.1358
Yida	98.95271	15.63019	223.5607	143.1224	178.6823	252.3714	287.5409	164.5857	193.4737	101.8443	122.6177	327.3386	382.5021	90.6765	73.03599	83.1495	310.7008	152.0563	108.5201	217.75	194.566	99.37902

Table 9: Distance matrix, a

	Mankein	Menime	Mingkaman	Mogok	Motot	Nyal	Old Fangak	Pagak	Pagil	Paloich	Pibor	Pochalla	Renk	Rubkona	Rumbek	Torit	Waat	Wai	Wau	Yambio	Yei	Yida
Agok	34.92092	134.328	264.3148	172.4773	217.9894	139.1165	137.1407	327.1184	163.6823	241.7778	310.4811	352.5116	284.4928	71.40387	165.2453	379.8103	223.0908	173.8323	104.5737	287.9093	338.7518	98.95271
Ajuong Thok	89.31861	85.308	246.0647	111.72	150.581	134.1256	64.07752	239.3661	94.96552	135.3915	254.0769	281.3476	180.0117	48.63825	191.8392	359.4401	153.7466	118.9605	191.9091	342.2885	350.9891	15.63019
Akobo	244.8893	144.5736	136.6716	106.4487	60.97367	163.9148	148.0401	73.60555	117.4213	167.0364	59.31318	74.29359	231.7535	211.8009	206.8133	203.2995	56.18518	105.6146	299.8016	334.7609	257.9067	223.5607
Alek	59.97028	152.2915	254.3922	189.2923	233.647	136.641	163.6297	348.8549	184.827	280.3297	316.1977	364.366	327.9797	104.4586	142.5895	366.4403	239.3036	187.997	57.33096	246.6062	312.4077	143.1224
Aweil	103.3271	199.5837	297.4914	236.8379	281.3362	183.3347	209.3519	396.1177	231.8319	321.0587	363.2884	412.0331	364.0865	147.1099	181.5435	407.0741	286.9708	235.6999	73.86641	262.1979	344.8875	178.6823
Bor	227.5364	153.9345	10.30093	134.2024	121.3203	122.1814	177.895	198.7287	152.4395	264.8294	97.79482	159.9965	334.3034	215.7667	121.3208	121.3478	124.156	124.5722	235.4279	213.8706	134.6731	252.3714
Dorein	291.7852	196.8549	109.5042	161.9676	122.1953	194.3844	208.832	124.3164	177.4579	243.5169	18.43022	61.95911	307.4386	266.2751	216.2942	133.9078	119.9457	156.9694	324.841	313.7214	210.3383	287.5409
Ganyiel	128.787	73.54387	102.3621	79.19226	104.4699	23.52803	102.9315	220.5947	91.57465	222.0572	161.9319	215.9554	289.1313	121.4031	59.54371	218.5254	110.4045	70.10516	150.7364	210.1015	197.1839	164.5857
Gum	235.278	138.3604	187.7417	103.9471	69.37096	177.7516	129.4098	61.32452	105.4483	108.9278	119.242	114.8177	171.7519	195.4985	232.7546	263.4304	63.65578	108.854	308.3473	374.2892	311.7119	193.4737
Haat	97.36865	7.422981	157.1253	41.84764	86.96428	52.63189	36.50968	201.439	39.42627	163.2496	180.4609	220.8882	227.2452	69.21244	116.7823	271.5905	92.37244	42.17321	165.6012	271.4349	263.3063	101.8443
Jiech	128.441	28.66354	141.0641	11.97301	55.91608	65.32726	44.95734	171.759	22.21371	152.2501	151.2096	189.8981	219.7529	97.92676	127.4599	251.9199	61.41189	11.64643	192.4502	279.9256	255.568	122.6177
Juba	291.6653	229.8304	70.39952	212.9739	199.3599	188.8435	255.3611	259.9872	231.1178	342.8428	147.2385	203.5505	412.0934	286.9335	163.8282	64.68725	201.6816	203.1578	275.9077	190.6867	68.62929	327.3386
Kapoeta	370.7911	286.1335	145.0779	256.135	222.6727	266.1132	303.7431	223.0273	273.4403	350.2812	124.1423	147.1704	413.9802	353.4902	263.9799	64.30852	221.6458	248.97	379.0421	309.1567	175.4851	382.5021
Koch	60.22415	42.86112	178.3355	80.12814	125.1288	55.05735	59.39419	239.4101	75.69473	187.3491	215.4767	258.385	246.3478	43.87315	109.0012	294.5794	130.5996	79.87374	131.8536	260.3207	272.3382	90.6765
Kuach	56.72458	44.52714	192.7559	82.58723	128.088	71.74136	51.49354	238.5964	74.27327	175.4437	223.1537	262.8295	232.2338	28.38847	126.652	308.7081	133.1564	84.51463	141.511	277.4552	289.4086	73.03599
Kurwai	123.5363	46.42744	193.0894	50.80368	83.74477	106.0496	19.54207	176.264	32.50159	110.0618	187.6056	213.9663	173.4704	81.34295	170.3798	301.5757	86.53483	60.32001	210.5193	324.9627	308.1639	83.1495
Labrab	323.3709	225.4733	149.8769	188.6158	145.0995	229.9003	233.5181	109.4208	202.2041	247.6855	50.1937	32.94724	306.8681	294.5848	255.7152	157.2321	141.4882	185.246	362.5282	353.4332	245.1562	310.7008
Lankien	179.5389	80.03457	152.1918	43.95768	21.45709	118.0501	77.54628	116.5705	48.65054	122.4293	121.2564	145.2665	191.9465	142.9478	175.4072	248.1756	20.72104	48.78062	248.3213	322.0302	275.61	152.0563
Leer	75.4379	40.17321	159.4509	72.73307	115.7835	35.99166	64.62062	232.6948	72.75672	193.8083	200.9233	246.4723	255.5692	63.01303	93.15214	275.7445	121.5594	70.15616	132.4007	246.4529	253.5855	108.5201
Maban	281.7458	197.6092	270.8597	171.3095	148.1271	247.8662	178.3874	92.0766	165.5263	80.56268	194.5411	169.3483	113.7554	237.8467	307.7416	341.043	143.1238	178.9932	368.8652	454.2435	394.7036	217.75
Mabior	178.7255	96.09067	66.97759	75.62494	71.77856	76.90143	119.082	174.6056	93.75863	211.821	104.6138	160.1524	281.3429	161.0014	105.5941	178.4589	76.27568	65.80946	207.7689	236.7441	186.311	194.566
Malakal	154.7836	79.12182	210.5325	70.75296	86.63371	137.5514	53.81931	155.262	55.15338	76.76869	187.1337	203.5	142.6495	111.057	201.7718	312.9404	87.10276	81.07716	244.5644	355.8302	330.1358	99.37902
Mankein	0	100.4904	231.0945	138.6504	184.1148	105.3568	106.4496	295.2461	130.9962	220.3681	275.5966	318.2294	269.2277	44.50754	138.1929	347.031	189.3527	139.5056	103.6756	272.2306	311.1613	86.52732
Menime	100.4904	0	160.7588	38.17488	83.6784	59.7876	29.57569	196.5689	33.20686	155.8266	179.6209	218.3044	219.8971	69.43698	124.0337	274.5049	88.86581	40.23567	171.9878	278.6707	268.7403	98.51503
Mingkaman	231.0945	160.7588	0	142.5771	131.1975	125.8757	185.4524	208.6289	160.7349	274.6201	106.0825	168.2136	344.1301	221.1153	119.6267	116.3138	134.1466	132.7861	234.4256	205.2951	124.457	258.9681
Mogok	138.6504	38.17488	142.5771	0	45.51225	76.56123	47.64364	160.0576	18.30332	143.8767	144.2264	180.3392	212.1363	106.0975	137.706	250.978	50.70514	10.32885	204.4099	289.1477	259.8327	126.6162
Motot	184.1148	83.6784	131.1975	45.51225	0	110.6328	88.42275	118.515	57.18578	143.5653	103.8756	135.0539	212.9918	150.9853	163.4471	226.9711	6.062246	45.66634	244.6886	306.2301	255.0031	166.062
Nyal	105.3568	59.7876	125.8757	76.56123	110.6328	0	89.12477	228.9554	84.75002	214.2183	180.0825	231.3187	279.4124	98.92973	64.35346	242.0534	116.695	69.47841	135.9124	218.9174	217.7362	143.9493
Old Fangak	106.4496	29.57569	185.4524	47.64364	88.42275	89.12477	0	190.1144	31.44346	129.237	190.8527	222.2727	191.4923	66.25446	153.1397	296.8768	92.29763	55.13255	191.1198	307.7879	296.8522	79.01816
Pagak	295.2461	196.5689	208.6289	160.0576	118.515	228.9554	190.1144	0	164.4421	152.5243	113.8129	77.60593	203.1149	256.3445	277.61	258.1753	112.4744	163.0151	363.086	408.3636	326.6651	254.4584
Pagil	130.9962	33.20686	160.7349	18.30332	57.18578	84.75002	31.44346	164.4421	0	130.5522	159.4408	191.7144	197.6815	94.6761	148.1732	269.2131	61.34396	27.97762	205.0196	301.4337	277.0894	110.1552
Paloich	220.3681	155.8266	274.6201	143.8767	143.5653	214.2183	129.237	152.5243	130.5522	0	226.2783	221.5752	69.57915	175.9542	278.3197	366.0275	141.1825	153.8704	316.7578	431.9814	397.7419	145.6002
Pibor	275.5966	179.6209	106.0825	144.2264	103.8756	180.0825	190.8527	113.8129	159.4408	226.2783	0	62.20165	290.9409	249.0558	206.0937	146.5586	101.5478	139.5639	312.3353	311.3766	215.0323	269.4401
Pochalla	318.2294	218.3044	168.2136	180.3392	135.0539	231.3187	222.2727	77.60593	191.7144	221.5752	62.20165	0	278.0818	286.0199	264.7224	189.0098	130.4398	178.7266	366.4564	373.4732	272.1462	296.9751
Renk	269.2277	219.8971	344.1301	212.1363	212.9918	279.4124	191.4923	203.1149	197.6815	69.57915	290.9409	278.0818	0	226.5304	343.7658	433.3617	210.4267	222.2932	370.5306	498.197	467.3203	185.5944
Rubkona	44.50754	69.43698	221.1153	106.0975	150.9853	98.92973	66.25446	256.3445	94.6761	175.9542	249.0558	286.0199	226.5304	0	149.5162	337.0951	155.616	109.5643	144.1052	296.3198	315.9257	51.21292
Rumbek	138.1929	124.0337	119.6267	137.706	163.4471	64.35346	153.1397	277.61	148.1732	278.3197	206.0937	264.7224	343.7658	149.5162	0	225.5653	169.2911	129.1252	115.071	154.6562	173.5333	199.2346
Torit	347.031	274.5049	116.3138	250.978	226.9711	242.0534	296.8768	258.1753	269.2131	366.0275	146.5586	189.0098	433.3617	337.0951	225.5653	0	227.674	242.1543	339.1555	248.5602	111.9086	373.0108
Waat	189.3527	88.86581	134.1466	50.70514	6.062246	116.695	92.29763	112.4744	61.34396	141.1825	101.5478	130.4398	210.4267	155.616	169.2911	227.674	0	51.40572	250.6958	311.4432	258.2618	169.289
Wai	139.5056	40.23567	132.7861	10.32885	45.66634	69.47841	55.13255	163.0151	27.97762	153.8704	139.5639	178.7266	222.2932	109.5643	129.1252	242.1543	51.40572	0	200.2755	279.7122	249.5524	133.538
Wau	103.6756	171.9878	234.4256	204.4099	244.6886	135.9124	191.1198	363.086	205.0196	316.7578	312.3353	366.4564	370.5306	144.1052	115.071	339.1555	250.6958	200.2755	0	191.8202	271.6674	190.1696
Yambio	272.2306	278.6707	205.2951	289.1477	306.2301	218.9174	307.7879	408.3636	301.4337	431.9814	311.3766	373.4732	498.197	296.3198	154.6562	248.5602	311.4432	279.7122	191.8202	0	140.3731	347.4814
Yei	311.1613	268.7403	124.457	259.8327	255.0031	217.7362	296.8522	326.6651	277.0894	397.7419	215.0323	272.1462	467.3203	315.9257	173.5333	111.9086	258.2618	249.5524	271.6674	140.3731	0	361.5423
Yida	86.52732	98.51503	258.9681	126.6162	166.062	143.9493	79.01816	254.4584	110.1552	145.6002	269.4401	296.9751	185.5944	51.21292	199.2346	373.0108	169.289	133.538	190.1696	347.4814	361.5423	0

Table 10: Distance matrix, b

D Daily Requests

Pax request out of location

Between 13/Apr/2015 and 13/Apr/2015		
From	**To**	**Count**
AGOK	JUB	9
AWL	JUB	14
AWL	RUM	2
BOR	JUB	6
BOR	PIBR	8
JUB	AGOK	11
JUB	AWL	17
JUB	BOR	6
JUB	KOCH	4
JUB	LER	4
JUB	MAK	35
JUB	MINGK	15
JUB	MNK	1
JUB	PIBR	16
JUB	RUB	28
JUB	RUM	13
JUB	WAU	27
JUB	YAM	12
JUB	YIDA	16
KOCH	JUB	1
LER	JUB	5
MAK	JUB	16
MINGK	JUB	7
PIBR	JUB	9
RUB	JUB	15
RUB	MNK	2
RUM	JUB	20
WAU	JUB	20
WAU	RUM	2
YAM	JUB	20
YIDA	JUB	10
	Total	**371**

Pax request out of location

Between 14/Apr/2015 and 14/Apr/2015		
From	**To**	**Count**
AJUON	JUB	10
ALEK	JUB	4
AWL	JUB	14
AWL	WAU	2
PCL	JUB	5
HSTR	JUB	1
JCH	JUB	2
JUB	AJUON	5
JUB	ALEK	9
JUB	AWL	12
JUB	BOR	1
JUB	GANY	5
JUB	PCL	7
JUB	HSTR	4
JUB	JCH	3
JUB	KAP	13
JUB	KURWA	5
JUB	LKEN	9
JUB	MABAN	27
JUB	MAK	7
JUB	MBR	6
JUB	MOGOK	7
JUB	MOTO	2
JUB	NYAL	6
JUB	OLDFG	4
JUB	PGK	23
JUB	RUM	18
JUB	WAU	11
KAP	HSTR	1
KAP	JUB	11
LKEN	JUB	3
MABAN	JUB	16
MAK	JUB	2
MBR	JUB	4
MOTO	JUB	1
NYAL	JUB	5
OLDFG	JUB	14
PGK	JUB	4
RUM	JUB	6
WAU	JUB	9
	Total	**298**

Pax request out of location

Between 15/Apr/2015 and 15/Apr/2015		
From	**To**	**Count**
AGOK	JUB	7
BOR	HSPA	5
GANY	JUB	1
HAAT	JUB	2
JUB	AGOK	11
JUB	BOR	1
JUB	GANY	1
JUB	LER	5
JUB	MABAN	23
JUB	MAK	25
JUB	MINGK	22
JUB	PAGL	1
JUB	HSPA	9
JUB	RUB	19
JUB	RUM	20
JUB	WAU	8
JUB	YEI	4
JUB	YIDA	13
LER	JUB	6
MABAN	JUB	7
MAK	JUB	23
MINGK	JUB	9
HSPA	BOR	5
HSPA	JUB	6
RUB	JUB	25
RUM	JUB	8
WAU	JUB	5
WAU	RUM	4
YEI	JUB	3
YIDA	JUB	15
	Total	**293**

Pax request out of location

Between 16/Apr/2015 and 16/Apr/2015		
From	**To**	**Count**
AJUON	JUB	9
ALEK	JUB	3
AWL	JUB	11
BOR	JUB	3
BOR	PIBR	4
GANY	JUB	4
HSAK	JUB	6
PCL	JUB	4
HSRN	JUB	1
HSTR	JUB	4
HSTR	KAP	1
JUB	AJUON	10
JUB	ALEK	7
JUB	AWL	13
JUB	BOR	15
JUB	GANY	2
JUB	HSAK	20
JUB	PCL	11
JUB	HSRN	3
JUB	KAP	5
JUB	LKEN	5
JUB	MABAN	10
JUB	PIBR	14
JUB	RUM	5
JUB	WAT	6
JUB	WAU	24
JUB	YAM	12
KAP	JUB	11
LKEN	JUB	10
MABAN	JUB	17
NYAL	GANY	1
NYAL	JUB	2
OLDFG	JUB	4
PGK	JUB	14
PIBR	BOR	3
PIBR	JUB	2
RUM	JUB	8
WAT	JUB	2
WAU	AWL	1
WAU	JUB	7
WAU	RUM	1
YAM	JUB	10
	Total	**305**

Pax request out of location

Between 17/Apr/2015 and 17/Apr/2015		
From	**To**	**Count**
AGOK	JUB	11
AWL	JUB	12
BOR	JUB	9
JUB	AGOK	10
JUB	AWL	20
JUB	BOR	11
JUB	KOCH	5
JUB	KUACH	3
JUB	LER	6
JUB	MABAN	22
JUB	MAK	21
JUB	MENIM	2
JUB	MINGK	6
JUB	RUB	16
JUB	RUM	22
JUB	WAI	5
JUB	WAU	27
JUB	YAM	19
JUB	YIDA	9
KOCH	JUB	8
LER	JUB	7
MABAN	JUB	34
MAK	JUB	16
MINGK	BOR	4
MINGK	JUB	15
RUB	JUB	23
RUM	JUB	29
RUM	WAU	2
WAU	JUB	16
YAM	JUB	16
YIDA	JUB	14
	Total	**420**

Table 11: Daily requests test data (13/04/2015 to 17/04/2015), 5 days

Pax request out of location		
Day 1		
From	**To**	**Count**
RUB	JUB	15
RUM	JUB	20
WAU	JUB	20
WAU	RUM	2
	Total	57

Pax request out of location		
Day 2		
From	**To**	**Count**
RUB	JUB	15
RUM	JUB	20
WAU	JUB	20
WAU	RUM	2
	Total	57

Pax request out of location		
Day 3		
From	**To**	**Count**
RUB	JUB	15
RUM	JUB	20
WAU	JUB	20
WAU	RUM	2
	Total	57

Pax request out of location		
Day 4		
From	**To**	**Count**
RUB	JUB	15
RUM	JUB	20
WAU	JUB	20
WAU	RUM	2
	Total	57

Pax request out of location		
Day 5		
From	**To**	**Count**
RUB	JUB	15
RUM	JUB	20
WAU	JUB	20
WAU	RUM	2
	Total	57

Table 12: Simple daily request test data, 5 days

Aircraft	**Aircraft type**	**Cruising Speed [nm/hr]**	**Cost [-/nm]**	**Seats**	**Range [nm]**	**Runway required [m]**	**Hub**
Fokker 50	Fokker 50	230	20	50	1080	3000	Juba
Dash 8_1	DHC8-106	200	18	37	1020	2000	Juba
Dornier 228	Dornier 228	220	11	15	1000	1000	Juba
Cessna 208_1	Cessna 208B	180	10	10	1070	1000	Juba
MilMi8_1	Mi8-T	120	32	17	355	50	Juba
Cessna 208_1R	Cessna 208B	180	11	10	1070	1000	Rumbek
MilMi8_1R	Mi8-T	120	32	17	355	50	Rumbek

Table 13: Fleet used by the simplified situation

E Tabu Search Model Result (Simple Scenario)

Day 2, Allow spillage

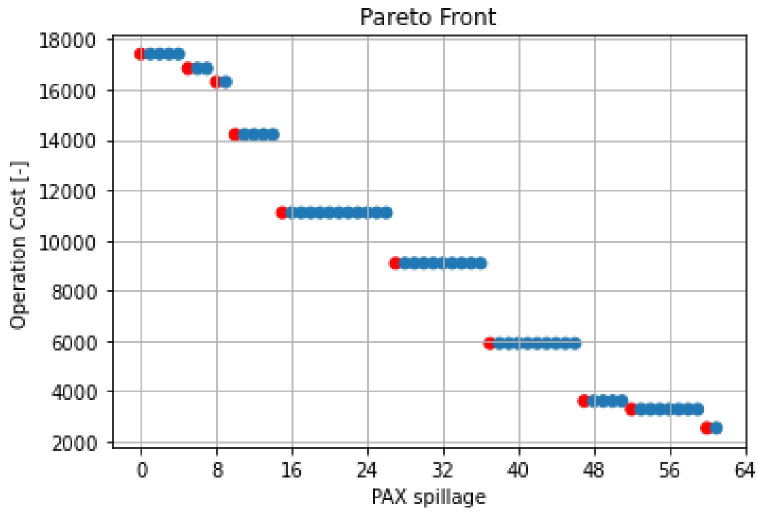


Figure 11: Pareto Front of simplified situation (day 2)

AC	From	Departure	To	Arrival	Request	PAX
Fokker 50	JUB	00:00	RUB	00:40	0	0
Fokker 50	RUB	01:40	WAU	02:00	1474	5
Fokker 50	WAU	03:00	RUM	03:16	1474	5
Fokker 50	WAU	03:00	RUM	03:16	1603	20
Fokker 50	WAU	03:00	RUM	03:16	1624	2
Fokker 50	RUM	04:16	JUB	04:40	1474	5
Fokker 50	RUM	04:16	JUB	04:40	1431	25
Fokker 50	RUM	04:16	JUB	04:40	1603	20

Table 14: Final flight plan of simplified situation (day 2)

