

Humanitarian Flight Optimization

Thesis Report

S.P. Niemansburg



Humanitarian Flight Optimization

Thesis Report

by

S.P. Niemansburg

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday July 5, 2019 at 15:00 PM.

Student number:	4036883	
Thesis committee:	Prof. dr. R. Curran,	TU Delft, Chairman
	Ir. P. C. Roling,	TU Delft, Daily Supervisor
	Dr. ir. J. T. van Essen,	TU Delft, Member
	Ir. W-J. C. D. Van Goethem,	Aviation Decision Sciences, Supervisor

Contents

Preface	i
Summary	ii
List of Figures	iii
List of Tables	iv
1 Introduction	1
2 Research Definition	2
3 Background	4
3.1 Humanitarian Flight Operations	4
3.1.1 Flight Scheduling Improvement Opportunity	5
3.2 Commercial and Humanitarian Airline Planning	5
3.2.1 Operations Research in Airline Planning	5
3.2.2 Fleet Planning	7
3.2.3 Route planning	8
3.2.4 Schedule Development	8
3.2.5 Aircraft Routing	10
3.2.6 Crew Scheduling	10
3.3 WFP UNHAS case	11
3.3.1 Manual Flight Routing and Scheduling procedure	11
3.3.2 UNHAS Requirements	13
3.3.3 UNHAS South Sudan Fleet and Operational Constraints	13
4 Methodology	14
4.1 Vehicle Routing Problem	14
4.2 VRP Adaptations	15
4.3 Mathematical Formulation of the Linear Program	18
4.3.1 Problem Description.	18
4.3.2 Mathematical Formulation	19
4.3.3 Decision Variables	20
4.3.4 Objective Function.	20
4.3.5 Constraints	21
4.3.6 Genetic Algorithm	23
5 Model	26
5.1 Process Flow	26
5.1.1 Pre-processing.	26
5.1.2 Optimization.	31
5.1.3 Post-processing	31
6 HFOM, UNHAS case	32
6.1 Test Data	32
6.2 Tuning of Parameters	32
6.3 HFOM Results	33
6.3.1 Day 1.	33
6.3.2 Day 2.	37
6.3.3 Day 3.	37
6.3.4 Day 4.	37
6.3.5 Day 5.	37

7	Verification and Validation	38
7.1	Model Verification	38
7.1.1	Heuristic Verification	38
7.1.2	Linear Program writing Verification	39
7.2	Model Validation	43
8	Conclusion	48
8.1	Applicability	48
8.2	Model Limitations	48
8.3	Research Question and Recommendations	49
A	Appendix A	50
B	Appendix B	52
C	Appendix C	56
D	Appendix D	59
E	Appendix E	64
F	Appendix F	76
	Bibliography	87

Preface

This thesis has been written in the first place for obtaining the degree of Master of Science at the faculty of Aerospace Engineering of Delft University of Technology. But more than that, it has been written to improve humanitarian operations. The idea for this thesis saw the light in early 2018, when Willem-Jozef Van Goethem and I interchanged ideas about optimization methods and fields to apply these in. The initial idea stems from recognising the potential to improve humanitarian air operations by combining industry knowledge with an academic research approach on specific humanitarian operations. The sharing of best practices between humanitarian specialists, commercial aviation experts and the operations research knowledge of the academic world led to a unique opportunity to significantly improve the lives of people in need around this world. This opportunity was pitched to WFP in June 2018, where it was well received and thanks to and with the help of Delft University of Technology, this idea was ultimately cultivated to the thesis that now lies before you.

First of all, I want to thank Willem, my friend, supervisor and the main creator of this idea, for his practical approach and for his outstanding knowledge of the humanitarian sector. His enthusiasm and interest were often what helped me push through when I was overcoming an obstacle. I would also like to thank Hugo Jackson for providing valuable help as an expert planner.

I want to thank Paul Roling, my TU Delft supervisor, for his guidance and advice during my thesis work.

I want to thank my parents and my sister for their support, belief and patience with me and I want to thank my friends for offering the well needed and well given distraction.

And especially, I want to thank Tessa for being sweet and caring over the last few months.

*S. P. Niemansburg
Delft, July 2019*

Summary

Humanitarian organisations have a goal to help people in need by providing help of all kinds. Supplies, like food and water, and humanitarian workers have to be transported to every place where support is required. Even remote locations, where transport over conventional roads might be impossible, have to be accessible. Aviation is the solution for these areas. UNHAS (United Nations Humanitarian Air Service) provides passenger and cargo transport for the humanitarian community. A logistic problem arises when passengers have to be transported to multiple destinations with multiple aircraft that all have different characteristics in terms of cost, capacity and range. Finding the combination of flights that create the least costly flight schedule becomes extremely complicated.