Requst	From	To	Demand	Spillage
1474	RUB	JUB	15	10
1431	RUM	JUB	25	0
1603	WAU	JUB	20	0
1624	WAU	RUM	2	0

Table 15: Summary of the overall passenger spillage of simplified situation (day 2)

Day 3, Allow spillage

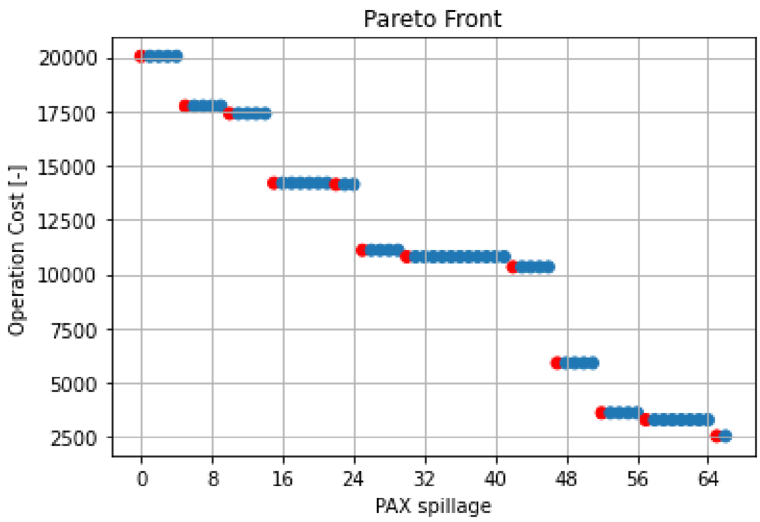


Figure 12: Pareto Front of simplified situation (day 3)

AC	From	Departure	To	Arrival	Request	PAX
Fokker 50	JUB	00:00	WAU	00:39	0	0
Fokker 50	WAU	01:39	RUB	01:59	1603	20
Fokker 50	WAU	01:39	RUB	01:59	1624	2
Fokker 50	RUB	02:59	JUB	03:39	1474	25
Fokker 50	RUB	02:59	JUB	03:39	1603	20
Fokker 50	RUB	02:59	JUB	03:39	1624	2
Dornier 228	JUB	04:39	RUM	05:03	1624	2
Dornier 228	RUM	06:03	JUB	06:27	1431	15

Table 16: Final flight plan of simplified situation (day 3)

Requst	From	To	Demand	Spillage
1474	RUB	JUB	25	0
1431	RUM	JUB	20	5
1603	WAU	JUB	20	0
1624	WAU	RUM	2	0

Table 17: Summary of the overall passenger spillage of simplified situation (day 3)

Day 4, Allow spillage

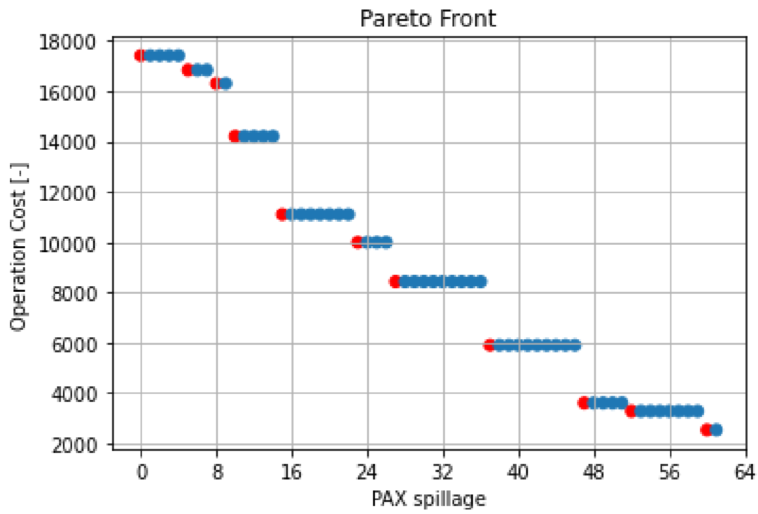


Figure 13: Pareto Front of simplified situation (day 4)

AC	From	Departure	To	Arrival	Request	PAX
Fokker 50	JUB	00:00	RUB	00:40	0	0
Fokker 50	RUB	01:40	WAU	02:00	1474	5
Fokker 50	WAU	03:00	RUM	03:16	1474	5
Fokker 50	WAU	03:00	RUM	03:16	1603	20
Fokker 50	WAU	03:00	RUM	03:16	1624	2
Fokker 50	RUM	04:16	JUB	04:40	1474	5
Fokker 50	RUM	04:16	JUB	04:40	1431	25
Fokker 50	RUM	04:16	JUB	04:40	1603	20

Table 18: Final flight plan of simplified situation (day 4)

Requst	From	To	Demand	Spillage
1474	RUB	JUB	15	10
1431	RUM	JUB	25	0
1603	WAU	JUB	20	0
1624	WAU	RUM	2	0

Table 19: Summary of the overall passenger spillage of simplified situation (day 4)

Day 5, Allow spillage

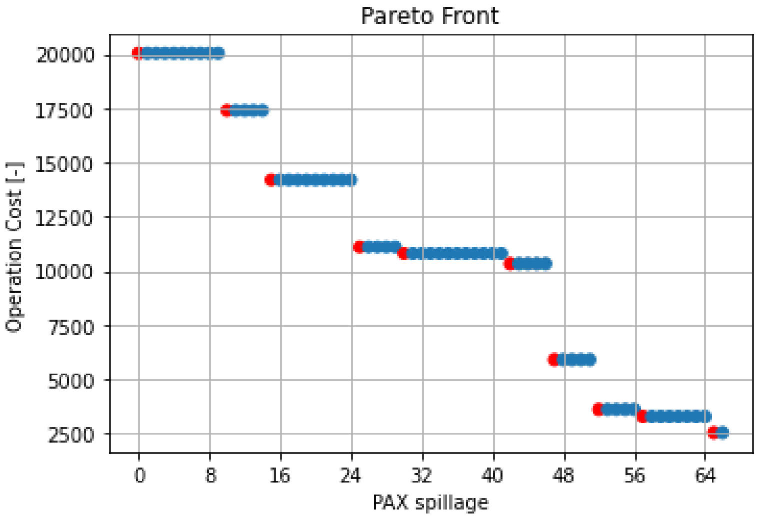


Figure 14: Pareto Front of simplified situation (day 5)

AC	From	Departure	To	Arrival	Request	PAX
Fokker 50	JUB	00:00	RUB	00:40	0	0
Fokker 50	RUB	01:40	WAU	02:00	1474	25
Fokker 50	WAU	03:00	RUM	03:16	1474	25
Fokker 50	WAU	03:00	RUM	03:16	1603	5
Fokker 50	WAU	03:00	RUM	03:16	1624	2
Fokker 50	RUM	04:16	JUB	04:40	1474	25
Fokker 50	RUM	04:16	JUB	04:40	1431	20
Fokker 50	RUM	04:16	JUB	04:40	1603	5

Table 20: Final flight plan of simplified situation (day 5)

Requst	From	To	Demand	Spillage
1474	RUB	JUB	25	0
1431	RUM	JUB	20	0
1603	WAU	JUB	20	15
1624	WAU	RUM	2	0

Table 21: Summary of the overall passenger spillage of simplified situation (day 5)

Day 1, No spillage

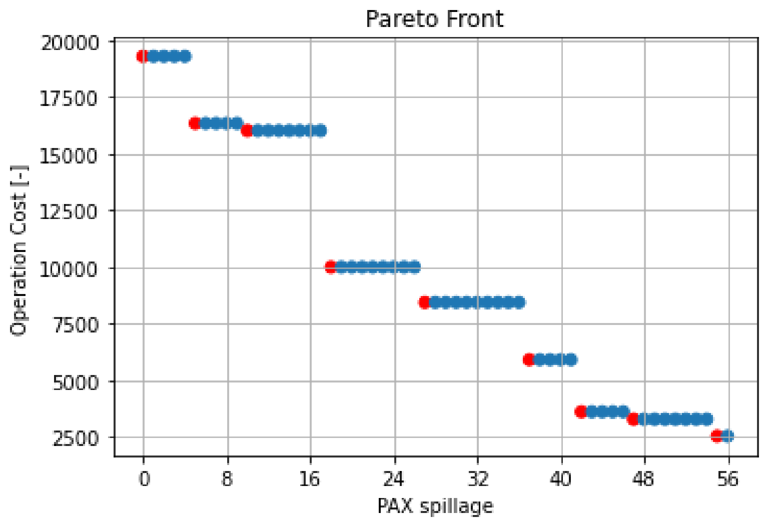


Figure 15: Pareto Front of simplified situation (day 1, no spillage)

AC	From	Departure	To	Arrival	Request	PAX
Fokker 50	JUB	04:54	RUM	05:17	1624	2
Fokker 50	RUM	06:17	JUB	06:40	1431	20
Dash 8_1	JUB	00:00	WAU	00:44	0	0
Dash 8_1	WAU	01:44	RUB	02:07	1603	20
Dash 8_1	WAU	01:44	RUB	02:07	1624	2
Dash 8_1	RUB	03:07	JUB	03:54	1474	15
Dash 8_1	RUB	03:07	JUB	03:54	1603	20
Dash 8_1	RUB	03:07	JUB	03:54	1624	2

Table 22: Final flight plan of simplified situation (day 1, no spillage)

Day 2, No spillage

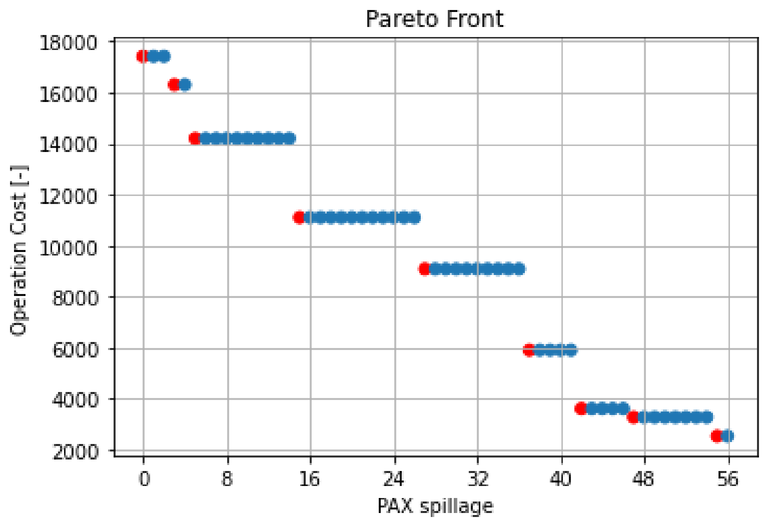


Figure 16: Pareto Front of simplified situation (day 2, no spillage)

AC	From	Departure	To	Arrival	Request	PAX
Fokker 50	JUB	00:06	WAU	00:45	0	0
Fokker 50	WAU	01:45	RUM	02:01	1603	20
Fokker 50	WAU	01:45	RUM	02:01	1624	2
Fokker 50	RUM	03:01	JUB	03:24	1431	20
Fokker 50	RUM	03:01	JUB	03:24	1603	20
Dornier 228	JUB	00:00	RUB	00:42	0	0
Dornier 228	RUB	01:42	JUB	02:24	1474	15

Table 23: Final flight plan of simplified situation (day 2, no spillage)

Day 3, No spillage

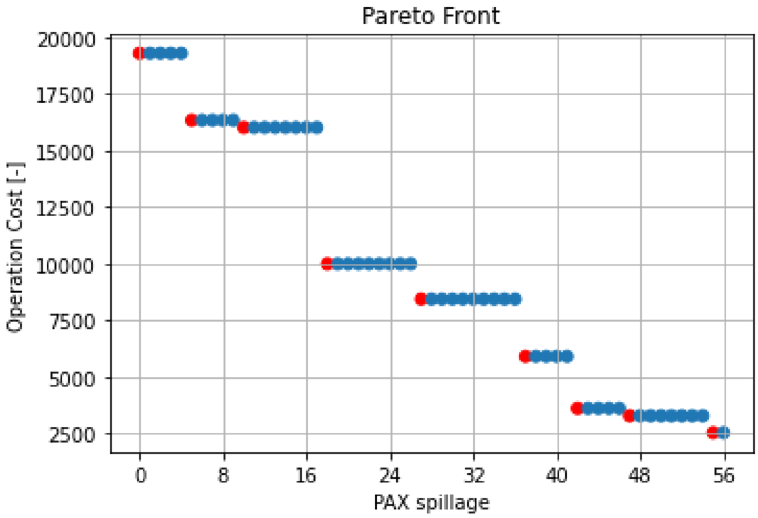


Figure 17: Pareto Front of simplified situation (day 3, no previous spillage)

AC	From	Departure	To	Arrival	Request	PAX
Fokker 50	JUB	04:54	RUM	05:17	1624	2
Fokker 50	RUM	06:17	JUB	06:40	1431	20
Dash 8_1	JUB	00:00	WAU	00:44	0	0
Dash 8_1	WAU	01:44	RUB	02:07	1603	20
Dash 8_1	WAU	01:44	RUB	02:07	1624	2
Dash 8_1	RUB	03:07	JUB	03:54	1474	15
Dash 8_1	RUB	03:07	JUB	03:54	1603	20
Dash 8_1	RUB	03:07	JUB	03:54	1624	2

Table 24: Final flight plan of simplified situation (day 3, no spillage)

Day 4, No spillage

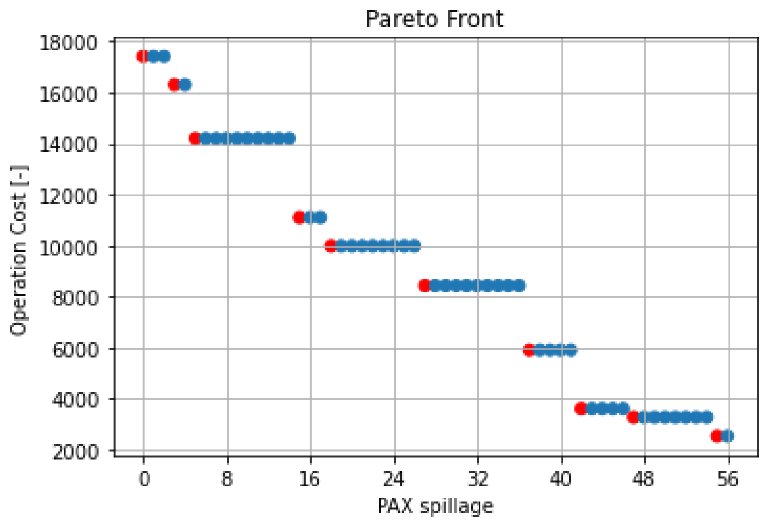


Figure 18: Pareto Front of simplified situation (day 4, no previous spillage)

AC	From	Departure	To	Arrival	Request	PAX
Fokker 50	JUB	00:06	WAU	00:45	0	0
Fokker 50	WAU	01:45	RUM	02:01	1603	20
Fokker 50	WAU	01:45	RUM	02:01	1624	2
Fokker 50	RUM	03:01	JUB	03:24	1431	20
Fokker 50	RUM	03:01	JUB	03:24	1603	20
Dornier 228	JUB	00:00	RUB	00:42	0	0
Dornier 228	RUB	01:42	JUB	02:24	1474	15

Table 25: Final flight plan of simplified situation (day 4, no spillage)

Day 5, No spillage

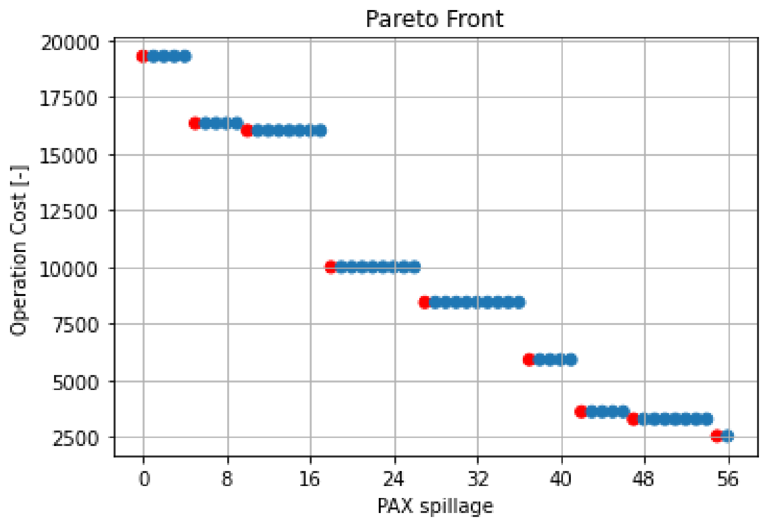
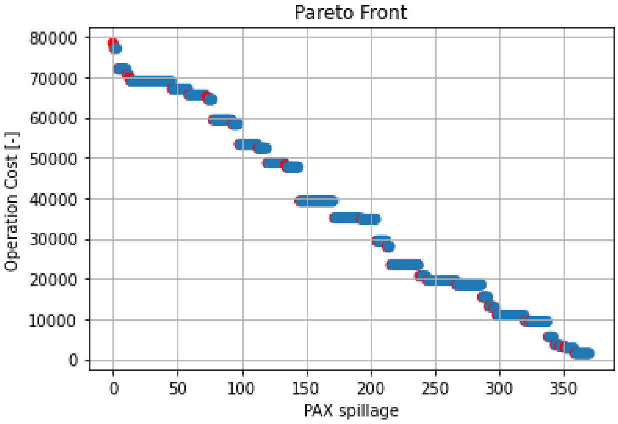


Figure 19: Pareto Front of simplified situation (day 5, no previous spillage)

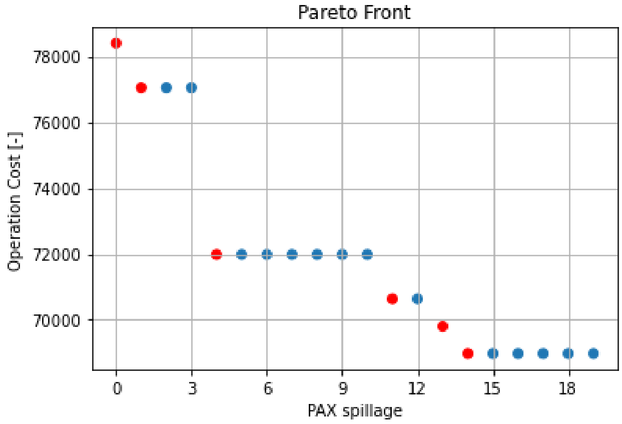
AC	From	Departure	To	Arrival	Request	PAX
Fokker 50	JUB	04:54	RUM	05:17	1624	2
Fokker 50	RUM	06:17	JUB	06:40	1431	20
Dash 8_1	JUB	00:00	WAU	00:44	0	0
Dash 8_1	WAU	01:44	RUB	02:07	1603	20
Dash 8_1	WAU	01:44	RUB	02:07	1624	2
Dash 8_1	RUB	03:07	JUB	03:54	1474	15
Dash 8_1	RUB	03:07	JUB	03:54	1603	20
Dash 8_1	RUB	03:07	JUB	03:54	1624	2

Table 26: Final flight plan of simplified situation (day 5, no spillage)

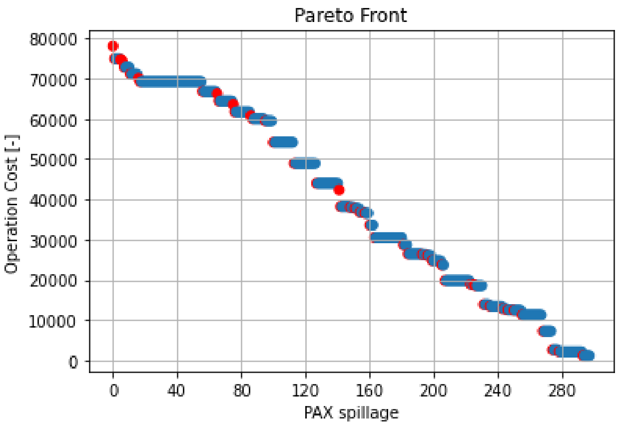
F Tabu Search Model Result (UNHAS Scenario)



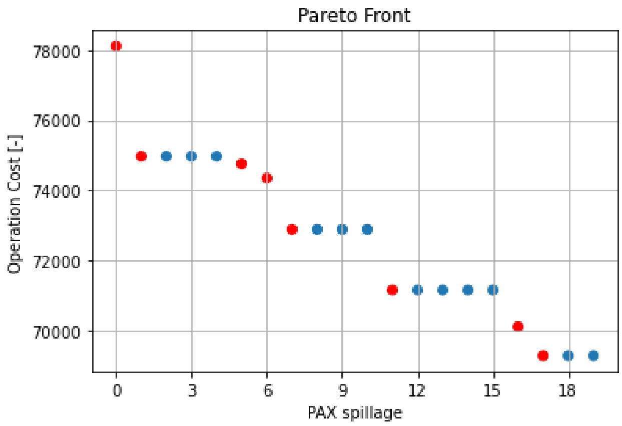
(a) Pareto Front of request on 13/04/2015