The efficiency and effectiveness of humanitarian flight scheduling can potentially be improved using optimization algorithms. This thesis addresses this issue by designing a Humanitarian Flight Optimization Model to improve both the efficiency in terms of costs and the effectiveness in terms of passengers being transported.

Literature concerning routing and scheduling for commercial airlines is available in abundance. However, commercial airlines mainly go for profit driven strategies. Humanitarian aviation is demand driven, aimed at maximizing transported people with a fixed budget. UNHAS operations in South Sudan are used as a test case to validate the model.

The problem of assigning aircraft to transportation requests is a class of vehicle routing problems, which have been researched immensely. Vehicle routing problems are NP-hard and are therefore time consuming problems to solve. The literature proposes exact algorithms, heuristics and metaheuristics to solve the vehicle routing problems. This research will focus on an exact solution method combined with a heuristic to have the program produce a near optimal solution in an acceptable amount of time.

The validation of the research shows that the model is capable of reducing the cost of flights between 2.2% to 8.4% in a typical week of operations, while having a similar demand satisfaction. On top of that, the scheduling time can be reduced by 77% on average.

Applying the Humanitarian Flight Optimization Model as a strategic decision support tool will give insight in network effects and route dependencies and will help with trading off demand satisfaction and flight routing cost in a humanitarian setting. This in turn will provide humanitarian organisations the possibility to supply increased support where it is needed most.

List of Figures

3.1 UNHAS Flight Scheduling	6
3.2 Commercial airline planning overview	7
3.3 UNHAS planning overview	7
3.4 UNHAS user categories (January-June 2018) (WFP, 2018 <i>a</i>)	11
4.1 Vehicle routing problem classification	16
4.2 Example routing solution	18
5.1 Process Flow diagram	26
5.2 Division of South Sudan regions	28
5.3 Test instance plot	30
5.4 Test instance plot, processed	30
6.1 Pareto front plot, 3/4/5 legs, day 1	34
6.2 Results of day 1, objective 79974, pax 368/371	37
7.1 Pareto front plot, day 1	45
7.2 Pareto front plot, day 2	46
7.3 Pareto front plot, day 3	46
7.4 Pareto front plot, day 4	47
7.5 Pareto front plot, day 5	47
A.1 Flight Tasking Officer decision flowchart	51
B.1 Daily requests test data, 5 days	53
B.2 Distance matrix, a	54
B.3 Distance matrix, b	55
E.1 Results of day 2, objective 91918, pax 291/298	64
E.2 Results of day 3, objective 72519, pax 287/293	67
E.3 Results of day 4, objective 87414, pax 305/305	70
E.4 Results of day 5, objective 92773, pax 417/420	73
F.1 Planner 1 results, day 1	77
F.2 Planner 1 results, day 2	78
F.3 Planner 1 results, day 3	79
F.4 Planner 1 results, day 4	80
F.5 Planner 1 results, day 5	81
F.6 Planner 2 results, day 1	82
F.7 Planner 2 results, day 2	83
F.8 Planner 2 results, day 3	84
F.9 Planner 2 results, day 4	85
F.10 Planner 2 results, day 5	86

List of Tables

3.1	Differences in industry characteristics	4
3.2	Nomenclature FAM	9
3.3	UNHAS aviation field operations performance (January-June 2018) (WFP, 2018 <i>a</i>)	11
3.4	Fleet used by UNHAS in South Sudan	13
4.1	Nomenclature CVRP	17
4.2	Nomenclature HFOM	20
4.3	Random Key chromosome representation	23
4.4	Decoded route	24
4.5	Two-point crossover at 3 and 5	24
4.6	Routes of parents and children	25
4.7	Mutation for request 1	25
5.1	Decision on spillage	28
5.2	Test instance data	30
5.3	Test instance data, processed	30
6.1	Results HFOM using CPLEX and Gurobi	34
6.2	Flight schedule for day 1	35
6.3	Transport breakdown for day 1	36
7.1	Pre-processing, dividing and splitting requests, day 1	39
7.2	Heuristic verification	39
7.3	Test instance used for verification	39
7.4	Test instance processed	39
7.5	Verification of the objective function	40
7.6	Verification of customers served constraint	40
7.7	Verification of start and end hub constraint	40
7.8	Verification of flow conservation constraint	41
7.9	Verification of same vehicle served constraint	41
7.10	Verification of first timing constraint	41
7.11	Verification of second timing constraint	42
7.12	Verification of third timing constraint	42
7.13	Verification of first loading constraint	42
7.14	Verification of second loading constraint	42
7.15	Verification of first range constraint	43
7.16	Verification of second range constraint	43
7.17	Verification of runway constraint	43
7.18	Overview of expert system validation	45
C.1	Routing cost for different demand satisfactions with a maximum of 3 legs in a routing for day 1	56
C.2	Difference in routing cost for different demand satisfactions with a maximum of 3 legs in a routing	56
C.3	Difference in routing for different demand satisfactions with a maximum of 3 legs in a routing	57
C.4	Routing cost for different demand satisfactions with a maximum of 4 legs in a routing for day 1	57
C.5	Difference in routing cost for different demand satisfactions with a maximum of 4 legs in a routing	57
C.6	Difference in routing for different demand satisfactions with a maximum of 4 legs in a routing	57
C.7	Routing cost for different demand satisfactions with a maximum of 5 legs in a routing for day 1	58
C.8	Difference in routing cost for different demand satisfactions with a maximum of 5 legs in a routing	58
C.9	Difference in routing for different demand satisfactions with a maximum of 5 legs in a routing	58