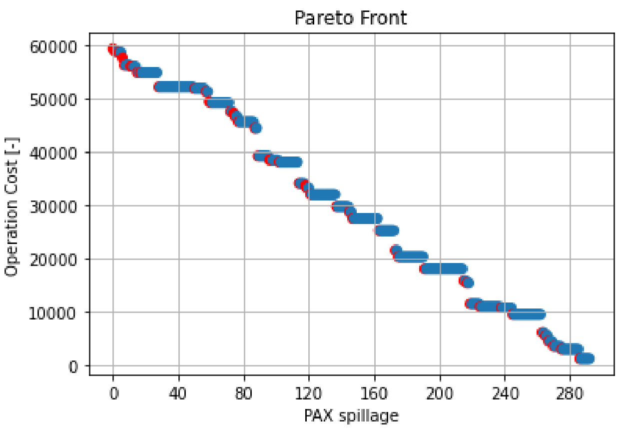
(b) Pareto Front of request on 13/04/2015 (PAX spillage<20)



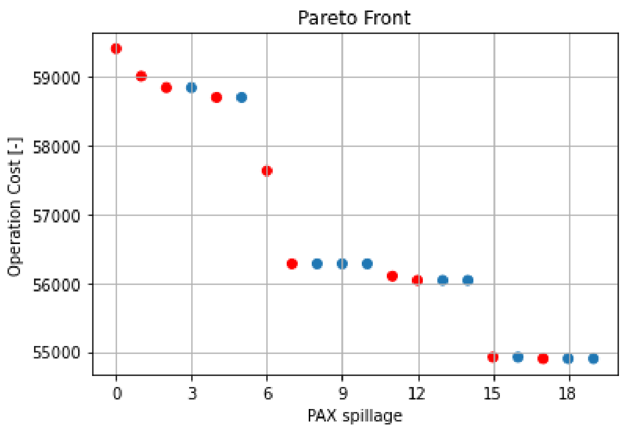
(c) Pareto Front of request on 14/04/2015



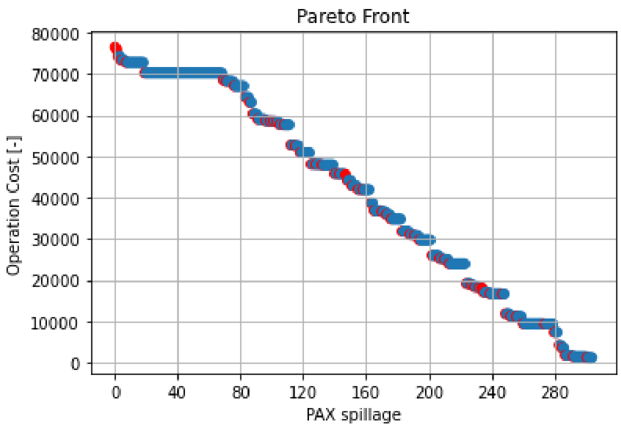
(d) Pareto Front of request on 14/04/2015 (PAX spillage<20)



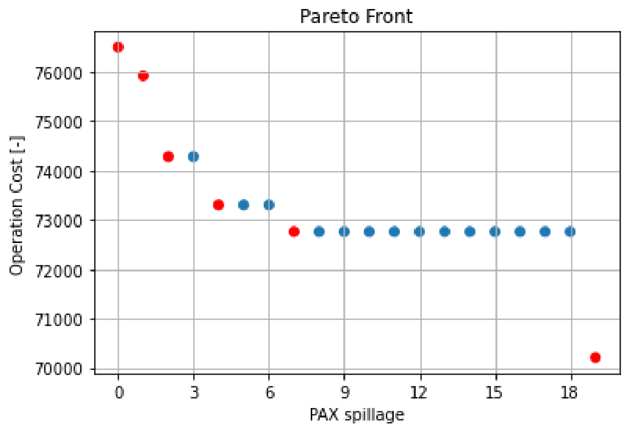
(e) Pareto Front of request on 15/04/2015



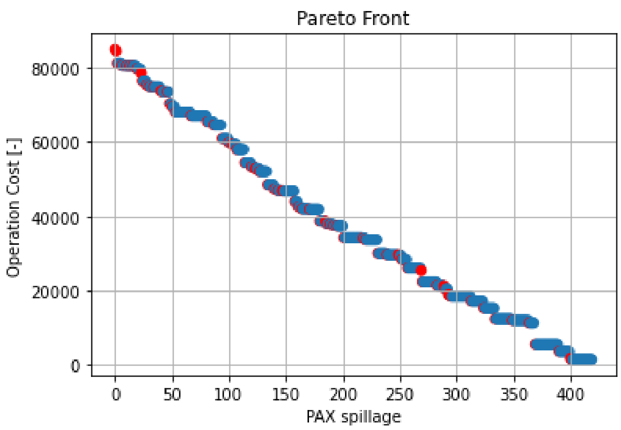
(f) Pareto Front of request on 15/04/2015 (PAX spillage<20)



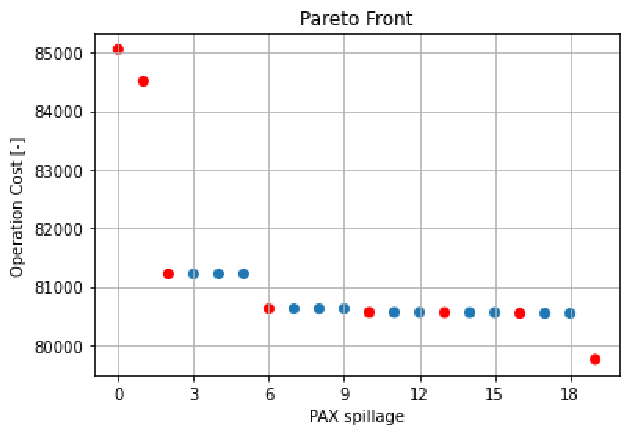
(g) Pareto Front of request on 16/04/2015



(h) Pareto Front of request on 16/04/2015 (PAX spillage<20)



(i) Pareto Front of request on 17/04/2015



(j) Pareto Front of request on 17/04/2015 (PAX spillage<20)

G Exact Method Verification

To verify and examine the LP model in B, few simple scenarios with obvious solutions are created to testify its functionality.

G.1 Set-up

G.1.1 Environment Set-up

Before implementing the actual situation of UNHAS service in South Sudan, a simple map is considered to validate the model. As shown in Figure 20, the map is formed by three airports, creating a simple right triangle network. For the convenience of calculation, the side lengths of the triangle are 300, 400 and 500 km. On top of this map, a number of aircraft will be distributed. Each of their hubs is assigned to one of the three airports. Moreover, requests will be created to transfer a certain amount of passengers from one airport to another within the map.

Table 27 provide the flight legs of the simplified map for the scenario test. the 2 in the 'Direction' section means that this flight leg is bidirectional. Table 28 provides each airport information and their locations in the Cartesian coordinate system for the scenario test environment.

From	To	Direction
AP1	AP2	2
AP1	AP3	2
AP2	AP3	2

Table 27: Map for scenario test.

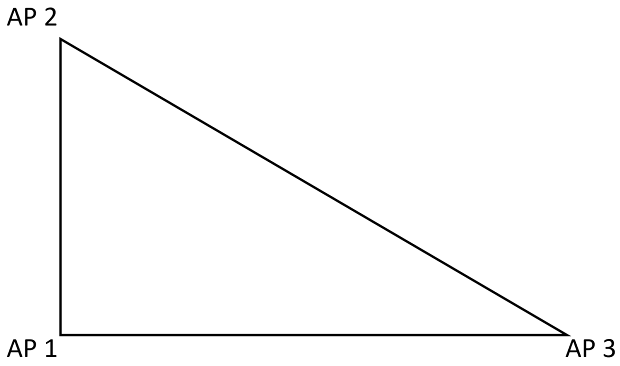


Figure 20: Simple map for model verification.

Name	x	y	l_{runway}	Type	TAT
AP1	0	0	1000	Refuel	1
AP2	0	300	600		1
AP3	400	0	2000	Refuel	1

Table 28: Airport information for scenario test.

G.1.2 Objective Function Set-up

As can be noticed from the objective function stated in polynomial Equation 84 of Appendix B, the total cost is mainly dominated by the operational cost and the penalty for passenger spillage. The penalty of each passenger spilt is considered to be the same, which is considered as the average passenger cost.

When creating different scenarios for the experiments, the Pareto front of each case also varies. Ignorance of this change can result in unexpected output, such as no passenger delivered during the day. Therefore, the weights of operational cost and the penalty for passenger spillage need to be re-balanced for each scenario in order to generate a reasonable plan.

In every scenario, all passenger requests are preferably accomplished in the plan. This tendency leads to high penalty of the passenger spillage. In order to solve the influence originated from the Pareto front variation, the Pareto front of all scenarios and the control group will first be generated to choose the point that is suitable for all cases.

G.2 Classification

Based on the introduction of the model constraints stated in Appendix B, constraints to be validated are divided into six types:

- Operational cost minimisation.
- Aircraft range limitation.
- Aircraft time limitation.
- Request time limitation.
- Request split to multiple aircraft.
- Request cannot be fulfilled directly.

The effectiveness of these aspects are essential criteria to evaluate the functionality of the model. Therefore, a number of scenarios are created to test the model. Each scenario consists of a number of requests to be filled with a number of passengers per request. A control group is designed as a reference to determine if the model can generate the desired output for the scenario. The control group set-up consists two parts: The fleet information is shown in Table 29, and Table 30 shows the parameters set for the initial request.

Since all parameters in the scenario tests are conceived number for the verification and they have limited relevance to the actual real-world scenario. Therefore, to simplify the process, All parameters discussed in the scenario test section are dimensionless quantities.

AC	type	$v_{cruising}$	Cost/Distance	Seats	Range	Runway required	Hub
1	2	200	11	15	600	500	AP1
2	2	200	18	15	1000	500	AP3

Table 29: Control group fleet set-up.

Origin	Destination	#PAX	Penalty (Weight)	$t^r_{p_1}$	$t^r_{p_2}$	$t^r_{d_1}$	$t^r_{d_2}$
AP3	AP2	15	1	1	5	1	8

Table 30: Control group request set-up.

G.3 Result structure

To best read and check the model output, they are oriented and presented from the aircraft's perspective. As shown in the Table 31, it shows that at time t_d, aircraft k takes off from airport i node a (airport i_a) and arrive at airport j node b (j_b) at time t_a with q passengers on board.

AC	From	Dep. time	To	Arr. time	#PAX
k	a_i	t_d	b_j	t_a	q

Table 31: Example of a line of output

Moreover, according to the need, the result can be arranged as a timeline in general, or it can be classified as timelines per aircraft to track each of their trajectory during the operation.

As a reference, the model output of the control group is:

AC	From	Dep. time	To	Arr. time	#PAX
2	3_2	1	1_1	4	15
2	1_1	5	3_2	8	0

Table 32: Output of the control group.

G.4 Scenario test on objective function

The first criterion is to test whether the system could generate the minimal cost operational plan with the given fleet and environment. As can be noticed from the objective function, there are 3 parts contributing to the total cost: operational cost, penalty of passenger spillage and extra aircraft operating time that exceeds the daily margin.

G.4.1 Operational cost minimisation

In general, operational cost can be reduced in two ways: choosing the shortest path or selecting the aircraft with lowest cost during the operation.

Shortest path (Scenario 1) To test the shortest path scenario, the flight leg between AP3 and AP2 has changed to be unidirectional, which forces the aircraft to return to its hub from AP2 to AP3 via AP1. In this case, extra cost will appear due to the extra distance travelled.

From	To	Direction
AP1	AP2	2
AP1	AP3	2
AP3	AP2	1

Table 33: Map for scenario 1.

In this case, by keeping every other parameters the same, the direction from AP3 to AP2 is changed to 1 in Table 27. And the resultant output of the model becomes:

Comparing this result with Table 32, it can be noticed that aircraft 2 has chosen a different route when flying back from AP 2 to AP 3. This circumstance proves that the aircraft will try to follow the possible shortest path during the operation.

Minimal operational cost (Scenario 2) On the other hand, when there are two aircraft available at the same place with different cost per unit distance, the system should be able to consider the one with lower cost to

AC	From	Dep. time	To	Arr. time	#PAX
2	3_2	1	2_1	4	15
2	2_1	5	1_1	7	0
2	1_1	8	3_2	10	0

Table 34: Output of scenario 1

operate. In some situations, two aircraft are located at different places, however there is still a possibility that the one with lower cost will be chosen if the saved cost per unit distance can compensate the extra distance.

The resultant output can be seen from Table 35. AC 1 is chosen for the operation rather than AC 2, which managed to choose a better solution with lower operational cost.

AC	From	Dep. time	To	Arr. time	#PAX
1	3_2	1	2_1	4	15
1	2_1	5	3_2	8	0

Table 35: Output of scenario 2

G.4.2 Passenger spillage minimisation (Scenario 3)

As the main mission of the UNHAS is to transfer officials and staff from place to place, passenger spillage is particularly not preferred. Minimal or no passenger spillage of the daily demand guarantees transport efficiency and timeliness.

AC	From	Dep. time	To	Arr. time	#PAX
1	1_2	0	3_1	2	0
1	3_1	3	1_1	5	15
1	1_1	6	2_1	8	15
1	2_1	9	1_2	11	0
2	3_2	1	1_1	3	3
2	1_1	4	2_1	6	3
2	2_1	7	3_2	10	0

Table 36: Output of scenario 3

G.5 Scenario test on aircraft characteristics

The second category to consider is the constraints of aircraft itself. Due to the technical limitation and service requirements, an aircraft cannot fly freely and continuously throughout the time. Two aspects are discussed here: the aircraft range limitation that hinders it to fly to anywhere and the time limitation at the hub restricts its departure and arrival time.

G.5.1 Aircraft range limitation (Scenario 4)

The range limitation of the aircraft restricts it from flying freely around airports, this is mainly due to the limited size of the fuel tank and other aircraft operational regulations. At certain locations, the aircraft needs to be refuelled in order to continue the operation.

In order to testify this constraint, the range of the aircraft is varied to stimulate a different flight route compared with the one from the control group. By decreasing the range of the aircraft, it is not possible to fly certain flight legs and the aircraft has to change route in order to reach the destination with multiple stages.

As can be seen from the result compared with the control group: the aircraft is not able to fly directly from AP 3 to AP 2 due to the range limitation but it has to take a longer route to avoid the long flight leg.

G.5.2 Aircraft time limitation (Scenario 5)

As stated by the lease contract, all leased aircraft are required to return to their hubs at the end of daily operation. Therefore, each deployed aircraft need to consider the return time during the planning and some destinations further from the hub will not be considered.

AC	From	Dep. time	To	Arr. time	#PAX
1	1_2	0	3_1	2	0
1	3_1	3	1_1	5	15
1	1_1	6	2_1	8	15
1	2_1	9	1_2	11	0

Table 37: Output of scenario 4

To prove the effectiveness of this constraint, a situation is created where the return time of the aircraft is brought forward, and therefore the aircraft is not able to accomplish the mission due to the insufficient time to return to its hub.

AC	From	Dep. time	To	Arr. time	#PAX
1	1_2	2	2_1	4	15
1	2_1	5	1_2	7	0
2	3_2	2	1_1	4	15
2	1_1	5	3_2	7	0

Table 38: Output of scenario 5

G.6 Scenario test on requests

The scenario tests on requests focuses on how aircraft manage the deliveries of passengers. Aircraft have to pick up and delivery the passengers on time and as much as possible. Due to the limitation of the seats on board, single request may be fulfilled by multiple aircraft or multiple times during the day.

G.6.1 Request time limitation (Scenario 6)

In real-time operation, passenger transportation must be carried within a certain time interval and fit to the need and effectiveness. In addition, each aircraft must return to its hub at the end of the day, which may results in the situation where a request cannot be operated by certain aircraft.

Originally, the request can be fulfilled successfully. However, in some occasions, an error can be discovered in the timeline of the plan. As can be seen from the initial model result of scenario 6 in Table 39: AC 1 has picked up the passengers from AC 2 at AP 1 before AC 2 has arrived at AP 1. This disorder in time is not desired and not permitted. Therefore the timeline of the daily operation need to be checked and necessary row generation is performed on the original model.

AC	From	Dep. time	To	Arr. time	#PAX
1	1_2	0	2_1	2	15
1	2_1	3	1_2	5	0
2	3_2	4	1_1	6	15
2	1_1	7	2_1	9	0
2	2_1	10	1_2	12	0
2	1_2	13	3_2	15	0

Table 39: Initial output of scenario 6

After implementing the timeline checking process, no operation is executed based on the model output of scenario 7, which meets the expectation of the result.

G.6.2 Request split to multiple aircraft (Scenario 7)

For request that has a large amount of passengers, it may not be able to achieved by a single aircraft. The overall efficiency can be improved when multiple aircraft cooperate to finish the demand.

G.6.3 Request cannot be fulfilled directly (Scenario 8)

Similar to the last condition, when there are limited aircraft to operate a certain flight route, passengers of the same request may be split to several groups and be transported per group by the aircraft. It is an efficient solution when the journey is short and if the assistance of another aircraft is impossible or much more expensive.

AC	From	Dep. time	To	Arr. time	#PAX
1	1_2	0	3_1	2	0
1	3_1	3	1_1	5	15
1	1_1	6	2_1	8	15
1	2_1	9	1_2	11	0
2	3_2	4	2_1	7	15
2	2_1	8	3_2	11	0

Table 40: Output of scenario 7

AC	From	Dep. time	To	Arr. time	#PAX
2	3_2	1	2_1	4	15
2	2_1	5	3_1	8	0
2	3_1	9	2_2	12	15
2	2_2	13	3_2	16	0

Table 41: Output of scenario 8

G.6.4 Flight transfer (Scenario 9)

For the situation where passengers are travelling from/to remote places. In this case, it is sometimes unnecessary to issue a direct flight for them, but it is more economic to arrange their route into multiple segments and allocate them into flights that commuting between places with large demands.

AC	From	Dep. time	To	Arr. time	#PAX
1	1_2	0	2_1	2	15
1	2_1	3	1_2	5	0
2	3_2	1	1_1	3	15
2	1_1	4	3_2	6	0

Table 42: Initial output of scenario 9

As can be noticed, the initial result has the same timeline error as mentioned in section G.6.1. After timeline check procedure and row generation, the model is solved again with a valid and feasible solution as shown in Table 43:

AC	From	Dep. time	To	Arr. time	#PAX
1	1_2	5	2_1	7	15
1	2_1	8	1_2	10	0
2	3_2	1	1_1	3	15
2	1_1	4	3_2	6	0

Table 43: Final output of scenario 9 after timeline check.

H UNHAS Scenario Result Comparison

	Origin	Destination
Fokker 50	Juba	Wau
	Wau	Aweil
	Aweil	Rumbek
	Rumbek	Juba
Dash 8_1	Juba	Rubkona
	Rubkona	Juba
Dash 8_2	Juba	Rumbek
	Rumbek	Juba
	Juba	Malakal
	Malakal	Juba
	Juba	Yambio
	Yambio	Juba
Dornier 228	Juba	Pibor
	Pibor	Juba
Cessna 208_1		
Cessna 208_2	Juba	Bor
	Bor	Pibor
	Pibor	Bor
	Bor	Juba
	Juba	Yida
	Yida	Juba
Cessna 208_3	Juba	Yida
	Yida	Rubkona
	Rubkona	Mankein
	Mankein	Juba
Cessna 208_4		
Cessna 208_5		
MilMi8_1	Juba	Mingkaman
	Mingkaman	Juba
MilMi8_2		
Cessna 208_1R		
Cessna 208_2R	Rumbek	Agok
	Agok	Rumbek
MilMi8_1R		
MilMi8_2R	Rumbek	Koch
	Koch	Leer
	Leer	Rumbek

Table 44: Flight schedule for requests on 13/04/2015, solved by Niemansburg

AC	From	Departure	To	Arrival	PAX
Fokker 50	JUB	00:42	WAU	01:21	17
Fokker 50	JUB	00:42	WAU	01:21	6
Fokker 50	JUB	00:42	WAU	01:21	27
Fokker 50	WAU	02:21	AWL	02:31	17
Fokker 50	WAU	02:21	AWL	02:31	6
Fokker 50	WAU	02:21	AWL	02:31	20
Fokker 50	WAU	02:21	AWL	02:31	2
Fokker 50	AWL	03:31	RUM	03:57	14
Fokker 50	AWL	03:31	RUM	03:57	2
Fokker 50	AWL	03:31	RUM	03:57	6
Fokker 50	AWL	03:31	RUM	03:57	20
Fokker 50	AWL	03:31	RUM	03:57	2
Fokker 50	RUM	05:33	JUB	05:56	4
Fokker 50	RUM	05:33	JUB	05:56	15
Fokker 50	RUM	05:33	JUB	05:56	20
Dash 8_1	JUB	00:54	MAK	02:30	35
Dash 8_1	JUB	00:54	MAK	02:30	2
Dash 8_1	MAK	03:30	RUM	04:03	2
Dash 8_1	MAK	03:30	RUM	04:03	16
Dash 8_1	RUM	05:39	JUB	06:06	14
Dash 8_1	RUM	05:39	JUB	06:06	1
Dash 8_1	RUM	05:39	JUB	06:06	1
Dash 8_1	RUM	05:39	JUB	06:06	1
Dash 8_1	RUM	05:39	JUB	06:06	20
Dash 8_2	JUB	00:18	AGOK	01:10	11
Dash 8_2	JUB	00:18	AGOK	01:10	25
Dash 8_2	AGOK	02:10	RUB	02:22	9
Dash 8_2	AGOK	02:10	RUB	02:22	25
Dash 8_2	RUB	03:22	JUB	04:08	9
Dash 8_2	RUB	03:22	JUB	04:08	15
Dornier 228	JUB	00:12	YAM	00:40	12
Dornier 228	YAM	01:40	JUB	02:08	15
Cessna 208_1	JUB	00:24	RUM	00:53	3
Cessna 208_1	JUB	00:24	RUM	00:53	2
Cessna 208_1	JUB	00:24	RUM	00:53	5
Cessna 208_1	RUM	01:59	JUB	02:28	0
Cessna 208_2	JUB	00:36	RUB	01:27	1
Cessna 208_2	JUB	00:36	RUB	01:27	3
Cessna 208_2	JUB	00:36	RUB	01:27	6
Cessna 208_2	RUB	02:27	YIDA	02:36	1
Cessna 208_2	RUB	02:27	YIDA	02:36	6
Cessna 208_2	RUB	02:27	YIDA	02:36	2
Cessna 208_2	YIDA	03:36	MNK	03:52	1
Cessna 208_2	YIDA	03:36	MNK	03:52	2
Cessna 208_2	YIDA	03:36	MNK	03:52	5
Cessna 208_2	MNK	04:52	YAM	05:41	5
Cessna 208_2	YAM	06:41	JUB	07:15	5
Cessna 208_2	YAM	06:41	JUB	07:15	5
Cessna 208_3	JUB	00:30	YIDA	01:28	10
Cessna 208_3	YIDA	02:28	BOR	03:14	5
Cessna 208_3	BOR	04:14	JUB	04:28	5
Cessna 208_3	BOR	04:14	JUB	04:28	5
Cessna 208_5	JUB	00:48	BOR	01:02	6
Cessna 208_5	BOR	02:02	PIBR	02:19	1
Cessna 208_5	BOR	02:02	PIBR	02:19	8
Cessna 208_5	PIBR	03:19	JUB	03:46	1
MilMi8_1	JUB	00:06	PIBR	00:55	16
MilMi8_1	PIBR	01:55	JUB	02:34	9
MilMi8_2	JUB	00:00	MINGK	00:19	15
MilMi8_2	MINGK	01:19	JUB	01:37	7
Cessna 208_1R	RUM	01:53	KOCH	02:13	3
Cessna 208_1R	RUM	01:53	KOCH	02:13	2
Cessna 208_1R	KOCH	03:13	LER	03:16	2
Cessna 208_1R	KOCH	03:13	LER	03:16	1
Cessna 208_1R	LER	04:16	RUM	04:33	1
Cessna 208_1R	LER	04:16	RUM	04:33	5

Table 45: Flight schedule for requests on 13/04/2015, solved by tabu search method

	Fokker 50	**Dash 8_1**	**Dash 8_2**	**Dornier 228**	**Cessna 208_1**	**Cessna 208_2**	**Cessna 208_3**	**Cessna 208_4**
Niemansburg	04:48	03:12	08:01	01:40	00:00	07:16	04:58	00:00
TS Model	05:14	05:12	03:50	01:56	02:04	06:39	03:58	00:00
Difference	00:26	02:00	-04:11	00:16	02:04	-00:37	-01:00	00:00

	Cessna 208_5	**MilMi8_1**	**MilMi8_2**	**Cessna 208_1R**	**Cessna 208_2R**	**MilMi8_1R**	**MilMi8_2R**	**Total**
Niemansburg	00:00	01:30	00:00	00:00	02:10	00:00	02:30	12:05
TS Model	02:58	02:28	01:37	02:40	00:00	00:00	00:00	14:36
Difference	02:58	00:58	01:37	02:40	-02:10	00:00	-02:20	02:31

	number of aircraft deployed	**Average aircraft utilisation time**	**Standard deviation**
Niemansburg	9	02:24	0.11367
TS Model	11	02:34	0.08794
Difference	+2	+00:10	-0.02572

Table 46: Fleet utilisation time comparison for requests on 13/04/2015

II

Literature Study

1
Introduction

1.1. Background

This thesis report presents the design of optimisation model for the UNHAS network problem and the evaluation of its performance.

The United Nation Humanitarian Air Service (UNHAS) is the world's main transporter of humanitarian personnel and aid, which provides aviation logistics to places of natural disasters and emergencies around the world. It is a non-commercial operator, which provides aviation service to not only the staff of the World Food Program (WFP)[6], but also other UN agencies and non-governmental organisations (NGOs) to deliver supplies to where it is needed[28]. The project aims to provide the air transport service to deliver life-saving assistance based on the request and transport humanitarian workers to where they are needed the most.

The UNHAS network optimisation problem is one type of vehicle routing problems. To be more specific, it is a capacitated VRP (CVRP), which is NP-hard and time-consuming to solve (Garey and Johnson[40]). Contrary to the general strategy of a commercial aviation company, which focuses on cost minimisation, profit maximisation and optimal utilisation of a certain fleet type (Abara[8]), the UNHAS emphasises more on cost minimisation of aircraft lease and efficiency maximisation when planning the flight schedule. Similar research and discussion have been made in other humanitarian programmes. Eftekar et al.[34] researched the real-life operations of the International Committee of the Red Cross (ICRC)[3] in order to seek an optimal policy that can be implemented for the operations. Liu et al.[67] focused on the epidemic logistics and developed the medical resource allocation models under different level of supplies with the help of epidemic dynamic method.

Due to this difference in characteristic, the problem cannot be directly solved by the regular aircraft assignment model, but few adjustments are needed in order to add relevant criteria into the model. For example, the total cost still need to be minimised, and due to the speciality of this mission, there is no revenue part but the number of transported passengers and the proportion of served passengers need to be maximised.

The report is structured as follows: All different variants of vehicle routing problem (VRP) with respect to their emphasis, complexity and structure are first introduced in chapter 2, this provides an overview of popular VRP variants. Existing heuristics are introduced along with their characteristics and suitable VRP scenarios in chapter 3. An additional emphasis of the dynamic VRP and dynamic and stochastic VRP is present in section 4.1 to have a thorough study of its characteristics and the experiences from previous research to inspire the methodology development in the UNHAS scenario.

1.2. Research Objective and Context

The first stage of the problem has been established by S.P. Niemansburg during his master thesis project in 2019[80], which solved the humanitarian flight optimisation in South Sudan on a daily timescale. There are few limitations in his model:

1. The model is unable to split requests.
2. The planning horizon of the model is short.

3. The recapture of spilt passengers are not considered in the planning.
4. No anticipation of the possible future demands.

In addition, the model needs to be improved further with new requirements: decision support tool development for the tasking officers, and the consideration of minimum guaranteed hours (MGH) for the contracts.

In conclusion, the master thesis project this year focuses on the following fields:

1. Optimisation by considering past and future demands.
2. Routing and scheduling optimisation with a longer planning horizon.
3. Unlimited dividing of requests over flights.
4. Decision support tool development.
5. Implementation of minimum guaranteed hours (MGH) requirements.

The points listed above brought new challenges and possibilities to explore alternatives to solve the problem. Scheduling over a planning horizon and consideration of the MGH requirements lead to VRP with time windows (VRPTW). The heuristic method is one of the many options to discover relatively good feasible solution within the required time limit. Coello et al.[20] introduced multiple metaheuristic methods to solve multi-objective problems. In their study, the tabu search algorithm is selected as the most suitable algorithm for the vehicle routing problems, which is further explained in section 3.4. Moreover, the consideration of past and future demands requires anticipatory algorithms and predictive routing strategies. They are considered to guarantee sufficient pre-planning when confronting uncertain or unknown demand from the passengers, which is introduced in section 4.1.

Therefore, the main research objective of this thesis is:

"To achieve a cost-efficient flight scheduling of the UNHAS concerning the operational and safety constraints to the non-commercial humanitarian setting by means of a decision support tool that improves the demand satisfaction with the help of metaheuristic method"

To achieve the design objective, a theoretical research question needs to be reformulated to provide a clear specification of the aspects that need to be considered during the project:

"How to develop a decision-making model that provides a cost-effective flight schedule by considering both the past and future demand?"

In order to answer the research question, the following sub-questions are framed:

1. Which type of VRP is this problem?
2. Which aspects to be included in the objective function?
3. What are the shortcomings of the existed algorithm?
4. Which metaheuristic is chosen for the model?
5. How to make a fast decision when choosing the most cost-effective aircraft?
6. How to determine the recapture rate of the previously spilled passengers during each flight?
7. How to sufficiently anticipate future demand?
8. How many days the time window should be?
9. How much margin should be reserved per day for possible spilled passengers?
10. How to meet the requirements of minimum guaranteed hours (MGH) from the contracts?

Theoretical Content/Methodology

The most challenging element in this project is the uncertainty of passenger demands, which is the most important input the optimisation model needs to generate the optimal flight plan. Due to the fact that the UNHAS does not have its own fleet, it operates the transportation service by relying on contributions from voluntary donors or nations and other UN funding. It charters and operates aircraft to serve the poor and landlocked areas in the world. In most cases, the passengers are personnel from the UN agencies and NGO staffs with limited budgets, which means that the UNHAS need to make an effective and cost-efficient flight plan. (Dorn and Cross [28]) The uniqueness of the work in these regions results in a frequent and quick change in demand, which requires the UNHAS to react fast enough to handle the requests.

Based on the classification by Toth and Vigo regarding the characteristics of the VRPs, this project is considered as a dynamic and stochastic capacitated VRP (CVRP), which contains either partial or all input data that are stochastic (e.g. forecasts, range values). It has a strong motivation to utilise and integrate the available information to anticipate future events in the solution process.[111]

Due to the fact that the exact solution procedure of large-scale VRP is too time-consuming and sometimes the problem is not solvable. The hypothesis to be proven in this project is that the computation time of the humanitarian flight service scheduling can be shortened with a relatively good solution by implementing the metaheuristic method into the model. The methodology is designed by combining the tabu search method and anticipatory algorithm to improve the previous strategy. To pursue maximal utilisation of the aircraft, it considers the past and future demand in the model, and it applies unlimited division of the requests over flights. The input is the daily demand matrix based on the passengers' request, which contains the origin and destination pairs of each passenger. The output of the model is a routing and time schedule of each aircraft as well as the percentage of demand satisfaction of each day.

Experimental Set-up

Overview

As a research topic that focuses on developing the new algorithm for the problem, there is no physical experiment throughout the project. However, simulation experiments are taking place within the programming environment.

For linear programming, there are numerous options available for the optimisation software, such as CPLEX[2] and Gurobi[7]. Due to the former programming and learning experience during previous group projects and assignments, CPLEX is chosen with respect to the knowledge and familiarity with it.

For coding environment, Python[5] is chosen instead of MATLAB[4] as the Application Programming Interface (API). Compared with MATLAB, which can only be used with a license, Python could realise almost all the functions in MATLAB with limited restrictions. Therefore using Python to design the mathematical model can ensure broader accessibility for colleagues, and the development process can be more identical to the coding environment in projects during the future career.

Development process

The book Introduction to Operations Research by Hillier and Lieberman[49] is considered as the main literature throughout the project. It provides the fundamental theoretical knowledge of operations research, multiple optimisation models for a variation of circumstances and a number of examples to study.

The preliminary model of the VRP has been developed by Niemansburg[80] under the same project, which managed to generate a near-optimal solution of single-day planning within an acceptable amount of time. On top of his model, the planning needs to consider both the past and possible future demands, by means of considering the past demand history, the spilled passengers and the anticipated future requests. The requests are preferably divided over flights during the planning in order to have better distribution and allocation over the available seats on-board. Moreover, the previous model itself needs to be improved as well by implementing metaheuristic technique, which shortens the computational time to obtain a solution that is close to the global optimum.

Based on the literature study and discussions with the supervisor, the tabu search is chosen as the method to improve the model. In order to implement it steadily to fit the project model, an experiment takes place in the first place. A simpler aircraft routing problem needs to be designed, which contains less type of aircraft and fewer destinations to consider. This simplified scenario is used to try the tabu search method as well as the dynamic programming technique, and the outcome will be evaluated to assess the algorithm performance. If the outcome is unsatisfying, the algorithm will be discussed with the supervisor and the necessary adjustment will be applied to improve the model. When the model is fully tested with the acceptable result, the algorithm is applied to the real-life data and scenario, followed by performance evaluation, verification and validation.

Results, Outcome and Relevance

The outcome of this project is expected to be a mathematical model that can be applied to real-life data with satisfactory performance. Verification and validation of the model can be done by comparing the model results with the outcome of the Niemansburg's model and the manual scheduling of experienced flight planners. Sensitivity analysis of the model will be carried out in order to test the model, and the necessary improvement will be implemented based on the analysis result.

In addition, due to the fact that one of the goals of the project is to provide decision support tool for the tasking officers, the individual pair of an input and its resultant flight assignment from the model can be summarised for further analysis to find the possible relations between each certain occasion and its optimal choice. The margin of error of the relations need to be tested with a number of test data and its reliability needs to be evaluated. Ideally, the summary of the correlations can assist the tasking officers to make faster decisions during the planning and therefore improve the efficiency of the UNHAS operations.

Based on the model outcome, difficulties faced during the analysis and recommendation of future development can be illustrated to improve the model and find new aspects to emphasise on.

2

Vehicle Routing Problem

2.1. Overview

The routing and scheduling optimisation of a fleet of aircraft dedicate to generate a fleet plan, which contains a selection of routes to be flown by particular aircraft as the output. The plan achieves numerous requirements by satisfying certain constraints by the company/organisation. The requirements vary from problem to problem, which are influenced by the goals set by the company or organisation.

For commercial airlines, one of the goals of the flight scheduling is usually maximising the profit and minimising the cost, because commercial airline companies seek maximum revenue during the operation. However, based on the different visions of a variety of companies and organisations, the goal of the routing and scheduling optimisation varies. Despite the uniqueness of each optimisation problem, they can all be generated as the vehicle routing problem.

2.2. Travelling Salesman Problem (TSP)

The vehicle routing problem (VRP) is a combinatorial optimisation and integer programming problem. It seeks the optimal planning of the vehicle routing and delivery plan to satisfy the customers' demand to its maximum. It was first introduced in 1959 by Dantzig and Ramser[24], which generalise the famous travelling salesman problem (TSP)[37].

The origin of this sort of problems is uncertain. There were many informal discussions among mathematicians over the problem during the meeting for many years, but little in the form of scientific articles delivered in the mathematical literature.[112] In 1954, Dantzig et al. stated the definition of the TSP for the first time. The TSP is described as a problem to find the shortest route from a certain city, visiting each one of a specified cluster of cities, and then returning to the place of departure.[25] The article presents a problem of finding the optimal sequence to serve 49 cities in the USA with a minimum total distance travelled. In this study, the problem does not consider the capacity and the mission of the salesman, it only requires that all cities except the origin are visited exactly once before returning to the origin. The binary decision variables x_{ij} are used to determine the moving direction from city i to city j. Due to the low complexity of the problem and only a few constraints to consider, the TSP can be solved manually by matrix operations.

On the other hand, the assignment and transportation problem was introduced by Hitchcock in 1941[51]. The Hitchcock distribution problem has two cases: the transportation case finds the transportation plan to transmit a certain amount of carriers from a set of old stations to a set of new stations; the assignment case seeks the plan to assign numerous men to a group of jobs. The decision variables x_{ij} in both cases are all positive integers, which illustrates the preliminary version of integer linear programming (ILP) problem.

In 1959, Dantzig and Rasmer introduced the "Truck Dispatching Problem", which simulates the operation of a delivery truck and focuses on scheduling the optimal order to travel through each of n given points once (P_1 to P_n).[24] Compared with the study in 1954, this problem considers the maximum capacity of the delivery truck as well as the demand required at each destination. Moreover, the terminal point P_0 can be visited multiple times to supply the truck. In this paper, a linear programming approach was first introduced and the problem was solved with a near-optimal solution by multiple stages of aggregations. In 1964, Clarke and Wright improved the method of Dantzig and Rasmer by using a greedy approach to solve the same truck dispatching problem with better solution[19].

Irnich et al. defined the term of the family of VRP: A variety of transportation requests and a group of vehicles are provided as input, where the problem is to determine a set of vehicle routes to operate requests with the fleet at minimum cost in such a way that each vehicle handles each request sequentially[56].

After World War II, Dantzig introduced the simplex method to solve the linear programming problem in a mathematically tractable way[78]. Based on this method, every VRP model can be formed mathematically as follow:

$$Minimise/Maximise \qquad \mathbf{c'x} \tag{2.1}$$

$$\begin{aligned} subject\ to \qquad \mathbf{A_1x} &\leq \mathbf{b_1} && (2.2)\\ \mathbf{A_2x} &= \mathbf{b_2} && (2.3)\\ \mathbf{A_3x} &\geq \mathbf{b_3} && (2.4)\\ \mathbf{x} &\geq \mathbf{0} && (2.5) \end{aligned}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{A_1} \\ \mathbf{A_2} \\ \mathbf{A_3} \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} \mathbf{b_1} \\ \mathbf{b_2} \\ \mathbf{b_3} \end{bmatrix} \tag{2.6}$$

and $\mathbf{c'}$ is the cost coefficient matrix of all the decision variables $\mathbf{x}$[77].

2.3. Capacitated Vehicle Routing Problem

Among different versions of the VRP, the Capacitated Vehicle Routing Problem (CVRP) is the one that has been most discussed and studied. In the CVRP, a single depot denoted as 0 handles numerous transportation requests to a group of other points (customers), $N = 1, 2, ..., n$. Each customer's demand from the depot is given by $q_i \geq 0$, $i \in N$. The fleet of K vehicles at the depot are all identical with the same capacity $Q > 0$ and operational cost. Each vehicle deals with the request of a cluster of customers $S \subseteq N$ from the depot, visits each customer in S sequentially and eventually returns to the depot. The travel cost from point i to j is defined as c_{ij}.

In general, the CVRP includes two independent assignments:

- The segmentation of the customer set N to numerous groups $S_1, ..., S_{|K|}$.
- The sequence within $0 \cup S_k$ for each vehicle $k \in K$ to visit.

As can be noticed, the second assignment is identical to the TSP scenario. The above two assignments interfere each other because the size of each cluster determines the routing within it, and the routing schedule determines the cost of each cluster. For complete graph $G = (V, A)$, where in-arcs and out-arcs of S are denoted as $\delta^-(S) = (i,j) \in A : i \notin S, j \in S$ and $\delta^+(S) = (i,j) \in A : i \in S, j \notin S$ to represent the flow going in and out a certain sub-set S. $r(S)$ represents the minimal routes a vehicle needs within each sub-set S, which can be calculated by resolving the bin packing problem[68]. The directive CVRP model in the traditional notation can be shown as follows:

$$minimise \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{2.7}$$

$$s.t. \sum_{j\in\delta^+(i)} x_{ij} = 1 \qquad \forall i \in N, \tag{2.8}$$

$$\sum_{j\in\delta^-(j)} x_{ij} = 1 \qquad \forall j \in N, \tag{2.9}$$

$$\sum_{j\in\delta^+(0)} x_{0j} = |K| \tag{2.10}$$

$$\sum_{(i,j)\in\delta^+(S)} x_{ij} \geq r(S) \qquad \forall S \subseteq N, S \neq \emptyset, \tag{2.11}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A. \tag{2.12}$$

Based on this fundamental model, the CVRP can be extended to multiple categories based on different requirements and characteristics to be considered in the application.

2.4. VRP categorisation

Researchers put more emphasis on studying the characteristics of a variety of VRP models. The international research community showed high interest in different categories of the VRP models regarding their complicity and real-life relevance. Scholars from the industrial world joint the academic community on the categorisation of the VRP problems[56]. Eksioglu et al. studied a broad span of the VRP literature and classified them based on their disparate characteristics and aspects considered, such as the applied method, time horizon and quality of information[35]. This classification is detailed enough for scientific research, but it is not practical to capture the dominant aspects when analysing a given VRP problem. A taxonomy of the VRP problems is done from six main angles[56]:

- Network characteristics
- Type of transportation requests
- Intra-route constraints
- Fleet types
- Inter-route constraints
- Objectives

These aspects are considered to be the fundamental characteristics of the VRP, which determine the specific scenario considered in the problem and the structure of the model.

2.5. Network characteristics

The network layout of the VRPs varies a lot depends on the real-life application as well as the operating environment each problem is considering. It determines how the environment is possibly visualised as well as the map where the optimisation problem is formulated. Mathematically, the network characteristics highly affect the scale of the problem and the size of the decision variable matrix. They can also provide a practical guide to improve the strategy to solve the problem in order to make the optimisation model more efficient and adaptive to the specifications.

In most CVRP models, the tasks are assigned to numerous vehicles to transport goods or passengers from one location to another. The delivery points or stations are typically modelled as vertices in the graph, and the corresponding VRP are named as node routing problem. In contrast, for street sweeping problem[102] and aircraft taxiing route problem [57], the problem is constructed based on the route segments and links between a certain amount of node pairs. These type of problems emphasise on the movements or tasks along edges or arcs, therefore they are denoted as arc routing problems (ARP)[30]. A mixture of these two types of problems can be possible, considering the tasks or demands on both vertices and edges, which leads to general routing problems (GRP)[82].

The other aspect to consider in the graph is the symmetry of the problem. For symmetric problem, the travel cost between a pair of nodes i and j are identical ($c_{ij} = c_{ji}$). An undirected graph can be formed for symmetric problems. On the other hand, for asymmetric problems, the movement from one node to the other may be one-way or the relevant travel cost is influenced by the direction ($c_{ij} \neq c_{ji}$). The underlying graphs for asymmetric problems can by purely directed, mixed[36] or windy[46][72].