D.1	Pre-processing, dividing and splitting requests, day 1	59
D.3	Pre-processing, dividing and splitting requests, day 2	60
D.5	Pre-processing, dividing and splitting requests, day 3	61
D.7	Pre-processing, dividing and splitting requests, day 4	62
D.9	Pre-processing, dividing and splitting requests, day 5	63
E.1	Flight schedule, day 2	65
E.2	Request transport, day 2	66
E.3	Flight schedule, day 3	68
E.4	Request transport, day 3	69
E.5	Flight schedule, day 4	71
E.6	Request transport, day 4	72
E.7	Flight schedule, day 5	74
E.8	Request transport, day 5	75



Introduction

This thesis investigates the implementation of a flight routing and scheduling model for humanitarian airlines. Can the effectiveness and efficiency of humanitarian airlines be increased by using such a model?

Since the establishment of the World Food Programme (WFP) in 1961 as the food-assistance branch of the United Nations (UN) it has grown to be the world's leading humanitarian organization in addressing the challenges of global hunger and nutrition (WFP, 2017*b*). WFP is active in around 80 different countries, assisting 80 million people by delivering food assistance in emergencies and working with communities to improve their daily life (WFP, 2018*b*).

WFP manages the United Nations Humanitarian Air Service (UNHAS), which offers "safe, reliable, cost-efficient and effective passenger and light cargo transport for the humanitarian community" (WFP, 2018*b*). On any day 92 planes are on the move, delivering life-saving assistance to those in need and transporting humanitarian workers where they are needed the most. As such WFP can be considered a major airline, albeit with a humanitarian mandate. UNHAS is a non-commercial humanitarian air service provider. Commercial passenger airline schedule optimisers are not usable within a humanitarian setting, as they optimise the flight schedule depending on seasonal demand forecasts rather than actual daily demand. Therefore, UNHAS, in its humanitarian setting, currently creates manually the daily flight schedule based on the actual demand, known only one day prior to the day of operation.

There is an opportunity for UNHAS to be more efficient and effective in their flight scheduling by optimizing their daily operations. A mathematical flight scheduling and routing model can be built that captures the humanitarian objective and specific constraints. This opportunity is used to come up with a research project on the topic of humanitarian flight schedule optimisation.

This thesis will present a model that is capable of creating routings and schedules in a humanitarian setting. The research objective and research question are provided in Chapter 2. Chapter 3 gives an overview of all the operations involved in the airline planning process for both commercial airlines and humanitarian air service providers. The methodology is described in Chapter 4. Chapter 5 focusses on the optimization model, appropriately named the Humanitarian Flight Optimization Model (HFOM). The model is tested with a real life case in Chapter 6 and the verification and validation is performed in Chapter 7. Finally, the results are discussed in Chapter 8.

2

Research Definition

This thesis delivers a report concerning a humanitarian flight routing and scheduling model. The humanitarian setting is non-commercial, focused on providing air transport to humanitarian workers and delivering humanitarian supplies to people in need. The main drivers in the routing and scheduling are demand satisfaction (effectiveness) and cost minimization (efficiency), instead of profit maximization as is the case for a commercial airline. In order to improve the flight scheduling done by humanitarian air service providers, a flight scheduling model will be designed and the model will be compared with the manual planning as is done at this moment. This can be captured by the following *Research Objective*:

To improve the efficiency and effectiveness of flight routing and scheduling in a humanitarian setting taking into account the operational and safety constraints specific to non-commercial humanitarian air transport.

In order to reach this objective, the following *Research Question* and sub-questions need to be answered:

Can a daily flight scheduling model be developed that improves the efficiency and effectiveness of flight routing and scheduling in a humanitarian setting?

1. *How to develop a daily routing and scheduling model for a humanitarian air service provider?*
Daily routing and scheduling models are not being used in commercial airlines. Flight routes and subsequent schedules are known months in advance.
2. *What are the constraints involved in humanitarian flight scheduling?*
The humanitarian setting is often a difficult environment, where operational and political factors are key drivers in deciding where and how often to fly.
3. *Which models are used to build daily routing schedules?*
It is of importance to know which models are being used to create daily routing schedules. Not only in aviation, but also more in general e.g. vehicle routing.
4. *Can the routing models be combined to create a model that works in the humanitarian environment?*
The individual routing models need to be combined to suit the needs of humanitarian operations.
5. *Can the daily scheduling operation be improved?*
In order to achieve a more efficient and effective daily operation, it is of interest to perform a comparison between the current manual scheduling versus the automatic scheduling of the model. How do expert systems fare against exact methods?
6. *How to balance the optimal daily flight schedule efficiency versus the monthly overall efficiency?*
Due to contracting constraints, the leased aircraft need to be utilized a certain amount of hours in a month, known as minimum guaranteed hours (MGH). For optimal efficiency the daily utilization needs to be linked to the monthly utilization.
7. *Can an optimized model still allow sufficient routing and scheduling flexibility for emergency missions?*
Forced flights might occur due to a humanitarian emergency in a certain area.

Theoretical Content/Methodology

The hypothesis to be proven is that the efficiency and effectiveness of humanitarian flight service providers can be improved by using a flight routing and scheduling model. This hypothesis will be assessed by creating the model and comparing the performance to the manual scheduling. The model is designed by combining multiple vehicle routing models created for the purpose of optimization truck dispatching and converting that to suit the needs of a humanitarian organisation. The input is a daily demand matrix with origin and destination for pax. The output of the model is a routing and time schedule for each aircraft for a given day.

Experimental Set-up

The research will be executed at the faculty of Aerospace Engineering of the Delft University, department Air Transport Operations. The supervisors involved are Paul Roling and Willem-Jozef Van Goethem (Aviation Decision Sciences). Paul is a lecturer and researcher at the Air Transport Operations Department and focus mainly on operations research. Willem is an Aerospace Engineering alumnus, who has worked for UNHAS in South Sudan, Nepal and Haiti as a flight tasking officer. Currently, he is a trainer in Lean Six Sigma providing companies education in continuous improvement.

The data used for the research is provided by UNHAS and contains origin-destination demand requests (flat table with date, origin, destination, pax). All data is confidential.

The optimization model is created using an Application Programming Interface (API), which will be Python (*Python*, 2019) in this case. The optimization engines CPLEX (*IBM ILOG CPLEX Optimization Studio*, 2019) and Gurobi (*Gurobi Optimizer*, 2019) will be called from the API. CPLEX and Gurobi are used because both offer a powerful engine and TU Delft has a student license. The integration with Python is the preferred one, as it is a programming language that pairs extremely well with CPLEX and Gurobi. The docplex (*IBM Decision Optimization CPLEX Modeling for Python*, 2019) and gurobipy (*The Gurobi Python Interface for Python Users*, 2019) packages for Python, for CPLEX and Gurobi respectively, allow for fast, flexible and powerful optimization.

Results, Outcome and Relevance

The research will result in a daily routing and scheduling model suited to be used in a non-commercial humanitarian setting. The objective (cost minimization, demand satisfaction maximization) and constraints (operational, contractual) give the model a unique position in the field of vehicle routing problems.

By creating a flight routing and scheduling model that can be used for non-commercial humanitarian air service providers, the efficiency and effectiveness of daily operations can be increased. This increase in efficiency (cost reduction) and effectiveness (demand satisfaction increase) gives humanitarian organisations the possibility to increase the level of help they can give to people in need. In the case of WFP this performance optimization will be a boost for their goal of zero hunger worldwide.

3

Background

This Chapter provides an overview of the difference between the airline planning process for commercial airlines and humanitarian air service providers. Firstly, the humanitarian flight operations will be discussed, along with the opportunities that exist for improving these operations. Next, the airline planning process will be elaborated upon. Finally, the operations of WFP in South Sudan are analysed.

3.1. Humanitarian Flight Operations

The goal of humanitarian air logistics is to deliver aid where it is needed, no matter how hard it is to reach the population in need. Aviation is a key transport modality that enables humanitarian workers and cargo to reach even the most cut-off areas, that can't be accessed through any other mean of transport. However, air operations are vastly more expensive compared to road transport. Using optimised flight scheduling, the humanitarian community could potentially save millions of dollars per year. This money can be used to reach the hard to reach locations with more aid for the same price tag.

Operations research has been used to create optimised flight routes and schedules for commercial airlines to maximize revenue and become profitable in an industry where the margins are very small (Belobaba et al., 2009). The research involving flight scheduling for humanitarian air service providers is non-existent. The expertise has simply not existed before. Models that are used in a commercial setting need to be adapted to work for a humanitarian airline. Table 3.1 shows the main differences between the commercial and humanitarian setting.