The last point to consider in the network is the granularity of the data. The ARP and GRP focus on simulating the actual layout of the street segments in the graph, whereas in the VRP, an edge between two vertices may represent a large number of sequential segments connecting two points in the real-world setting. A finer granularity is beneficial for the ARP and GRP problems to find the optimal routes, but coarse granularity is more suitable for VRP to avoid redundancy in the model and large computational time. The distance and travelling time between two vertices of the VRP is usually calculated based on the shortest path from one point to another. Because of these characteristics, the VRPs are generally suitable to solve the large-scale problems in the real-life applications, such as the famous travelling salesman problem[25] and truck dispatching problem[24], both of which consider the transportation across the whole country.

In general, the UNHAS consists the transportation of passengers between a number of cities (airports) among the country as well as the flight arc and ground arc between each time points, which is a node routing problem that considers the connections between two types of vertices. An undirected graph will be formed as the flight path between two cities is two-directional and the flight time in each direction are considered to be identical. Moreover, a coarse granularity is employed to only consider the flow between cities, and the aircraft taxiing procedure is simply included in the landing and take-off time (LTO) for each aircraft.

2.6. Type of transportation requests

The characteristics of the transportation requests are closely related to the VRP, which determines the 'theme' throughout the problem. the requests are

In contrast to the tradition delivery problem, which transports a variety of goods from the depot(s) to customers, the procedure and routes can be reversed to create another type of problem. By collecting certain products from the customers to certain location(s), the collection or pickup problems are formed. The related routing problems can be found at the initial steps of a certain supply chain or logistics operation, such as the raw-milk collection[98] and waste collection[61].

Besides the pure pickup or delivery problems, a mixture of both cases can occur to shape distinct variants of VRP. A typical VRP is the VRP with backhauls (VRPB), which contains two steps during the operation. First, multiple deliveries need to be carried out to the so-called linehaul customers. The vehicle is completely unloaded after the mission, the collection tasks are followed to collect the goods from backhaul customers[110]. In VRPB, backhaul customers are always visited after all linehaul customers have been served. The two steps are not intermingled with each other and therefore no movement between the backhaul customers and the linehaul customers is permitted. A more complex version of the VRPB is the mixed VRPB (MVRPB), where the backhaul customers are allowed to be visited when serving the linehaul customers[114]. In MVRPB, as the vehicles are allowed to collect and unload products alternately, the occupancy of the space is not varying monotonically, therefore the capacity constraint of the vehicle needs to be checked continuously for each segment travelled.

To extend the problem further and consider the real-world application, each customer can request for both collection and delivery in the problem. In 1898, Min introduced the VRP with simultaneous pickup and delivery (VRPSPD) to plan the library material delivery and pickup routing around the Columbus metropolitan area in the USA[70]. Dethloff applied the VRPSPD approach to solve the reverse logistics problem of material recycling[26] and a heuristic method is developed by Montané and Galvão [73]. Since the load of the vehicle is a mixture of pick-up and delivery loads, capacity constraints need to be applied in order to avoid overloading. A relaxation of the VRPSDP is the VRP with divisible deliveries and pickups (VRPDDP), where the same customer can be visited once or twice to control the space occupancy in the vehicle. A reduction of the total operation cost is possible compared with the same solution of VRPSDP.

Regarding the number of depots and customers, the simplest pickup and delivery VRPs only consist point-to-point transports, where each request correspond to one collecting location and one delivery location. However, in reality, multiple pick-up and delivery requests are possible when forming the many-to-many VRP. In the background of the passenger transportation applications, the problem is also named as Dial-a-Ride Problem (DARP), which is suitable for schooling bus routing problem (SBRP) that design a route from an original point to the destination via multiple pick-up or delivery stops[79][84].

On the other hand, the transportation tasks are considered to be non-split by default, while some services can be split under different circumstances. When the demand is larger than the vehicle capacity, multiple visits of the same customer is inevitable and in some situations, division of the requests may result in a significant saving of the total cost and better utilisation of the vehicles. This typical characteristic leads to the introduction and study of split delivery VRP (SDVRP)[32][31]. For customers with continuous or regular demands of goods, such as the order of raw material by a factory, the supplier can concern a longer planning horizon to schedule the repeat delivery. The periodic VRP (PVRP) was first introduced by Beltrami and Bodin[12] in 1974 to plan the periodic garbage collection activity of New York city with 2-opt and 3-opt heuristics, and Cordeau et al.[23] developed the tabu search approach of the general PVRP.

As the delivery tasks can be split into several parts, so can the procedure of shipping tasks be split into multiple stages by a number of different vehicles. Combined shipments accomplish individual shipment task by using different vehicles to transport the goods from the supplier to the customer, via intermediate transfer spots or regional distribution centres. The two distinctive features of this problem are: 1) goods from different suppliers are generated or distributed at multiple centres before reaching the destinations, and 2) different mode of transport and vehicles are used at numerous segments. A common application of this strategy can be found in the logistics operation within a large area, such as the global courier service and supermarket supply chain, as well as the hub-and-spoke network structure for airline operation[69]. A study has been done by Song et al.[101] to plan the third-party consolidated distribution service for different suppliers in Hong Kong.

Traditionally, in the two-staged procedure, the planning of the route is followed by confirmation and acceptance of the requests from customers. Rather than believing all the delivery or collection tasks as mandatory, some of the delivery tasks can be temporarily ignored in certain conditions. Rejection of demand is possible due to the limitation of the fleet size, which can not fulfil all the requests within a certain amount of time in the area. One may also consider the unworthiness of accepting a certain task because of the relatively higher cost compared with the revenue. On the other hand, if the route optimisation and filtering of request can be executed simultaneously, there is an opportunity to acquire additional revenues concerning the traditional decision procedure. To solve this tricky situation, additional constraints containing the service levels and costs can be supplemented, penalty and reward can be set when a request is ignored or accomplished.

An important characteristic of the requests to consider is the uncertainty and variability in the system. Depending on the availability and detail of the information, such as the delivery location, quantity demanded and the number of requests within a certain time period, the routing problem can branch out two alternatives:

- A priori optimisation: When sufficient input data and relevant information are available, even with a degree of expected uncertainty, the problem is initially solved as a static problem with a preliminary routing plan. Then, based on the changes observed in the execution phase, the plan will be modified gradually to fit the variation.
- Dynamic optimisation: Changes and new information are observed in real-time operation. Decisions and execution plan are upgraded in parallel with the variation of the environment. This sort of operation requires strong technical support and live communication between the decision-makers (dispatchers) and the operators (drivers).

Based on the definition of Psaraftis[90], two essential dimensions of the inputs need to be considered when classifying the VRPs: evolution (static versus dynamic) and quality of information (deterministic versus stochastic). The combination of the two aspects results in four alternatives, from the simplest static and deterministic problem to the laborious and time-consuming dynamic and stochastic problem.

2.7. Intra-route constraints

The intra-route constraints are the keys to decide the feasibility of a route. All the constraints can be examined when a route or vertices sequence is determined, regardless of other routes. To be aligned with the UNHAS requirements, constraints of e.g. vehicle capacity, route length, multi-use of vehicles and planning horizon are the aspects that need to be studied and extended to the airline operational requirements.

One of the most important parameters to consider is the capacity of the vehicle. In order to fulfil the loading requirements in the real-world application, the vehicle capacity needs to be constrained to contain a limited amount of goods and overload is impossible at any moment. The general capacity constraint is illustrated by bounding the number of goods to be delivered at every dropping point the vehicle passes. For the general weighted graph of CVRP: $G = (V, E, c_{ij}, q_i)$, where V is the set of all vertices in the graph, $E = \{e = \{i, j\} = \{j, i\} : i, j \in V\}$ is the set of all undirected edges in the graph, c_{ij} is the edge cost for $\{i, j\} \in E$ and q_i is the demand at vertex $i \in V$. For homogeneous fleet K with the same capacity Q, the binary decision variable $x_{ij}^k \in \{0, 1\}$ represents that vehicle $k \in K$ deploy the arc $(i, j) \in A$, and $y_i^k \in \{0, 1\}$ indicates that vertex $i \in V$ is served with vehicle $k \in K$. The capacity constraint can be presented in Equation 2.13:

$$\sum_{i \in V} q_i y_i^k \leq Q \quad \forall k \in K. \tag{2.13}$$

The constraint can be extended further to adapt a more complicated scenario. For heterogeneous VRP, the individual capacities within the fleet are not identical to each other, therefore the capacity Q is replaced by Q^k for each vehicle $k \in K$. In the UNHAS scenario, passengers can get on or off the aircraft at the airport, which means that the loading and unloading of the aircraft occur at the same time and the UNHAS mission is considered to be VRPSPD. A similar problem can be found in the study of Min[70], where the vehicle capacity is checked at each vertex to avoid overload.

Another common type of constraints considered in the VRPs is the resource consumption on segments or edges. The meaning can be miscellaneous, energy or fuel consumption in particular, which can be denoted as the amount of energy used, distance travelled or the time spent on the edge. Consideration of the additional distance constraints in the CVRP results in the distance-constrained CVRP (DCVRP)[18]. The resource consumption between vertex $i \in V$ and $j \in V$ is denoted as t_{ij} and the distance constraints is shown in Equation 2.14:

$$\sum_{(i,j) \in A} t_{ij} x_{ij}^k \leq L \quad \forall k \in K \tag{2.14}$$

where $L > 0$ is the upper bound of route length. In aircraft assignments, the flight distant between two cities $\{i, j\} \in A$ is denoted as d_{ij}. The distance constraint only compares the total flight distance travelled with the maximum range of the aircraft. For heterogeneous fleet K, the maximum range L_{range}^k varies per aircraft type. The distance constraint is modified as:

$$\sum_{(i,j) \in A} d_{ij} x_{ij}^k \leq L_{range}^k \quad \forall k \in K \tag{2.15}$$

When the vehicle capacity Q^k is relatively small or the fleet size $|K|$ can not satisfy the large scale of demand, a feasible solution can be possibly found if each vehicle can be assigned to multiple routes over the planning horizon T. This kind of problem is regarded as VRP with multiple uses of vehicles (VRPM)[104]. For the UNHAS case, each aircraft is encouraged to be utilised thoroughly during the day to minimise the total amount of leased aircraft. The same aircraft may be employed to accomplish multiple requests continuously during the day. Therefore, for all possible route P^k for vehicle $k \in K$, the duration of each route is defined as $T_p, (p \in P^k, k \in K)$. Each vehicle is able to execute all the routes if Equation 2.16 applies.

$$\sum_{p \in P^k} T_p \leq T \quad \forall k \in K \tag{2.16}$$

Another aspect involved in most VRP variants is the time window constraints, which considers the travel, service and waiting times during the operation. The VRP with time window (VRPTW)[22] is an extension of the CVRP where the service of customer must be performed within a certain time interval. Mathematically, t_{ij} indicates the travelling time of arc $(i, j) \in A$, s_i^k is the service time of aircraft $k \in K$ at vertex $i \in V$ and time window $[a_i, b_i]$ indicates the earliest and latest starting time at vertex $i \in V$. The start time T_i^k for visit of $k \in K$ at vertex $i \in V$ is considered to be feasible when:

$$a_i \leq T_i^k \leq b_i \quad \forall i \in V, k \in K \tag{2.17}$$

where if the arc $(i,j) \in A$ is assigned to vehicle $k \in K$, $x_{ij}^k = 1$:

$$T_i^k + t_{ij} + s_i^k \leq T_j^k \quad \forall (i,j) \in A, k \in K \tag{2.18}$$

In the fleet assignment, s_i may represent a combination of all the time consumed at the airport (e.g. time for landing, taxiing, tasks at boarding gate and departure) and the time window $[a_i, b_i]$ may indicate the interval appointed by the airport to expect the arrival of the aircraft. Moreover, other constraints such as the runway length and balance at each node need to be considered as well when developing the optimisation model:

For the runway constraint, the aircraft is only allowed to visit the airport where the runway length L_{runway_i} is no shorter than the minimum required runway length l_{runway}^k:

$$l_{runway}^k x_{ij}^k = 1 \leq L_{runway_j} \quad \forall (i,j) \in A, k \in K \tag{2.19}$$

The other aspect to consider is the balance at the node, where the incoming and outgoing flight of each type at each vertex need to be equal, to ensure that there are no additional aircraft left at the airport. The basic airline fleet assignment model has been developed by Abara[8] in 1989, which applied the linear programming model to solve the problem. Rexing et al.[95] developed the model further by combining the airline fleet assignment with a time window, with ground arc considered. In the UNHAS case, all the aircraft are wet-leased, therefore the aircraft is required to return to its origin after the operation is completed. The balance constraint can be described as:

$$\sum_{j \in A, j \neq i} x_{ji}^k = \sum_{j \in A, j \neq i} x_{ij}^k \quad \forall i \in V, k \in K \tag{2.20}$$

2.8. Fleet types

The fleet is defined as a cluster of vehicles to accomplish certain tasks in the operation. The simplest type is the homogeneous fleet, where all vehicles in the fleet are considered to be identical to each other regarding the capacity, operational cost and speed, etc. Regarding the origin of the fleet, many VRPs only consider a single depot in the operation where all the vehicles start from the same location. In this section, a discussion about the multi-depot RP and heterogeneous VRP is illustrated.

Considering multiple depots or hubs in the model brings more complexity in the solving process. For homogeneous fleet initiating and finishing their routes at different depots, the problem is defined as the multiple depot VRP (MDVRP)[94]. In this type of problem, every vehicle may be assigned to have the unique starting and ending locations, and the vehicles are allocated to a fewer amount of depots. Example of these kinds of problem can be found by Nagy and Salhi[76] and Min et al.[71], and The capacity limitation may hinder the depot to hold a certain amount of vehicles and the depots may act as an intermediate supply station for the vehicle to continue further operation[107].

On the other hand, for heterogeneous or mixed fleet VRP (HFVRP)[10], the vehicles in the fleet differ in capacity, fixed costs and reachable locations. The fleet K is considered to be a combination of $|P|$ homogeneous vehicles subsets, where $K = K^1 \cup K^2 \cup \cdots \cup K^{|P|}$ and all vehicles $k \in K^p (p \in P)$ in each subset is considered to have the same capacity $Q^k = Q^p$, routing costs $c_{ij}^k = c_{ij}^p$, etc. Example of HFVRP can be found by research of Taillard[103], which considers a problem of $|K|$ types of vehicles with limited amount of n^k of each type, with total number of $|T|$ routes:

$$min \sum_{k=1}^{K} \sum_{j=1}^{m} c_{jk} x_{jk} \tag{2.21}$$

$$s.t. \sum_{k=1}^{K} \sum_{j=1}^{m} a_{ij} x_{jk} = 1 \qquad i = 1, \dots, n \tag{2.22}$$

$$\sum_{j=1}^{m} x_{jk} \leq n_k \qquad k = 1, \dots, K \tag{2.23}$$

$$x_{jk} \in \{0,1\} \qquad j = 1, \dots, m, \quad k = 1, \dots, K \tag{2.24}$$

where $a_{ij} = 1$ when the customer $i \in j^{th}$ tour of T, and 0 otherwise.

The UNHAS is considered to have multiple hubs and multiple different types of aircraft in the fleet, therefore it is considered to be a multiple depot and heterogeneous VRP to consider when establishing the mathematical model.

2.9. Inter-route constraints

In section 2.7, the properties listed determine whether one single route is feasible or not, regardless of other routes in the plan. In this section, aspects need to be considered in inter-route or global constraints are presented. In contrast with inter-route constraints, the inter-route constraints consider the influence of a combination of routes to the feasibility of the solution.

The elements to be considered in these constraints are typically related to the characteristics of the routes or the vehicles. For example, consideration of fairness in the problem, such as even assignment of workload among drivers. These constraints are considered as balancing constraints, which dedicates to treat each vehicle evenly. The street routing problem has been studied by Bodin et al.[13] to consider the balancing of workload for long haul truck routing problem.

The second aspect to be considered is the distribution of limited resource. In VRPs, the resource is commonly considered as the capacity to handle the incoming goods at the depot, such as the problem of mail collection from postboxes or parcels pickup from clients. Restriction of routes with certain characteristics is possible, such as routes with long distance, a large number of stops and late arriving time. Another aspect to be considered can be the limited number of docks or slots at the depot or hub. The limited capacity at the depot requires the staggered arrival of the vehicles to reduce the waiting times and avoid late start for the vehicles for further operation. An example can be found by Rieck and Zimmermann[96]. The feasibility of routes relies on the vehicle arrival time as well as the remaining amount to be handled at the depot before the cut off[48].

The third aspect to consider is the synchronisation issue, which considers the coordination of related vehicles or interdependent tasks. The first study of this type of constraint is done by Drexl[29] over the VRP with Multiple Synchronization constraints (VRPMS), which classifies the synchronisation with respect to task clustering, the order of operation, parallel movement, loading amount and resource utilisation.

2.10. Objectives

The objective function is the most essential element in the optimisation model, it provides the ultimate goal the problem is pursuing. In most cases, the VRPs are merely seeking the minimisation of the routing cost in the operation. The objectives may include multiple goals to consider. A discussion of single objective problem to the multi-criteria problem is illustrated to provide an overview of disparate VRP scenario.

2.10.1. Single Objective

Mathematically, as shown in Equation 2.1, the objective function normally consists the multiplication of two matrices: matrix c' consisting the cost and profit parameters and the other matrix x consisting all the decision variables. In the simplest situation, some of the elements in the routing cost matrix c' can be set to zero for irrelevant decision variables, or to a large number for elimination of infeasible or undesired edges using the big-M method. The objective function can be constituted by multiple elements, such as the variable routing costs c_{ij}^k and the fixed cost, as well as the profit or penalty components. However, the single objective can only be minimised or maximised in the model, therefore each component needs to be weighted wisely in the function.

In service industries, customer satisfaction is often a critical index to measure the operational performance, common applications can be found in goods delivery and transportation service. The cost component is relatively hard to measure, which is often related to the waiting time of the customer (passenger), which can be represented as $p_i \max(T_i - a_i)^+$, where weight p_i illustrates the importance of the individual customer i. a_i is the earliest possible service time and T_i is the actual service time[56]. Another real-world case for the latency objective is the humanitarian services, where only a limited fleet of vehicles are available, and the utmost goal is to provide assistance and aid to affected regions at the earliest. In addition, waiting time at the customers are often undesired when concerning the customer satisfaction or additional cost. These aspects can be contemplated by introducing the soft time-window in the model. An example can be found in the study of Heilporn et al.[47], where the total duration of every delivery request is minimised in the objective function.

2.10.2. Hierarchical Objectives

In contrast with a single objective, hierarchical objectives consider a variety of objectives that are conflicting with each other. Different goals are mutually influencing the optimal solutions of each other and therefore they can not be solved simultaneously. For example, in the Vehicle Routing Problem with Time Windows (VRPTW), the minimisation of the route length can not be achieved when the minimisation of the deployed vehicles is considered as well, since the utilisation of the vehicles and drivers leads to high fix costs with respect to the routing plan.

In order to resolve this dilemma, a common hierarchical approach is to split the optimisation process into multiple steps. For the previously mentioned VRPTW example, one can first optimise the number of vehicles and then the second objective can be optimised with the fixed result from the first optimisation problem. The hierarchical objective function is commonly applied with the heuristics, while the exact algorithms do not take account the number of vehicles in the objective function[14][15].

2.10.3. Multi-criteria Optimisation

The construction and real-life applications of the traditional VRP have been extensively studied by numerous researchers since the introduction of the TSP. Although many routing problems are established to model the real-life scenarios, they are usually formed with a single objective to minimise the cost of the routing plan. However, in the real-life practice of the transportation and logistics industry, a majority of the problems they confront are multi-objective. These problems are generalised as multi-objective integer linear programming (MOILP) problem.

From the analysis of multi-objective VRP by Jozefowiez et al.[60], there are three predominant situations where the multi-objective routing problems are implemented:

- Extension of classic academic problems: The model is modified further to improve its practical performance, while the preliminary objective is equally valued.
- Generalisation of classic problems: The classical model is generalised by adding more objectives rather than constraints and parameters, especially when the time window constraints are considered[41][53].
- Adaption of the real-life cases: Some real-life problems or applications are specified by the decision-maker with multiple objectives to be considered.

Numerous examples and studies of multi-objective and bi-objective optimisation approach can be find by Jozefowiez et al.[59][60], Lee and Ueng[64], Ombuki et al.[81] and Rahoual et al.[92].

3

Algorithms for VRP

Since the first study of VRP by Flood[37] in 1956, the methodology has been developed for decades to solve a variety of VRP. Along with the research of more complex VRP, the mathematical algorithm is improved as well to obtain faster and better results. Particularly, with the help of computer and programming languages, the solution can be computed faster and relatively larger scale LP problem can be resolved.

There are basically two aspects to consider when solving the VRP: quality of solution and computing time. For large scale problem, a trade-off between the two aspects is crucial when developing the model and heuristics based on the expectation of the solution process. A thorough study of the former strategies to the VRPs brings a better insight into the methodology and inspires the heuristics development for the project. This chapter introduces the development of heuristic methods and presents the characteristics and suitable situation of the methods, from the simplest heuristics with trial and error method developed by Dantzig and Rasmer[24] to the recent hybridisation strategies.

3.1. Overview of heuristic method

The first study of VRP is done by Flood[37] to solve the famous travelling salesman problem (TSP), which solved the problem with pure matrix operation. Dantzig introduced the simplex method to solve the TSP in a mathematically tractable way and provided the general procedure to solve linear programming problem. The simplex method is a pure algebraic procedure, which is beneficial to transform and solve the problem on the computer. Geometrically, the constraints form a polytope in the n-dimensional coordinate system to represents the constraints in the problem and the objective function is applied to the polytope to find the optimal solution[78]. The famous CPLEX Optimiser developed by IBM[2] is also developed based on the simplex method in the C language.