Commercial	Humanitarian
Profit maximization	Cost minimization/Access maximization
Market driven	Demand driven
Economic climate	Country/emergency specific demand
Seasonal demand	Unexpected demand peaks
Long aircraft delivery times	Wet lease
Market funded	Donor funded

Table 3.1: Differences in industry characteristics

The United Nations Humanitarian Air Service (UNHAS) is a non-commercial humanitarian air service provider. Other humanitarian air service providers are Mission Aviation Fellowship (MAF) and ECHO Flight. UNHAS does not only serve the World Food Programme (WFP), but also other UN agencies and non-governmental organisations (NGOs) (Dorn and Cross, 2014). Their mandate is to offer "safe, reliable, cost-efficient and effective passenger and light cargo transport for the wider humanitarian community to and from areas of crisis and intervention" (WFP, 2018*b*). UNHAS contracts over 90 aircraft and helicopters, instead of owning them. The service they provide consists of air support for WFP (airlifts and food drops) and air service for the humanitarian community (WFP, 2018*a*). Because of the risks involved in operating in violent and unstable conditions, operations according to international safety standards are often not available in the areas where humanitarian assistance is greatest. This in turn means that the humanitarian air service providers are often the only means of reaching a remote place. According to the operation snapshot of WFP (2018*a*), operational

agility is a “key pillar of the WFP Aviation Strategy”. Across all operational elements in aviation, quick response is a necessity. All the aviation activities contribute to the sustainable goals that are set by WFP and NGOs. As flight scheduling is done on a daily basis for humanitarian air service providers, a quick reaction and scheduling flexibility is also vital in this area.

3.1.1. Flight Scheduling Improvement Opportunity

Currently UNHAS, in its humanitarian setting, creates the daily flight schedule manually based on the actual demand, known only one or two days prior to the day of operation. The current flow of a typical UNHAS flight scheduling process can be found in Figure 3.1. It can be seen that an improvement in efficiency of the flight planning process can be made, since this process generally takes on average 400 minutes to complete, thus creating a bottleneck.

Not only will the efficiency of the flight planning process benefit from a schedule optimiser, the effectiveness of the routes to be flown will also increase. More passengers and cargo can be transported by using an optimal, cost minimizing aircraft routing compared to suboptimal manual scheduling. Commercial passenger airline schedule optimisers are not usable within a humanitarian setting, as they optimise the flight schedule depending on seasonal demand forecasts rather than actual daily demand driven by the humanitarian emergency. However, a strong similarity in the traffic demand exists between UNHAS operations and cargo flight service operations for online retailers such as Amazon. These companies recently started to automatically optimise their daily flight schedule based on the actual known demand of packages which are ordered online, and which need to be delivered within two days of the reception of the online order. This development in the air cargo industry is in line with the general growing trend for flight schedule optimisers used in commercial passenger aviation (Clough et al., n.d.).

Nowadays, all major and mid-sized airlines use flight schedule optimisers to satisfy traffic demand and drive down costs. Humanitarian air service providers with considerable fleet sizes can be compared to major airlines, albeit with a humanitarian mandate having a similar complicated flight scheduling task, while often operating in extremely challenging environments. This means that there are currently no off-the-shelf flight optimisation packages available which are suitable to UNHAS. The use of a tailored flight schedule optimiser could significantly improve UNHAS’ effectiveness and cost efficiency.

3.2. Commercial and Humanitarian Airline Planning

As previously mentioned, UNHAS can be seen as a large airline but it is inherently different because of the humanitarian setting it operates in. The main objective is not to be profitable but to deliver cargo and assistance in the form of workers. Therefore, the airline planning process is different as profitability is not the leading factor, but demand satisfaction and accessibility. However, this does not mean that minimizing cost should be a neglected factor. The cost savings made by using an optimal flight schedule can be used to further enhance the effectiveness and will in turn increase the aid that can be delivered by WFP. Thus, an optimal flight schedule is preferred and should be strived for.

3.2.1. Operations Research in Airline Planning

Operations research in the field of airline operations started on a large scale in the 1960’s with the first AGIFORS (Airline Group of the International Federation of Operational Research Societies). Since then, a tremendous growth can be seen in using operations research activities, most prominently in the schedule development. With the rise of the computer and computing power, the mathematical models that were devised could be translated to computer programs and subsequently solved. Nowadays, with the advancement in computational power, even larger problems can be solved with more sophisticated algorithms. The flight schedule plays a central role in an airline’s planning process and optimization of this process is key to finding the most efficient and effective utilization of resources.

The early work was mostly on (1) direct and (2) stepwise approaches for flight schedule optimization (Etschmaier and Mathaisel, 1985). The direct approach involved a heuristic procedure to prepare a schedule by choosing flights in sequence and if needed changing flights that were previously selected. The stepwise approach was more elaborate and is more closely related to what is done in the airline planning of today. Firstly, the routes to be served are selected, the frequency of flights is determined and the departure times are set (Gagnon, 1967; Simpson, 1966). Afterwards, the departure times are checked for operational feasibility and the aircraft rotation planning is developed (Labombarda and Nicoletti, 1971; Richter, 1968; Simpson, 1969). A difference can be seen in the research that is done now and the work that was done 50 years ago. Airlines nowadays have

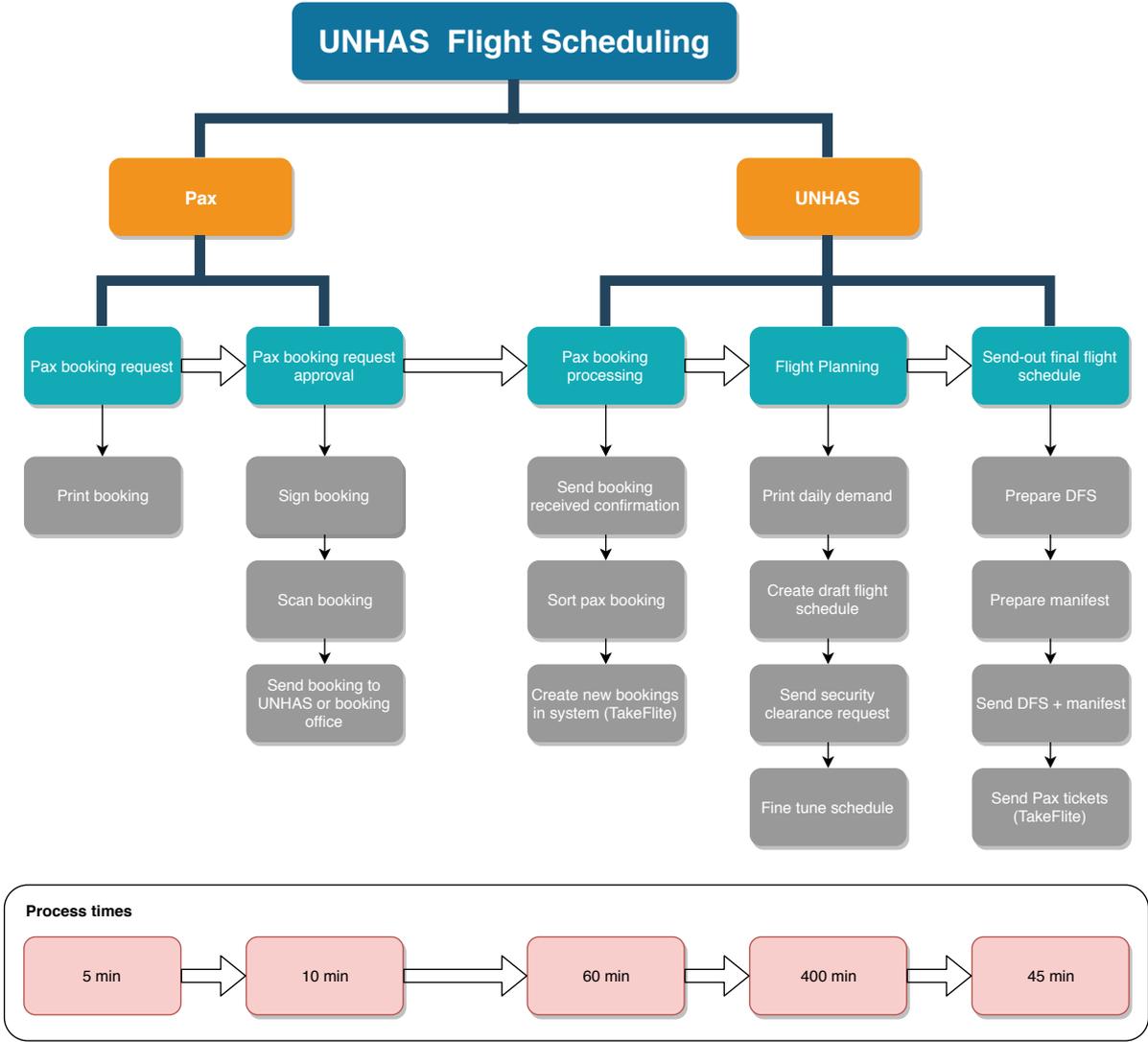


Figure 3.1: UNHAS Flight Scheduling

a ‘fixed’ schedule which can change slightly, but not drastically (e.g. airport slots, crew requirements). Most operations research is done on fleet assignment, maintenance rotation and crew scheduling. For large airlines this is where the impact is made. Start-up airlines however need to still figure out where to fly (network development) and make a schedule (timetable development).