The simplex algorithm is considered as a pure exact method to solve the operation in a straightforward way, which can be used to solve small scale of problem manually or by computer with the global optimal solution. After decades of development in both the mathematical theory in linear algebra as well as software engineering, multiple commercial software applications exist nowadays. In contrast to the manual calculation in the mid-20th century, mathematical software and programming tools are applied to solve the linear programming problems based on the simplex algorithm. Commercial optimisation solvers have been developed, IBM ILOG CPLEX Optimization Studio[2] and Gurobi[7] are the pioneers among them. The optimisation solver can also be applied in the mainstream programming environments such as Python[5] and MATLAB[4]. Therefore, with the help of computer and programming languages, the solution can be computed faster and relatively larger scale LP problem than before can be resolved. However, when dealing with large-scale problems with numerous decision variables and constraints, the computing time is relatively long and sometimes not solvable due to the restriction of computing memory. In order to solve this dilemma, the development of heuristic method attracts the interests of researchers.

The development in the transportation industry leads to more complicated and large problems to solve, which reveal the disadvantages of the exact method. Exact algorithm consumes an enormous amount of computational time to seek the global optimum when applied to an LP problem with a large network. Based on the study of Toth and Vigo, the exact algorithm is not applicable to consistently solve a VRP problem with more than 50 customers[111].

Generally, a heuristic technique is a method to solve the problem in a reasonable time. The result may not be optimal, but is close enough to the global optimum which the exact method can possibly obtain. The simplest heuristics include trial and error and rule of thumb, but these methods are quite inefficient when solving large scale problem. The heuristic function approximates the exact method solution by searching the branching steps to follow the branch with the best result[85]. The main heuristic methods are constructive heuristics, improvement heuristics and metaheuristics.

3.2. Constructive Heuristics

The constructive heuristics are considered to provide a preliminary solution for implementation in the improvement heuristics. Many of the heuristics are easy to be implemented and fast for simple problems, such as the classical Clarke and Wright heuristic for the simple TSP and petal algorithm for the simple route scheduling problem.

The Clarke and Wright heuristic[19] was developed to solve the travelling salesman problem, which only considers one depot and the distances between every two vertices on the map. The method first generates the return routes between the depot to all other vertices $i \in V,\ i \neq 0$ on the map, and applies the saving criterion in every step by merging two routes $(0, \dots, i, 0)$ and $(0, j, \dots, 0)$ to a single route $(0, \dots, i, j, \dots, 0)$ with a saving in total cost of $s_{ij} = c_{i0} + c_{0j} - c_{ij}$. The solution is considered to be optimum when no further saving in the total cost is possible. A similar approach was developed by Laporte and Semet[62] as well, which only implement the largest saving in every step until no more saving is possible. However, since this method can only be applied to a really simple problem, and the algorithm can be easily replaced by computer and robustness of more advanced metaheuristics, this heuristic is no more advantageous in today's application.

On the other hand, the petal algorithm generate a set of S feasible VRP routes through set partitioning[97]. The mathematical expression is generated by Laport et al.[63]:

$$minimise \sum_{k \in S} d_k x_k \tag{3.1}$$

$$s.t. \sum_{k \in S} a_{ik} x_k = 1 \qquad \forall i = 1, \dots, n,\ i \neq 0 \tag{3.2}$$

$$x_k \in \{0, 1\} \qquad \forall k \in S \tag{3.3}$$

In this model, d_k is the cost of route $k \in S$ x_k is the binary decision variable equals to 1 if and only if route k is selected in the final routing plan. The binary coefficient a_{ik} is the most crucial parameter to solve the problem, which equals to 1 in and only if route k is assigned to customer i. The algorithm is suitable to solve problems with constraints other than capacity and route duration. However, the column generation has become a better choice for hard constraints.

3.3. Improvement heuristics

Classical improvement heuristics considers the intra-route and inter-route moves. The intra-route moves rearrange the sequence of the points traversed whin a certain route, and the inter-route moves interchange certain partial elements from one route with those from another route to generate a new schedule. The underlying strategy of these methods is to generate new candidates of the solution by transforming the result from the last iteration. The feasibility of every new solution is examined and the result is recorded for the final comparison.

The common intra-route move is the λ-optimality (λ-opt) method by Lin[65], which was introduced to solve the famous travelling salesman problem (TSP). Since there is only one continuous route as a solution for the TSP, no consideration of inter-route move is needed. The fundamental strategy is to first form a permutation of n nodes: $P = (i_1, i_2, \dots, i_n)$, which has a set of n links $u_{i_1 i_2}, u_{i_2 i_3}, \dots, u_{i_n i_1}$ and a initial total cost $C = d_{i_1 i_2} + d_{i_2 i_3} + \cdots + d_{i_n i_1}$. Then in each iteration, a set of links are replaced by another set of links to obtain a smaller overall cost with a new route. The study of Lin and Kernighan[66] dynamically change the value of the λ in the solving process. The solution is considered to be optimal when it is 1-optimal, which indicates that interchange of any two nodes in the route cannot get a better result. The limitation of this method is obvious, modification can only be done within a single route, which cannot solve the global optimisation of multi-route problems.

Another aspect is the inter-route improvement moves, which is used for a scheduling problem with multiple vehicles and routes in the plan. Most common methods are Relocate, Swap and 2-opt*[63]:

- Relocate: Removal of a number of k consecutive customers from one route and relocate them in another route.
- Swap: Interchange a number of k consecutive customers in two routes.
- 2-Opt*: For two random routes, eliminate one edge in each route and reconnect the remaining parts differently to form two new routes.

The moves mentioned above are randomly performed during the solving process, which generates an enormous amount of candidates to be examined and consumes a lot of times. Some obvious combinations with poor results are ideally avoided. Selection of possible combinations in the process is essential to shorten the computing time and increase efficiency. Improvement of the method can be found in numerous literature, such as the granular search[58], which considers the geographical information in the TSP to prevent moves between distant customers. More examples can be found in the research of Thompson and Psaraftis[109], Shaw[100] and Pisinger and Ropke[87].

3.4. Metaheuristics

Metaheuristic algorithms are high-level procedures that select various lower-level heuristics to perform a partial search in the solving process. For optimisation problem with incomplete information and limited computation capacity, metaheuristics can provide sufficiently good solution[113]. Current metaheuristics are categorised into two groups: local search algorithms and population-based algorithms. The main difference between the two groups is that the local search algorithm starts from a single solution point in the graph to search the best result, but the population-based algorithm generates the optimal result by evolvements of numerous solutions.

After years of study and development of metaheuristics, the frontiers between the two types is vague, where different algorithms can be implemented together or different concepts can be borrowed and emerged for the solving process. This leads to the hybrid approach of these algorithms and a brief introduction of the hybridizations will be performed.

3.5. Local search algorithm

The main idea of the local search algorithm is to first generate an initial solution x_i as the starting point, then in each iteration t, the corresponding neighbourhood $N(x_t)$ is searched to find another solution x_{t+1}. The selection of the new solution x_{t+1} depends on the method used. The simplest local search algorithm is the hill climbing, with a given optimisation problem to minimise the objective cost function $f(x)$, new solution x_{t+1} is chosen if $f(x_{t+1}) < f(x_t)$ in each iteration until no better solution can find. The hill climbing is optimal for convex optimisation problem[50], where the local optimum is also the global optimum. However, for solution space with multiple peaks, valleys or ridges, the solution can be trapped at different local optimum with different starting location or permanently circle around the ridge. Due to this disadvantage, the comparison between the surrounding solutions and the latest solution is not considered in the later metaheuristics development.

3.5.1. Simulated annealing (SA)

Compared with hill climbing, the new solution x_{t+1} in simulated annealing is selected randomly in the space to avoid cycling and stuck in local optimum. In this method, every potential new solution $x \in N(x_t)$ is still compared with the last solution. If $f(x) < f(x_t)$, then $x_{t+1} = x$. If the new solution x does not result in a better result, it is still chosen with a probability p_t, otherwise the last solution x_t remains. Commonly the probability p_t is defined as Equation 3.4, which negatively correlates to $f(x) - f(x_t)$ and the temperature θ_t is a decreasing function of t.

$$p_t = exp(-\frac{f(x) - f(x_t)}{\theta_t}) \tag{3.4}$$

The simulated annealing was first introduced by Pincus[86] by introducing the Metropolis algorithm with the Markov chain to solve the minimisation problem. It is proved that the simulated annealing can provide a convergent global optimum, which means that the starting point does not variate the final optimal solution. VRP example with simulated annealing can be found in the research of Osman[83].

3.5.2. Deterministic annealing (DA)

A slightly different variant of the simulated annealing is the deterministic annealing[33], which provides better than the simulated annealing. In contrast to SA, it uses deterministic method to decide the acceptance of the new solution x, but the comparison is between the new solution and the record $x*$, which is the best known solution. where σ is a number slightly larger than 1:

$$x_{t+1} = \begin{cases} x, \ if \ f(x) \leq \sigma f(x*) \\ x_t, \ otherwise. \end{cases} \tag{3.5}$$

3.5.3. Tabu search (TS)

Tabu search algorithm was initially invented by Fred W. Glover[43] in 1986 and formally introduced in 1989. It is an algorithm that tries to enable the search process to escape from a local optimum, and it continues to search the neighbourhood to find the global optimum[20][49]. It uses the tabu list to generate the areas that have been searched during the previous iterations, and the list is used as a reference to discourage the search from coming back to the previously-visited solutions and therefore avoid cycling.

Tabu search is a metaheuristic that aims to extend the neighbourhood with a particular focus on preventing local optimum. The previously visited solutions are prohibited or labelled as tabu for a certain amount of iterations θ, and the tabu list is generated to record the forbidden solutions to avoid cycling. The list is arranged to contain all forbidden moves sequentially and the first element from the tabu list will be eliminated when a new move has been made and inserted at the end of the list.

Gendreau et al.[42] and Zheng et al.[116] introduced an effective parallel improving tabu search algorithm for the Heterogeneous Fixed Fleet Vehicle Routing Problem (HFFVRP), which optimally serve several customers with known demands. In the research of Alonso et al.[9], the tabu search algorithm is implemented to solve site-dependent multi-trip periodic vehicle routing problem (SDMTPVRP), which considers a fleet of heterogeneous vehicles as well as multiple accessibility restrictions and periods during the solving process.

3.5.4. Iterated Local Search (ILS)

The iterated local search (ILS), literally is an algorithm that continuously applies a single type of local search algorithm until the end. It is a simple strategy to be implemented on top of any local search method, from the simplest steepest descent method in the neighbourhood or complicated tabu search algorithm. The concept is to start the designated local search mechanism normally from a selected starting point until it is allowed to stop. The solution at the end is perturbed to generate the starting point for the next iteration, which is used as the starting point to apply the next stage of the same mechanism again. Iterations of the same procedure continue until the solution is not improving anymore or a certain amount of iteration or time has been reached. Although the concept is easy to understand, the perturbation needs to be carefully designed to ensure the constitution of the original solution is not completely disrupted. An example can be found in the study of Chen et al.[17] about neighbourhood search descent heuristic for CVRP.

3.6. Population-based algorithms

The local search algorithms dedicate to avoiding cycling and escaping from the local optima, population-based methods are inspired by natural phenomena, such as biological characteristics. Furthermore, all these heuristics are relying on local search components to generate optimal solutions and most of them in the VRP applications are inherently hybrid.

3.6.1. Ant colony optimisation (ACO)

The ant colony optimisation (ACO) was first introduced by Dorigo et al.[27], which is inspired by the ant behaviour during the food search. In this method, the searching process mimics the behaviour of ants by introducing the pheromone trails. Each artificial ant is doing random searching in the space, it leaves some pheromone with varying quantities along the path it travelled. When another ant crosses the trail, it can detect the pheromone and decide whether or not to follow it with a high probability. Any ant that follows the previous trail leaves its own pheromone along the trail and the accumulation of pheromone will increase the possibility for other ants follow the same trail. The process ends when all ants are choosing the shortest path. Successful implementation of the ant colony optimisation on VRP can be found in the study of Reimann et al.[93], where the pheromone value τ_{ij} is introduced to measure the necessity of linking i and j compared with the last iteration for a better result, rather than using the Clarke and Wright algorithm[19].

3.6.2. Genetic algorithm (GA)

Another population-based algorithm is the genetic algorithm, which simulates the procedure of evolvement and natural selection. The first introduction of the genetic algorithm is done by Holland[52] based on the concept of Darwin's theory of evolution. Prins[89] first implemented the genetic algorithm on VRP, which outcompete most tabu search heuristics on multiple instances. Similar to the formation of chromosome in the next generation, an enormous amount of candidate solutions are generated for the iterative process, where the genetic operators are applied to allow the candidates to randomly recombined with each other or possibly mutated. The resultant solutions form a new generation, which is used in the next iteration of the algorithm. The new solution s_{new} with objective value $z(s_{new})$ is only allowed to be included in the population P if there is no existing solution $s \in P$ that the difference between the two objective value does not exceed a certain threshold Δ: $|z(s_{new}) - z(s)| \leq \Delta$. Normally the algorithm stops when a maximum number of iterations has been done or all the elements in the population satisfy the fitness level. Practical examples can be found by Nagata and Bräysy[75] for CVRP and Nagata and Kobayashi[] for solving the TSP by genetic algorithm (edge assembly crossover (EAX) in specific).

3.7. Hybridisations

After decades of heuristic method development, the algorithms are evolved by borrowing the concepts from different developed algorithms or emerging different heuristics together to solve more complex VRP or improve the existing method. Using hybrid methods to solve the VRP is common in the current research and the frontier between heuristics is fuzzy.

Hybrid methods have multiple examples. Adaptive memory programming (AMP)[105] considers multiple local search processes in parallel during the solving procedure to better handle real and dynamic application. This method led successful application on CVRP[106], which can provide the solution with quality in short computational time for multiple cases. Meta-meta hybridisations combined different metaheuristics sequentially or in parallel[21] in the algorithm structure. An implementation of metaheuristics with mathematical programming solver is also considered as a successful hybrid method, which shows improvements of the VRP result compared with the result from previous research[38].

4

Dynamic vehicle routing problems

4.1. Dynamic vehicle routing problems

When developing the model for the vehicle routing problem, it is usually assumed that all the relevant inputs are known, however, it is too idealistic in the real-life applications (Toth and Vigo[111]). Gounaris et al.[45] pointed out that parameters such as demands, travel and service times as well as the moment when the customer requests the service are often uncertain or unknown during the route design phase.

Besides considering the possible spillt passengers in the UNHAS planning, it is equally important to have an insight into the possible future demands and take them into account. Accurate demand anticipation can lead to more effective and cost-efficient planning. Based on the study of Bekta et al.[11], dynamic programming (DP) and linear (mixed) integer programming are the existing approaches to dynamic problems. The anticipatory algorithm has been first introduced by Powell et al.[88] when concerning the vehicle dispatching problem for long-haul truckload trucking applications. The technique has been used to assign drivers to random pickup and delivery requests over a given time window. Gendreau et al.[42] adapt the tabu search heuristic to the dynamic case on a parallel platform to increase the computational effort of the VRP. Similarly, the technique can be also applied to the aviation industry to face on-line operation conditions. Moudani and Mora-Camino[74] applied a mixture of dynamic programming approach with heuristic technique on a medium charter airline, which resulted in sufficient outcome on fleet assignments and maintenance scheduling. Godfrey and Powell[44] solved the stochastic dynamic resource allocation problem (SDRAP) with an adaptive dynamic programming algorithm which uses nonlinear functional approximations to evaluate the number of future resources, from which a better result can be obtained than the rolling-horizon methods (Sethi and Sorger[99]) on stochastic problems.

4.2. Source of dynamism

Based on the definition of Psaraftis[90], the main difference between dynamic VRPs and traditional VRPs is the certainty of information and the input data. In dynamic VRPs, the route schedule is planned with future or immediate requests under various operational constraints, where the immediate requests appear during the execution of the routing process. The solution of dynamic VRP attempts to respond all the available input data and requirements as well as reserving a certain amount of margin and flexibility to handle the unexpected changes and to implement new information during the execution.

Numerous examples can be found on problems handling dynamic requests. Common requests can be demands of goods, services regarding the number of requests, or variability in travel times[16] or service times. Moreover, issues such as service cancellations, unexpected accidents and changers in locations and demands may disrupt the preliminary scheduling significantly[115].

4.3. Dynamic programming

The dynamic programming (DP) is a mathematical optimisation method, it simplifies a complicated problem by disassembling it into multiple sub-problems in a recursive manner. In the study of Psaraftis[91], the many-to-many dial-a-ride problem is solved with a dynamic programming approach. A state vector

$(L, k_1, \ldots, k_N)$ is formed to indicate the current delivery stop L and the status k_j of every customer $j \in N$, and the feasibility of the next state needs to be recognised prior to the determination of the feasibility of current state.

4.4. Dynamic and stochastic problems

The dynamic and stochastic problems are regarded as dynamic problems that cannot be solved and followed concretely prior to the actual implementation of routing plan, partial input data is considered to be stochastic, where the parameter may be expected within a certain range or with a provided distribution[11]. The main difference between the deterministic and stochastic dynamic problems is that there is a strong motivation for the dynamic and stochastic problem to utilise all the accessible information to anticipate future events in the solving process. Some of the stochastic input can be foreseen based on the experience in previous operations or based on common sense. For example, the geographical location is a common reference, where the urban area with high population density may have more requests for the service[55].

A common situation is that the locations of the potential customers are known or within in a certain range. However, the demand amount from each customer is provided as a random variable with a know probability distribution.

4.5. Anticipation of future requests

In order to generate a preliminary solution for the stochastic problem, it is essential to have rough anticipation of the possible demand amount and location. Powell et al.[88] first introduced the anticipatory algorithm for the dynamic VRP for long-haul trucking applications. In this study, the future path of the truck is split into three stages: (1) the deterministic movements that are known at the time of the first dispatch; (2) the first uncertain dispatch after the deterministic moves and (3) the further uncertain movements which follows the first uncertain dispatch. The situation of the furthest in the future is first analysed to estimate the expected contribution of the truck, then stage 2 is evaluated to generate the marginal value of an extra truck in the near future at a region and finally, the deterministic movements are planned.

In practice, the Markov decision process is commonly used in dynamic programming and reinforcement learning when solving the optimisation problem. It is a mathematical model derived from the Markov property. The main idea of the Markov property is the memoryless property for the stochastic process. The conditional probability distribution of the future state in the process only depends on the current state, regardless of the past states[39]. Mathematically, for a stochastic process $X(t), t > 0$, the Markov property is represented as:

$$P[X(t+h) = y \mid X(s) = x(s), s \leq t] = P[X(t+h) = y \mid X(t) = x(t)] \quad \forall h > 0 \tag{4.1}$$

Thomas[108] developed the waiting strategies for the vehicle to handle stochastic future service requests from known customer locations. The Markov decision process is implemented in the model to derive the optimal policy. From the analysis result, the customer location information is more valuable than the likelihood of the customer to request service. Hvattum et al.[54] did research of dynamic and stochastic VRP with unknown customer locations and demands in advance. In this study, they developed a dynamic sample scenario hedge heuristic, where the historical data of the customer locations and demands are used to determine the probability distributions. The distribution is then used to anticipate the future demands.

III

Supporting work

1

Appendix 1

The design of the mathematical model is one of the most important element in this project. In this chapter, some constraints of the MILP model is discussed to provide a comprehensive explanation of how they are designed.

1.1. Pick-up/Delivery of the request

1.1.1. Determination of the pick-up/delivery node for each aircraft

In order to determine if the aircraft k picks up/delivers the passengers at the origin O^r or the destination D^r of the request r for the first or second visit, the pick-up/delivery sequence binary decision variable $z_{p_a}^{rk}$ and $z_{d_a}^{rk}$ are used as indicators. Several constraints are collaborating with each other to be effective.

The main constraints to determine the presence of $z_{p_a}^{rk}$ and $z_{d_a}^{rk}$ are constraints 1.1 to 1.4. This determination consists two parts:

- Overall identification of $z_{p_a}^{rk}$ & $z_{d_a}^{rk}$ existence.
- Determination of $z_{p_a}^{rk}$ and $z_{d_a}^{rk}$.

$$\sum_{a\in\{1,2\}} z_{p_a}^{rk} \leq 1 \qquad \forall r \in R, \forall k \in K \tag{1.1}$$

$$\sum_{a\in\{1,2\}} z_{p_a}^{rk} - \sum_{j:(O^r,j)\in A^k} q_{O^r j}^{rk} \leq 0 \qquad \forall r \in R, \forall k \in K, \forall (O^r, j) \in A^k \tag{1.2}$$

$$\sum_{a\in\{1,2\}} z_{d_a}^{rk} \leq 1 \qquad \forall r \in R, \forall k \in K \tag{1.3}$$

$$\sum_{a\in\{1,2\}} z_{d_b}^{rk} - \sum_{i:(i,D^r)\in A^k} q_{iD^r}^{rk} \leq 0 \qquad \forall r \in R, \forall k \in K, \forall (i, D^r) \in A^k \tag{1.4}$$

The first step is done by evaluating the number of passengers to be transferred from the origin or to the destination of each request. If one or more aircraft pick up a number of passengers for a certain request at its origin, then it means that this request is will be accomplished, and therefore the pick-up or delivery decision variable $z_{p_a}^{rk} = 1$ or $z_{d_a}^{rk} = 1$ for certain combination of k, r and a. Therefore

$\forall r \in R, \forall k \in K$:

$$if \sum_{j:(O^r,j)\in A^k} q^{rk}_{O^r j} > 0: \quad \Rightarrow \quad \sum_{a\in\{1,2\}} z^{rk}_{p_a} = 1 \tag{1.5}$$

$$if \sum_{j:(O^r,j)\in A^k} q^{rk}_{O^r j} = 0: \quad \Rightarrow \quad \sum_{a\in\{1,2\}} z^{rk}_{p_a} = 0 \tag{1.6}$$

$$if \sum_{i:(i,D^r)\in A^k} q^{rk}_{iD^r} > 0: \quad \Rightarrow \quad \sum_{a\in\{1,2\}} z^{rk}_{d_b} = 1 \tag{1.7}$$

$$if \sum_{i:(i,D^r)\in A^k} q^{rk}_{iD^r} = 0: \quad \Rightarrow \quad \sum_{a\in\{1,2\}} z^{rk}_{d_b} = 0 \tag{1.8}$$

Constraints 1.1 and 1.3 indicates that a certain request can only be picked up/delivered by the same aircraft once. Combined with constraints 1.2 and 1.4, they attain the consideration of pick-up and delivery decision variables.

Since it is allowed for each aircraft to visit any airport at most twice. After the presence of $z^{rk}_{p_a}$ or $z^{rk}_{d_a}$ is determined, the second step is to identify if it is the first or the second time this aircraft arrives at the origin/destination of this request. Due to the fact that $\sum_{a\in\{1,2\}} z^{rk}_{p_a} = 1$ or $\sum_{a\in\{1,2\}} z^{rk}_{d_a} = 1$ has been proven in the previous step and the pick-up/delivery decision variable is binary, therefore it is only necessary to determine which decision variable in the couple is equal to 1. This step is crucial for the problem. The specific arrival/departure time decision variable can therefore be selected and constrained by the pick-up/delivery time requirement. Take the pick-up decision variable as example, the underlying logic of the second step is shown as follow:

$$\begin{aligned}
&\sum_{b\in\{1,2\}} u^k_{O^r_a j_b} = 1, q^{rk}_{O^r j} > 0 \quad \Rightarrow \quad z^{rk}_{p_a} = 1,\\
&\sum_{b\in\{1,2\}} u^k_{O^r_a j_b} = 1, q^{rk}_{O^r j} = 0 \quad \Rightarrow \quad z^{rk}_{p_a} \geq 0,\\
&\sum_{b\in\{1,2\}} u^k_{O^r_a j_b} = 0, q^{rk}_{O^r j} > 0 \quad \Rightarrow \quad z^{rk}_{p_a} \geq 0,\\
&\sum_{b\in\{1,2\}} u^k_{O^r_a j_b} = 0, q^{rk}_{O^r j} = 0 \quad \Rightarrow \quad z^{rk}_{p_a} \geq 0,
\end{aligned} \quad \forall r \in R, \forall k \in K, \forall (O^r, j) \in A^k, \forall a \in \{1,2\} \tag{1.9}$$

Consequently, the resultant mathematical expression of pick-up/delivery decision variables determination can be achieved by constraints 1.10 and 1.11:

$$z^{rk}_{p_a} \geq \sum_{b\in\{1,2\}} u^k_{O^r_a j_b} + \frac{1}{Q^k} q^{rk}_{O^r j} - 1, \quad \forall a \in \{1,2\}, \forall r \in R, \forall k \in K, \forall (O^r, j) \in A^k \tag{1.10}$$

$$z^{rk}_{d_b} \geq \sum_{a\in\{1,2\}} u^k_{i_a D^r_b} + \frac{1}{Q^k} q^{rk}_{iD^r} - 1, \quad \forall b \in \{1,2\}, \forall r \in R, \forall k \in K, \forall (i, D^r) \in A^k \tag{1.11}$$

1.1.2. Pick-up/Delivery time interval

After the pick-up/delivery decision variables have been selected, they can be constrained by the pick-up/delivery time interval. This step can be achieved by the following decision strategy:

$$z^{rk}_{p_a} = 1: \quad \Rightarrow \quad t^r_{p_a} \leq w^k_{d_{O^r_a}} \leq t^r_{p_b}, \quad \forall a \in \{1,2\}, \forall r \in R, \forall k \in K \tag{1.12}$$

$$z^{rk}_{p_a} = 0: \quad \Rightarrow \quad 0 \leq w^k_{d_{O^r_a}} \leq \infty, \quad \forall a \in \{1,2\}, \forall r \in R, \forall k \in K \tag{1.13}$$

$$z^{rk}_{d_b} = 1: \quad \Rightarrow \quad t^r_{d_a} \leq w^k_{a_{D^r_a}} \leq t^r_{d_b}, \quad \forall b \in \{1,2\}, \forall r \in R, \forall k \in K \tag{1.14}$$

$$z^{rk}_{d_b} = 0: \quad \Rightarrow \quad 0 \leq w^k_{a_{D^r_a}} \leq \infty, \quad \forall b \in \{1,2\}, \forall r \in R, \forall k \in K \tag{1.15}$$

'

The above mathematical expression can be represented by constraints 1.16 to 1.17:

$$t^r_{p_a} z^{rk}_{p_a} \leq w^k_{d_{O^r_a}} \leq t^r_{p_b} z^{rk}_{p_a} + M(1 - z^{rk}_{p_a}), \quad \forall a \in \{1,2\}, \forall r \in R, \forall k \in K \tag{1.16}$$

$$t^r_{d_a} z^{rk}_{d_b} \leq w^k_{a_{D^r_b}} \leq t^r_{d_b} z^{rk}_{d_b} + M(1 - z^{rk}_{d_b}), \quad \forall b \in \{1,2\}, \forall r \in R, \forall k \in K \tag{1.17}$$

1.2. Minimal time difference between two departures/arrivals

In order to ensure safety on the runway, enough time gap needs to be guaranteed between every two departures or arrivals. Therefore constraints 1.18 and 1.19 are designed to compare the time difference between every two departures/arrivals with the minimum time interval. $\forall k_1, k_2 \in K, k_1 \neq k_2, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\}$:

$$|w^{k_2}_{d_{i_b}} - w^{k_1}_{d_{i_a}}| \geq \Delta t \tag{1.18}$$

$$|w^{k_2}_{a_{i_b}} - w^{k_1}_{a_{i_a}}| \geq \Delta t \tag{1.19}$$

When utilising the IBM CPLEX Optimizer to solve linear programming problem, it is impossible to operate absolute value in any constraint. Therefore, the absolute values signs in constraints 1.18 and 1.19 need to be decomposed. In order to determine whether the value in between the absolute value signs is positive or negative, the indicator $s^{k_1k_2}_{a_{i_{ab}}}$ and $s^{k_1k_2}_{d_{i_{ab}}}$ are introduced to represent the sequence of aircraft arrivals or departures.

On the other hand, it is also considered that not all aircraft will stop at the airport. In this case, for the specific combination of k, i and a, the time decision variables $w^{k_1}_{a_{i_a}}, w^{k_2}_{a_{i_b}}, w^{k_1}_{d_{i_a}}, w^{k_2}_{d_{i_b}}$ all equal to 0. Consequently, they are not considered in this constraint and therefore the relevant $s^{k_1k_2}_{a_{i_{ab}}}, s^{k_2k_1}_{a_{i_{ba}}}, s^{k_1k_2}_{d_{i_{ab}}}, s^{k_2k_1}_{d_{i_{ba}}} = 0$ as well and will be excluded in this constraint.

$$w^{k_1}_{a_{i_a}} \leq w^{k_2}_{a_{i_b}} \Rightarrow s^{k_1k_2}_{a_{i_{ab}}} = 1, s^{k_2k_1}_{a_{i_{ba}}} = 0, \quad \forall k_1, k_2 \in K, k_1 \neq k_2, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{1.20}$$

$$w^{k_1}_{a_{i_a}}, w^{k_2}_{a_{i_b}} = 0 \Rightarrow s^{k_1k_2}_{a_{i_{ab}}}, s^{k_2k_1}_{a_{i_{ba}}} = 0, \quad \forall k_1, k_2 \in K, k_1 \neq k_2, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{1.21}$$

$$w^{k_1}_{d_{i_a}} \leq w^{k_2}_{d_{i_b}} \Rightarrow s^{k_1k_2}_{d_{i_{ab}}} = 1, s^{k_2k_1}_{d_{i_{ba}}} = 0, \quad \forall k_1, k_2 \in K, k_1 \neq k_2, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{1.22}$$

$$w^{k_1}_{d_{i_a}}, w^{k_2}_{d_{i_b}} = 0 \Rightarrow s^{k_1k_2}_{d_{i_{ab}}}, s^{k_2k_1}_{d_{i_{ba}}} = 0, \quad \forall k_1, k_2 \in K, k_1 \neq k_2, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{1.23}$$

However, these conditional constraints are still not applicable in linear programming. Therefore by considering the characteristics of binary decision variables $\{s^{k_1k_2}_{a_{i_{ab}}}\}$ and $\{s^{k_1k_2}_{d_{i_{ab}}}\}$, 1.20 to 1.23 are adapted as follow:

$$(w^{k_1}_{a_{i_a}} - w^{k_2}_{a_{i_b}}) + M s^{k_1k_2}_{a_{i_{ab}}} \geq 0, \quad \forall k_1, k_2 \in K, k_1 \neq k_2, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{1.24}$$

$$(w^{k_1}_{d_{i_a}} - w^{k_2}_{d_{i_b}}) + M s^{k_1k_2}_{d_{i_{ab}}} \geq 0, \quad \forall k_1, k_2 \in K, k_1 \neq k_2, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{1.25}$$

$$s^{k_1k_2}_{a_{i_{ab}}} + s^{k_2k_1}_{a_{i_{ba}}} \leq 1, \quad \forall k_1, k_2 \in K, k_1 \neq k_2, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{1.26}$$

$$s^{k_1k_2}_{d_{i_{ab}}} + s^{k_2k_1}_{d_{i_{ba}}} \leq 1, \quad \forall k_1, k_2 \in K, k_1 \neq k_2, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{1.27}$$

$$s^{k_1k_2}_{a_{i_{ab}}} + s^{k_2k_1}_{a_{i_{ba}}} - (w^{k_1}_{a_{i_a}} + w^{k_2}_{a_{i_b}}) \leq 0, \quad \forall k_1, k_2 \in K, k_1 \neq k_2, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{1.28}$$

$$s^{k_1k_2}_{d_{i_{ab}}} + s^{k_2k_1}_{d_{i_{ba}}} - (w^{k_1}_{d_{i_a}} + w^{k_2}_{d_{i_b}}) \leq 0, \quad \forall k_1, k_2 \in K, k_1 \neq k_2, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{1.29}$$

Constraints 1.24 and 1.25 determines the sequence of aircraft arrival and departure. Constraints 1.26 and 1.27 are the general constraints for the binary decision variables, as they can have at most one decision variable in the pair that equals to 1. Constraints 1.28 and 1.29 reveals the exceptional situation where the aircraft does not visit the airport in question.

After determination of both arrival and departure sequence decision variables, they are used to form the minimal time interval constraints between every two arrivals/departures:

$$s^{k_1k_2}_{a_{i_{ab}}} = 1 \quad \Rightarrow \quad w^{k_2}_{a_{i_b}} - w^{k_1}_{a_{i_a}} \geq \Delta t, \qquad \forall k_1, k_2 \in K, k_1 \neq k_2, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{1.30}$$

$$s^{k_1k_2}_{a_{i_{ab}}} = 0 \quad \Rightarrow \quad w^{k_2}_{a_{i_b}} - w^{k_1}_{a_{i_a}} \leq 0, \qquad \forall k_1, k_2 \in K, k_1 \neq k_2, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{1.31}$$

$$s^{k_1k_2}_{d_{i_{ab}}} = 1 \quad \Rightarrow \quad w^{k_2}_{d_{i_b}} - w^{k_1}_{d_{i_a}} \geq \Delta t, \qquad \forall k_1, k_2 \in K, k_1 \neq k_2, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{1.32}$$

$$s^{k_1k_2}_{a_{i_{ab}}} = 0 \quad \Rightarrow \quad w^{k_2}_{d_{i_b}} - w^{k_1}_{d_{i_a}} \leq 0, \qquad \forall k_1, k_2 \in K, k_1 \neq k_2, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{1.33}$$

Based on the definition of $s^{k_1k_2}_{a_{i_{ab}}}$, it can only equal to 0 when $w^{k_2}_{a_{i_b}} < w^{k_1}_{a_{i_a}}$ or $w^{k_2}_{a_{i_b}}, w^{k_1}_{a_{i_a}} = 0$. Similar explanation also applies when $s^{k_1k_2}_{d_{i_{ab}}} = 0$. These can be achieved by conditional constraints 1.31 and 1.33.

After implementing these condition into the constraints to generate a linear relation, the resultant mathematical expression of these constraints are reformulated as follows:

$$w^{k_2}_{a_{i_b}} - w^{k_1}_{a_{i_a}} \geq (\Delta t + M) s^{k_1k_2}_{a_{i_{ab}}} - M, \qquad \forall k_1, k_2 \in K, k_1 \neq k_2, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{1.34}$$

$$w^{k_2}_{d_{i_b}} - w^{k_1}_{d_{i_a}} \geq (\Delta t + M) s^{k_1k_2}_{d_{i_{ab}}} - M, \qquad \forall k_1, k_2 \in K, k_1 \neq k_2, \forall i \in V^{k_1} \cap V^{k_2}, \forall a, b \in \{1,2\} \tag{1.35}$$

1.3. Passenger transfer

In some situations, passengers are arranged to transfer between flights in order to decrease unnecessary aircraft utilisation and save the overall aircraft operational cost. The exact method considers the passenger transfer when forming the mathematical model, but an additional timeline check is needed after the flight plan has been generated. However, for the tabu search approach, the passenger transfer planning can be determined by introducing the decision variable set $\{t^k_{i^+_a}\}$ and $\{t^k_{i^-_a}\}$ in corresponding constraints.

In the tabu search method model, five sets of constraints are designed to consider the passenger transfer requirements. As stated in the definition of $\{t^k_{i^+_a}\}$ and $\{t^k_{i^-_a}\}$, they are determined by the corresponding passenger flow going in or coming out of the airport, which is represented by $\{q^{rk}_{ij}\}$:

$$\begin{aligned} q^{rk}_{ij} > 0, q^{rk}_{ij} = f^r_{uq}(u^k_{i_a j_b}) \quad &\Rightarrow \quad t^{rk}_{i^-_a} = 1, t^{rk}_{j^+_a} = 1, \\ q^{rk}_{ij} = 0, q^{rk}_{ij} = f^r_{uq}(u^k_{i_a j_b}) \quad &\Rightarrow \quad t^{rk}_{i^-_a} = 0, t^{rk}_{j^+_a} = 0, \end{aligned} \qquad \forall u^k_{i_a j_b} \in U \tag{1.36}$$

Therefore, decision variables $\{t^k_{i^+_a}\}$ and $\{t^k_{i^-_a}\}$ determines their value by normalising the relevant passenger flows, which result in the fact that the decision variable sets $\{t^k_{i^+_a}\}$ and $\{t^k_{i^-_a}\}$ represent the normalised passenger flow going in or out the airport:

$$\begin{aligned} q^{rk}_{ij} > 0 \quad &\Rightarrow \quad 0 < \frac{1}{Q^k} q^{rk}_{ij} \leq 1, \\ q^{rk}_{ij} = 0 \quad &\Rightarrow \quad \frac{1}{Q^k} q^{rk}_{ij} = 0, \end{aligned} \qquad \forall u^k_{i_a j_b} \in U \tag{1.37}$$

The last constraint examines if passengers of a particular request can transfer between two aircraft at the airport, which ensures that the passengers have enough transfer time between flights if necessary. It is only considered when there is passenger flow passing the airport and both flights are operated for the same request:

$$\begin{aligned} &At\ airport\ i: \quad q^{rk_1}_{j'i} > 0, q^{rk_2}_{ij} > 0 \quad \Rightarrow \quad w^{k_1}_{a_{i_{a'}}} + T^i_{transfer} \leq w^{k_2}_{d_{i_a}} \\ &where: \ q^{rk_1}_{j'i} = f^r_{uq}(u^{k_1}_{j'_{b'} i_{a'}}),\ q^{rk_2}_{ij} = f^r_{uq}(u^{k_2}_{i_a j_b}), \forall u^{k_1}_{j'_{b'} i_{a'}}, u^{k_2}_{i_a j_b} \in U \end{aligned} \tag{1.38}$$

Therefore, the decision variable sets $\{t^k_{i^+_a}\}$ and $\{t^k_{i^-_a}\}$ can be introduced to substitute the passenger flow for convenience:

$$t^{rk_1}_{i^+_a} = 1, t^{rk_2}_{i^-_b} = 1(t^{rk_1}_{i^+_a} + t^{rk_2}_{i^-_b} = 2) \Rightarrow w^{k_1}_{a_{i_a}} + T^i_{transfer} \leq w^{k_2}_{d_{i_b}}, \quad \forall r \in R, \forall k_1, k_2 \in K \quad (1.39)$$

$$t^{rk_1}_{i^+_a} = 1, t^{rk_2}_{i^-_b} = 0(t^{rk_1}_{i^+_a} + t^{rk_2}_{i^-_b} = 1) \Rightarrow no\ constraint, \quad \forall r \in R, \forall k_1, k_2 \in K \quad (1.40)$$

$$t^{rk_1}_{i^+_a} = 0, t^{rk_2}_{i^-_b} = 1(t^{rk_1}_{i^+_a} + t^{rk_2}_{i^-_b} = 1) \Rightarrow no\ constraint, \quad \forall r \in R, \forall k_1, k_2 \in K \quad (1.41)$$

$$t^{rk_1}_{i^+_a} = 0, t^{rk_2}_{i^-_b} = 0(t^{rk_1}_{i^+_a} + t^{rk_2}_{i^-_b} = 0) \Rightarrow no\ constraint, \quad \forall r \in R, \forall k_1, k_2 \in K \quad (1.42)$$

Based on the different sum value of $t^k_{i^+_a}$ and $t^k_{i^-_a}$, the above circumstances can be classified. The conditional constraint can be therefore derived as:

$$w^{k_2}_{d_{i_{a'}}} - w^{k_1}_{a_{i_a}} - (M + T^i_{transfer})(t^{rk_2}_{i^-_{a'}} + t^{rk_1}_{i^+_a}) \geq -2M - T^i_{transfer}, \quad \forall r \in R, \forall u^{k_1}_{j_b i_a}, u^{k_2}_{i_{a'} j'_{b'}} \in U \quad (1.43)$$

1.4. Timeline Error Processing

In some cases, passengers are allowed to transfer at airport and take another aircraft for the remaining journey. However, due to the lack of relevant constraints within the model, the output may show that some passengers are picked up by another aircraft before they physically arrived at that place. An error example can be found in Appeldix G. 6.1 of the scientific report.

In order to resolve this disorder systematically, a timeline checking process in designed as shown in Figure 1.1. The daily operation schedule generated from the model output is checked after each run of the model. If one or more timeline errors have been detected, a row generation is issued per each error to the model. The process will continue until a feasible solution with no timeline error is found.

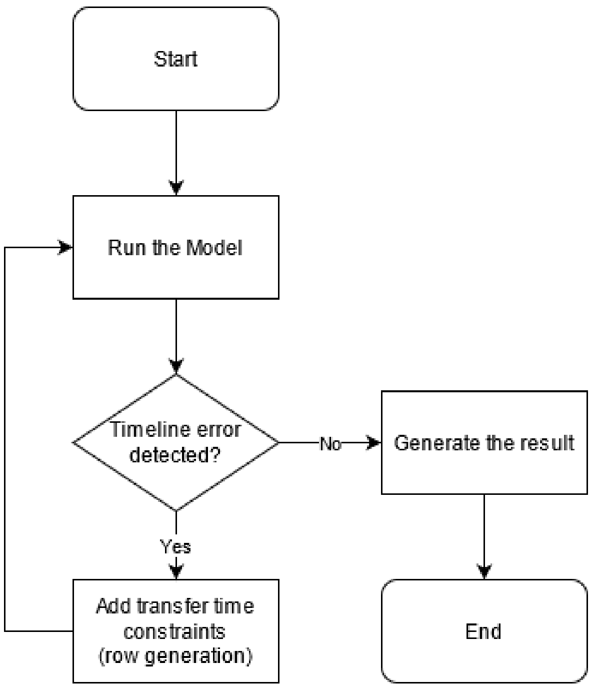


Figure 1.1: Flow chart of timeline error process

1.4.1. Row generation

The row generation focuses on the timeline error at the same place. If an error is detected, an additional transfer time constraint will be added in the model, which forces the outgoing aircraft to departure later than the incoming aircraft when transferring the passengers at the airport. Suppose a situation where the passengers from aircraft k_1 landed at node i_a needs to transfer to the aircraft k_2 that departures from node i_b, then the additional transfer time constraint is expressed as 1.44:

$$w_{d_{i_b}}^{k_2} - w_{a_{i_a}}^{k_1} - (M + \frac{1}{2}T_{transfer}^{i})(y^{k_1} + y^{k_2}) \geq -2M \tag{1.44}$$

where $T_{transfer}^{i}$ is the time needed for the passengers to transfer from an aircraft to another at airport i. Currently, the $T_{transfer}^{i}$ is assumed to be the same as the TAT at the airport.

Bibliography

[1] UN Humanitarian Air Service. URL https://www.wfp.org/unhas.

[2] ILOG CPLEX Optimization Studio. URL https://www.ibm.com/products/ilog-cplex-optimization-studio.

[3] International Committee of the Red Cross. URL https://www.icrc.org/.