According to Belobaba et al. (2009) the commercial airline planning process can be divided in 3 distinct stages, being:

1. Fleet Planning (**Section 3.2.2**)
2. Route Planning (**Section 3.2.3**)
3. Schedule Development (**Section 3.2.4**)

Commercial airlines follow the flow shown in Figure 3.2 in making the decision what aircraft to fly, where to fly and when to fly. Long-term strategic decisions are made during the fleet planning phase and medium-term decisions relate to route planning and scheduling (Belobaba et al. (2009)).

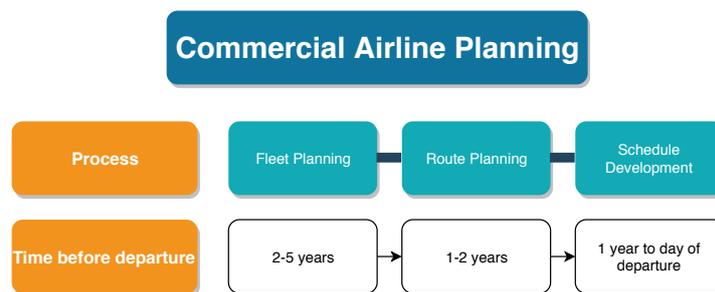


Figure 3.2: Commercial airline planning overview

Humanitarian air operators differ from commercial airlines primarily in the first two categories. The UNHAS fleet has been determined already and it mainly consists of wet leased aircraft, which makes for a flexible fleet. Route planning is different to UNHAS as opposed to a commercial airline in that UNHAS does not have to find a route that is profitable, but the chosen route is driven by demand; if humanitarian assistance is needed in place B, a route will be opened to supply destination B. The main opportunity for UNHAS lies in the schedule development. Flight schedules are generated manually, which leads to suboptimal routes being flown and a suboptimal allocation of resources. Operations research techniques can be used for planning and optimizing operations and will be the main focus of this study. Figure 3.3 illustrates the planning process for UNHAS.



Figure 3.3: UNHAS planning overview

A summary of the fleet planning, route planning and schedule development will be given and afterwards a dive will be made in the field of operations research models focused on aircraft routing and flight scheduling.

3.2.2. Fleet Planning

Fleet planning is the process of an airline to determine what and how many aircraft should be operated to meet the needs of the airline (Belobaba et al., 2009). An airline’s fleet is comprised of “the total number of aircraft that an airline operates at any given time, as well as by the specific aircraft types that comprise the total

fleet” (Belobaba et al., 2009). An airline’s decision to buy a new aircraft or retire an existing one has an impact on the financial situation and on the routes that can be flown. A huge capital investment needs to be made for a long-term payout. Although large costs are associated with fleet planning as well as long-term impact of the decisions on the airline’s operations, the level of the decision support tools is significantly lower than that of scheduling optimization models. The main reason for this is the uncertainty that comes with decisions made for 10-20 years in the future (Belobaba et al., 2009). Even a very sophisticated tool would not be able to handle the unpredictability of economic changes over a large timespan. Several key factors that influence the decision on the fleet plan include the performance characteristics of the aircraft, the fleet commonality, economics and the environment (Belobaba et al., 2009). The fleet planning in the humanitarian sector is vastly different from the fleet planning for the commercial airlines. The fleet is determined first by the Contracting Unit (CU) based on a certain budget and afterwards the known fleet of aircraft are distributed in such a way that the most efficient and effective routes can be flown. The Chief Air Transport Officer (CATO) on a specific mission defines how the fleet can be best used to transport the most passengers. The CATO tracks the demand and which aircraft are most used and provides this as feedback to the CU. The CATO, who has the knowledge of the operations, supplies the CU with information of what type of aircraft are needed in terms of range, capacity and runway required. The CU then decides which specific aircraft will be leased. Most contracts are short term wet lease contracts, which gives a lot of flexibility for the humanitarian operator to change the fleet swiftly. This might be needed in a case of a sudden or unpredicted emergency.

3.2.3. Route planning

Traditionally once a fleet plan had been established, a planning is made for the routes to be flown. The main drivers in this decision are economic considerations and the expected profitability (Belobaba et al., 2009), further constrained by landing rights, airport slots and aircraft ground handling. What is also done nowadays for commercial airlines is identifying a profitable route and changing the fleet plan accordingly. Both these methods should be used simultaneously in order to capture the most profitable routes.

As stated before, fleet and route planning are vastly different for a humanitarian air service provider compared to a commercial airline. The fleet consists mainly of wet leased aircraft and routes are chosen not for profitability but for humanitarian needs. The route planning is done daily based on the demand that exists for a route. Therefore, the route planning part for a humanitarian airline becomes a complicated vehicle routing problem.