[4] Matlab. URL https://www.mathworks.com/products/matlab.html.

[5] Python. URL https://www.python.org.

[6] United Nations World Food Programme (WFP). URL https://www.wfp.org/.

[7] Gurobi Optimiser, Jun 2020. URL https://www.gurobi.com/.

[8] Abara, J. Applying integer linear programming to the fleet assignment problem. Interfaces, 19 (4):2028, 1989. doi: 10.1287/inte.19.4.20.

[9] Alonso, F., Alvarez, M. J. and Beasley, J. E. A tabu search algorithm for the periodic vehicle routing problem with multiple vehicle trips and accessibility restrictions. Journal of the Operational Research Society, 59(7):963976, 2008. doi: 10.1057/palgrave.jors.2602405.

[10] Baldacci, R., Battarra, M. and Vigo, D. Routing a heterogeneous fleet of vehicles. In The Vehicle Routing Problem: Latest Advances and New Challenges, page 327. doi: 10.1007/978-0-387-77778-8_1.

[11] Bekta, T., Repoussis, P. P. and Tarantilis, C. D. Dynamic vehicle routing problems. In Vehicle routing: problems, methods, and applications (2nd Edition), chapter 11, page 299347. 2014. doi: 10.1137/1.9781611973594.ch11.

[12] Beltrami, E. J. and Bodin, L. D. Networks and vehicle routing for municipal waste collection. Networks, 4(1):6594, 1974. doi: 10.1002/net.3230040106.

[13] Bodin, L. D., Maniezzo, V. and Mingozzi, A. Street routing and scheduling problems. In International Series in Operations Research Management Science Handbook of Transportation Science, volume 23, page 413449. doi: 10.1007/0-306-48058-1_12.

[14] Bräysy, O. and Gendreau, M. Vehicle routing problem with time windows, part i: Route construction and local search algorithms. Transportation Science, 39(1):104118, 2005. doi: 10.1287/trsc.1030.0056.

[15] Bräysy, O. and Gendreau, M. Vehicle routing problem with time windows, part ii: Metaheuristics. Transportation Science, 39(1):119139, 2005. doi: 10.1287/trsc.1030.0057.

[16] Chen, H-H., Hsueh, C-F. and Chang M-S. The real-time time-dependent vehicle routing problem. Computers Operations Research, 42:383–408, 2006. doi: 10.1016/j.tre.2005.01.003.

[17] Chen, P., Huang, H-K. and Dong, X-Y. Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem. Expert Systems with Applications, 37(2):16201627, 2010. doi: 10.1016/j.eswa.2009.06.047.

[18] Christofides, N., Mingozzi, A. and Toth, P. The vehicle routing problem. In Christofides, N., Mingozzi, A., Toth, P. and Sandi, C., editor, Combinatorial Optimization, chapter 11, pages 315–338. Wiley, 1979.

[19] Clarke, G. and Wright, J. W. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research, 12(4):568581, 1964. doi: 10.1287/opre.12.4.568.

[20] Coello, C. C., Lamont, G. B. and Van Veldhuizen, D. A. Evolutionary algorithms for solving multi-objective problems. Springer, 2002.

[21] Cordeau, J-F. and Maischberger, M. A parallel iterated tabu search heuristic for vehicle routing problems. Computers Operations Research, 39(9):20332050, 2012. doi: 10.1016/j.cor.2011.09.021.

[22] Cordeau, J-F., Desaulniers, G., Desrosiers, J., Solomon, M. M. and Soumis, F. VRP with time windows. In The vehicle routing problem, chapter 7, pages 155–194. Society for Industrial and Applied Mathematics, 2002.

[23] Cordeau, J-F., Gendreau, M. and Laporte, G. A tabu search heuristic for periodic and multi-depot vehicle routing problems. Networks, 30(2):105119, 1997. doi: 10.1002/(sici)1097-0037(199709)30:2<105::aid-net5>3.0.co;2-g.

[24] Dantzig, G. B. and Ramser, J. H. The truck dispatching problem. Management Science, 6(1):8091, 1959. doi: 10.1287/mnsc.6.1.80.

[25] Dantzig, G. B., Fulkerson, R. and Johnson, S. Solution of a large-scale traveling-salesman problem. Journal of the Operations Research Society of America, 2(4):393410, 1954. doi: 10.1287/opre.2.4.393.

[26] Dethloff, J. Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up. OR Spektrum, 23(1):7996, 2001. doi: 10.1007/pl00013346.

[27] Dorigo, M., Maniezzo, V. and Colorni, A. Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics), 26(1):2941, 1996. doi: 10.1109/3477.484436.

[28] Dorn, A. W. Air power in UN operations: wings for peace. Ashgate, 2014.

[29] Drexl, M. Synchronization in vehicle routinga survey of vrps with multiple synchronization constraints. Transportation Science, 46(3):297316, 2012. doi: 10.1287/trsc.1110.0400.

[30] Dror, M. Arc Routing Theory, Solutions, and Applications. Kluwer Academic Publishers, 2000.

[31] Dror, M. and Trudeau, P. Savings by split delivery routing. Transportation Science, 23(2):141145, 1989. doi: 10.1287/trsc.23.2.141.

[32] Dror, M. and Trudeau, P. Split delivery routing. Naval Research Logistics (NRL), 37(3):383402, 1990. doi: 10.1002/nav.3800370304.

[33] Dueck, G. New optimization heuristics: The great deluge algorithm and the record-to-record travel. Journal of Computational Physics, 104:86–92, 1993. doi: 10.1006/jcph.1993.1010.

[34] Eftekhar, M., Masini, A., Robotis, A. and Van Wassenhove, L. N. Vehicle procurement policy for humanitarian development programs. Production and Operations Management, 23(6):951964, 2013. doi: 10.1111/poms.12108.

[35] Eksioglu, B., Vural, A. V. and Reisman, A. The vehicle routing problem: A taxonomic review. Computers Industrial Engineering, 57(4):14721483, 2009. doi: 10.1016/j.cie.2009.05.009.

[36] Fischetti, M., Toth, P. and Vigo, D. A branch-and-bound algorithm for the capacitated vehicle routing problem on directed graphs. Operations Research, 42(5):846859, 1994. doi: 10.1287/opre.42.5.846.

[37] Flood, M. M. The traveling-salesman problem. Operations Research, 4(1):6175, 1956. doi: 10.1287/opre.4.1.61.

[38] Foster, B. A. and Ryan, D. M. An integer programming approach to the vehicle scheduling problem. Journal of the Operational Research Society, 27(2):367384, 1976. doi: 10.1057/jors.1976.63.

[39] Gagniuc, P. A. Markov chains: from theory to implementation and experimentation. Wiley, 2017.

[40] Garey, M. R. and Johnson, D. S. Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman Co., USA, 1990. ISBN 0716710455.

[41] Geiger, M. Genetic algorithms for multiple objective vehicle routing. Metaheuristic International Conference 2001 (MIC2001), pages 348–353, 2001.

[42] Gendreau, M., Guertin, F., Potvin, J., and Taillard, É. Parallel tabu search for real-time vehicle routing and dispatching. Transportation Science, 33(4):381390, 1999. doi: 10.1287/trsc.33.4.381.

[43] Glover, F. Future paths for integer programming and links to artificial intelligence. Computers Operations Research, 13(5):533549, 1986. doi: 10.1016/0305-0548(86)90048-1.

[44] Godfrey, G. A. and Powell, W. B. An adaptive dynamic programming algorithm for dynamic fleet management, i: Single period travel times. Transportation Science, 36(1):2139, 2002. doi: 10.1287/trsc.36.1.21.570.

[45] Gounaris, C. E., Repoussis, P. P., Tarantilis, C. D., Wiesemann, W. and Floudas, C. A. An adaptive memory programming framework for the robust capacitated vehicle routing problem. Transportation Science, 50(4):12391260, 2016. doi: 10.1287/trsc.2014.0559.

[46] Guan, M. On the windy postman problem. Discrete Applied Mathematics, 9(1):4146, 1984. doi: 10.1016/0166-218x(84)90089-1.

[47] Heilporn, G., Cordeau, J-F. and Laporte, G. The delivery man problem with time windows. Discrete Optimization, 7(4):269282, 2010. doi: 10.1016/j.disopt.2010.06.002.

[48] Hempsch, C. and Irnich, S. Routing a heterogeneous fleet of vehicles. In The Vehicle Routing Problem: Latest Advances and New Challenges, page 421444. doi: 10.1007/978-0-387-77778-8_1.

[49] Hillier, F.S. and Lieberman, G.J. Introduction to Operations Research. Mcgraw-Hill Education, 2014.

[50] Hiriart-Urruty, J-B. and Lemarechal, C. Convex analysis and minimization algorithms. I. Springer-Verlag, 1993.

[51] Hitchcock, F. L. The distribution of a product from several sources to numerous localities. Journal of Mathematics and Physics, 20(1-4):224230, 1941. doi: 10.1002/sapm1941201224.

[52] Holland, J. H. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence. 1975.

[53] Hong, S-C. and Park, Y-B. A heuristic for bi-objective vehicle routing with time window constraints. International Journal of Production Economics, 62(3):249258, 1999. doi: 10.1016/s0925-5273(98)00250-3.

[54] Hvattum, L. M., Løkketangen, A. and Laporte, G. Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. Transportation Science, 40(4):421438, 2006. doi: 10.1287/trsc.1060.0166.

[55] Ichoua, S., Gendreau, M. and Potvin, J-Y. Planned route optimization for real-time vehicle routing. In Operations Research/Computer Science Interfaces Series Dynamic Fleet Management, chapter 1, page 118. doi: 10.1007/978-0-387-71722-7_1.

[56] Irnich, S., Toth, P. and Vigo, D. The family of vehicle routing problems. In Vehicle routing: problems, methods, and applications (2nd Edition), chapter 1, pages 1–33. 2014. doi: 10.1137/1.9781611973594.ch1.

[57] Jiang, Y.,Xu, X., Zhang, H. and Luo, Y. Taxiing route scheduling between taxiway and runway in hub airport. Mathematical Problems in Engineering, 2015:114, 2015. doi: 10.1155/2015/925139.

[58] Johnson, D. S. and Mcgeoch, L. A. The traveling salesman problem: A case study in local optimization. In Local Search in Combinatorial Optimization, page 215310. doi: 10.2307/j.ctv346t9c.13.

[59] Jozefowiez, N., and Semet, F. and Talbi, E.-G. Parallel and hybrid models for multi-objective optimization: Application to the vehicle routing problem. Parallel Problem Solving from Nature PPSN VII, Lecture Notes in Computer Science, 2439:271280, 2002. doi: 10.1007/3-540-45712-7_26.

[60] Jozefowiez, N., and Semet, F. and Talbi, E-G. Multi-objective vehicle routing problems. European Journal of Operational Research, 189(2):293309, 2008. doi: 10.1016/j.ejor.2007.05.055.

[61] Kim, B-I., Kim, S. and Sahoo, S. Waste collection vehicle routing problem with time windows. Computers Operations Research, 33(12):36243642, 2006. doi: 10.1016/j.cor.2005.02.045.

[62] Laporte, G. and Semet, F. Classical heuristics for the capacitated vrp. In The Vehicle Routing Problem, chapter 5, page 109128. 2002. doi: 10.1137/1.9780898718515.ch5.

[63] Laporte, G., Ropke, S. and Vidal, T. The family of vehicle routing problems. In Vehicle routing: problems, methods, and applications (2nd Edition), chapter 4, pages 1–33. 2014. doi: 10.1137/1.9781611973594.ch1.

[64] Lee, T-R. and Ueng, J-H. A study of vehicle routing problems with loadbalancing. International Journal of Physical Distribution Logistics Management, 29(10):646658, 1999. doi: 10.1108/09600039910300019.

[65] Lin, S. Computer solutions of the traveling salesman problem. Bell System Technical Journal, 44 (10):22452269, 1965. doi: 10.1002/j.1538-7305.1965.tb04146.x.

[66] Lin, S. and Kernighan, B. W. An effective heuristic algorithm for the traveling-salesman problem. Operations Research, 21(2):498516, 1973. doi: 10.1287/opre.21.2.498.

[67] Liu, M., Cao, J., Liang, J. and Chen, M. Epidemic-logistics modeling: a new perspective on operations research. Science Press, 2020.

[68] Martello, S. and Toth, P. Knapsack Problems: Algorithms and Computer Implementations. Wiley, 1990.

[69] McShan, S. and Windle, R. The implications of hub-and-spoke routing for airline costs. Logistics and Transportation Review, 25(3):209–230, 09 1989. URL https://search.proquest.com/docview/197443739?accountid=27026.

[70] Min, H. The multiple vehicle routing problem with simultaneous delivery and pick-up points. Transportation Research Part A: General, 23(5):377386, 1989. doi: 10.1016/0191-2607(89)90085-x.

[71] Min, H., Current, J. and Schilling, D. The multiple depot vehicle routing problem with backhauling. Journal of Business Logistics, 13(1):259–288, 1992.

[72] Minieka, E. The chinese postman problem for mixed networks. Management Science, 25(7):643648, 1979. doi: 10.1287/mnsc.25.7.643.

[73] Montané, F. A. T. and Galvão, R. D. A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. Computers Operations Research, 33(3):595619, 2006. doi: 10.1016/j.cor.2004.07.009.

[74] Moudani, W. E. and Mora-Camino, F. A dynamic approach for aircraft assignment and maintenance scheduling by airlines. Journal of Air Transport Management, 6(4):233237, 2000. doi: 10.1016/s0969-6997(00)00011-9.

[75] Nagata, Y. and Bräysy, O. Edge assembly-based memetic algorithm for the capacitated vehicle routing problem. Networks, 54(4):205215, 2009. doi: 10.1002/net.20333.

[76] Nagy, G. and Salhi, S. Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. European Journal of Operational Research, 162(1):126141, 2005. doi: 10.1016/j.ejor.2002.11.003.

[77] Nash, J. C. The (dantzig) simplex method for linear programming. Computing in Science Engineering, 2(1):2931, 2000. doi: 10.1109/5992.814654.

[78] Nash, S.G. A History of Scientific Computing. ACM, 1990.

[79] Newton, Rita M. and Thomas, Warren H. Design of school bus routes by computer. Socio-Economic Planning Sciences, 3(1):7585, 1969. doi: 10.1016/0038-0121(69)90051-2.

[80] Niemansburg, S.P. Humanitarian Flight Optimization - Thesis Report. Jul. 2019.

[81] Ombuki, B., Ross, B. J. and Hanshar, F. Multi-objective genetic algorithms for vehicle routing problem with time windows. Applied Intelligence, 24(1):1730, 2006. doi: 10.1007/s10489-006-6926-z.

[82] Orloff, C. S. Routing a fleet of m vehicles to/from a central facility. Networks, 4(2):147162, 1974. doi: 10.1002/net.3230040205.

[83] Osman, I. H. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Annals of Operations Research, 41(4):421451, 1993. doi: 10.1007/bf02023004.

[84] Park, Junhyuk and Kim, Byung-In. The school bus routing problem: A review. European Journal of Operational Research, 202(2):311319, 2010. doi: 10.1016/j.ejor.2009.05.017.

[85] Pearl, J. Heuristics: Intelligent search strategies for computer problem solving. Addison-Wesley Pub. Co., Inc., 1 1984.

[86] Pincus, M. Letter to the editora monte carlo method for the approximate solution of certain types of constrained optimization problems. Operations Research, 18(6):12251228, 1970. doi: 10.1287/opre.18.6.1225.

[87] Pisinger, D. and Ropke, S. A general heuristic for vehicle routing problems. Computers Operations Research, 34(8):24032435, 2007. doi: 10.1016/j.cor.2005.09.012.

[88] Powell, W. B., Sheffi, Y., Nickerson, K. S., Butterbaugh, K. and Atherton, S. Maximizing Profits for North American Van Lines' Truckload Division: A New Framework for Pricing and Operations. Interfaces, 18(1):21–41, 1988.

[89] Prins, C. A simple and effective evolutionary algorithm for the vehicle routing problem. Computers Operations Research, 31(12):19852002, 2004. doi: 10.1016/s0305-0548(03)00158-8.

[90] Psaraftis, H. N. Dynamic vehicle routing problems. In Vehicle Routing: Methods and Studies, pages 223–248. 1988.

[91] Psaraftis, Harilaos N. A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. Transportation Science, 14(2):130154, 1980. doi: 10.1287/trsc.14.2.130.

[92] Rahoual, M., Kitoun, B., Mabed, M-H., Bachelet, V. and Benameur, F. Multicriteria genetic algorithms for the vehicle routing problem with time windows. Metaheuristic International Conference (MIC2001), pages 527–532, 2001.

[93] Reimann, M., Doerner, K. and Hartl, R. F. D-ants: Savings based ants divide and conquer the vehicle routing problem. Computers Operations Research, 31(4):563591, 2004. doi: 10.1016/s0305-0548(03)00014-5.

[94] Renaud, J., Laporte, G. and Boctor, F. F. A tabu search heuristic for the multi-depot vehicle routing problem. Computers Operations Research, 23(3):229235, 1996. doi: 10.1016/0305-0548(95)o0026-p.

[95] Rexing, B., Barnhart, C., Kniker, T., Jarrah, A. and Krishnamurthy, N. Airline fleet assignment with time windows. Transportation Science, 34(1):120, 2000. doi: 10.1287/trsc.34.1.1.12277.

[96] Rieck, J. and Zimmermann, J. A. A new mixed integer linear model for a rich vehicle routing problem with docking constraints. Annals of Operations Research, page 337358, 2010. doi: 10.1007/s10479-010-0748-4.

[97] Ryan, D. M., Hjorring, C. and Glover, F. Extensions of the petal method for vehicle routeing. Journal of the Operational Research Society, 44(3):289296, 1993. doi: 10.1057/jors.1993.54.

[98] Sankaran, J. K. and Ubgade, R. R. Routing tankers for dairy milk pickup. Interfaces, 24(5):5966, 1994. doi: 10.1287/inte.24.5.59.

[99] Sethi, S. and Sorger, G. A theory of rolling horizon decision making. Annals of Operations Research, 29(1):387415, 1991. doi: 10.1007/bf02283607.

[100] Shaw, P. A new local search algorithm providing high quality solutions to vehicle routing problems. 1997.

[101] Song, H., Hsu, V. N. and Cheung, R. K. Distribution coordination between suppliers and customers with a consolidation center. Operations Research, 56(5):12641277, 2008. doi: 10.1287/opre.1070.0488.

[102] Sánchez, L. F. H., Lomelí, L. E. C. and Martínez, F. J. Z. Approximation algorithms for the street sweeping problem. 2014 11th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), 2014. doi: 10.1109/iceee.2014.6978276.

[103] Taillard, E. D. A heuristic column generation method for the heterogeneous fleet vrp. RAIRO - Operations Research, 33(1):114, 1999. doi: 10.1051/ro:1999101.

[104] Taillard, E. D., Laporte, G. and Gendreau, M. Vehicle routeing with multiple use of vehicles. The Journal of the Operational Research Society, 47(8):10651070, 1996. doi: 10.2307/3010414.

[105] Taillard, É. D., Gambardella, L. M., Gendreau, M. and Potvin, J-Y. Adaptive memory programming: A unified view of metaheuristics. European Journal of Operational Research, 135(1):116, 2001. doi: 10.1016/s0377-2217(00)00268-x.

[106] Tarantilis, C. D. Solving the vehicle routing problem with adaptive memory programming methodology. Computers Operations Research, 32(9):23092327, 2005. doi: 10.1016/j.cor.2004.03.005.

[107] Tarantilis, C. D., Zachariadis, E. E. and Kiranoudis, C. T. A hybrid guided local search for the vehicle-routing problem with intermediate replenishment facilities. INFORMS Journal on Computing, 20(1):154168, 2008. doi: 10.1287/ijoc.1070.0230.

[108] Thomas, B. W. Waiting strategies for anticipating service requests from known customer locations. Transportation Science, 41(3):319331, 2007. doi: 10.1287/trsc.1060.0183.

[109] Thompson, P. M. and Psaraftis, H. N. Cyclic transfer algorithm for multivehicle routing and scheduling problems. Operations Research, 41(5):935946, 1993. doi: 10.1287/opre.41.5.935.

[110] Toth, P. and Vigo, D. VRP with backhauls. In The vehicle routing problem, chapter 8. Society for Industrial and Applied Mathematics, 2002.

[111] Toth, P. and Vigo, D. Vehicle routing: problems, methods, and applications (2nd Edition). Society for Industrial and Applied Mathematics, 2015. ISBN 978-1-611973-58-7.

[112] Verblunsky, S. On the shortest path through a number of points. Proceedings of the American Mathematical Society, 2(6):904904, 1951. doi: 10.1090/s0002-9939-1951-0045403-1.

[113] Vikhar, P. A. Evolutionary algorithms: A critical review and its future prospects. 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), page 261265, 2016. doi: 10.1109/icgtspicc.2016.7955308.

[114] Wade, A. C. and Salhi, S. An investigation into a new class of vehicle routing problem with backhauls. Omega, 30:479–487, 2002.

[115] Wang, X., Ruan, J., Shang, H. and Ma, C. A combinational disruption recovery model for vehicle routing problem with time windows. Intelligent Decision Technologies Smart Innovation, Systems and Technologies, page 312, 2011. doi: 10.1007/978-3-642-22194-1_1.

[116] Zheng, K., Lu, Z. and Sun, X. An effective parallel improving tabu search algorithm for heterogeneous fixed fleet vehicle routing problem. 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), 2010. doi: 10.1109/iccae.2010.5452005.