3.2.4. Schedule Development

The 1990’s saw a boom in especially the fleet assignment optimization. Mathematical programming algorithms and continued increase in computing power made it possible to solve optimization problems of large scale during that time. The fleet assignment started with several papers (Abara, 1989; Subramanian et al., 1994) and the fleet assignment model (FAM) that is widely used nowadays was formulated by Hane et al. (1995). All models that revolve around FAM go back to this basic version. Models that integrate fleet assignment, aircraft maintenance routing and crew scheduling are being researched as well (Clarke, Hane, Johnson and Nemhauser, 1996; Lohatepanont and Barnhart, 2004; Papadakos, 2009). Schedule development is the process that involves around five interrelated tasks being:

1. Frequency Planning
2. Timetable Development
3. Fleet Assignment
4. Aircraft Routing
5. Crew Scheduling

Each of these topics will be discussed in the following sections. The schedule is developed in iterative way which uses the routes and the fleet of aircraft to be flown as an input. Each step of the schedule development is being assessed and evaluated as the schedules progresses to its final form. Since the schedule development is fully integrated in the route planning process for the humanitarian air service, only an overview of the tasks to be performed will be given for completeness (Section 3.3.1).

Frequency Planning & Timetable Development

The schedule development starts roughly a year before the flight and continues to be updated until the flight departs. The frequency planning is executed based on the output of the route planning process. Timetables are developed simultaneously and involve the creation of a specific timetable for flight departures. Because the development of the flight scheduling is complex, with many variables involved, there is no model yet to fully capture the flight scheduling (Grosche et al., 2001). It is an iterative process that is prone to changes. The parts where mathematical models can be used to optimize operations are in the fleet assignment, aircraft routing and crew scheduling (see Sections 3.2.4, 3.2.5 and 3.2.6 respectively).

Frequency planning is vital in deciding what the market share of an airline will be. Increasing the frequency of flights will improve the market share and offer more flights to the passengers. However, an airline is not able to quickly increase the frequency of flights. It is constrained by the availability of aircraft and the airport slots (Belobaba et al., 2009). Typically, demand forecasts and expected market share are used to achieve a 'baseline' frequency (Belobaba et al., 2009). After the frequency planning is completed, the specific timetable of flight departures is developed. A balance should be found in maximizing aircraft utilization and providing flights at peak periods. Turnaround times should be kept at a minimum, but sometimes it might be more beneficial to wait for a peak in demand. Because of the huge amount of variables involved in the timetable development (e.g. airport slot times, arrival/departure times, operational constraints), there still does not exist a model that can give an optimal solution to this problem (Grosche et al., 2001). Most airlines have a timetable already and the computer models being developed are for the purpose of making incremental changes to the existing timetable (Belobaba et al., 2009).

Fleet Assignment

Fleet assignment deals with assigning the right fleet to a flight given a flight schedule, a fixed fleet and a network of routes and has been a hot topic for research (Abara, 1989; Barnhart et al., 2002; Hane et al., 1995; Rexing et al., 2000; Rushmeier and Kontogiorgis, 1997; Subramanian et al., 1994). The focus lies on optimizing a certain objective (maximize profit, minimize cost) while meeting several operational constraints. Abara (1989) gives one of the first formulations for the fleet assignment problem as an integer linear programming model and solves it based on data provided by American Airlines. The model incorporates multiple fleets to be used using a given flight schedule and the objective can vary from profit maximization to optimal utilization of a certain fleet. Subramanian et al. (1994) provides a mixed-integer linear program that "assigns fleet types to flight legs as to minimize a combination of operating and passenger 'spill' costs, subject to a variety of operational constraints". The most famous and often referred to as the *basic* fleet assignment model (FAM) is formulated by Hane et al. (1995). The model utilizes a time-space network to select aircraft paths and assigns aircraft types to flight legs. An overview of fleet assignment models and algorithms can be found in Sherali et al. (2006). The definition and the mathematical model of the FAM (based on Hane et al. (1995)) is given below:

The **nomenclature** for the FAM can found in Table 3.2. This nomenclature is based on the definitions used in Hane et al. (1995), Bazargan (2010) and Belobaba et al. (2009).

<i>Parameters</i>		<i>Sets</i>	
n	number of nodes	F	set of flight legs
n^+	ground arc originating at node n	K	set of fleet types
n^-	ground arc terminating at node n	N^k	set of nodes for fleet k in time-space network
c_f^k	cost of assigning aircraft of type k to flight leg f	G^k	set of ground arcs for fleet k in time-space network
M^k	number of aircraft of type k	$O(k, n)$	set of flight legs originating at node n in fleet k 's time-space network
<i>Variables</i>		$I(k, n)$	set of flight legs terminating at node n in fleet k 's time-space network
x_f^k	$= \begin{cases} 1, & \text{if flight leg } f \text{ is assigned to fleet } k \\ 0, & \text{else} \end{cases}$	$CL(k)$	set of flight legs for fleet k
y_a^k	number of aircraft of type k on ground arc a	$CG(k)$	set of ground arcs for fleet k

Table 3.2: Nomenclature FAM

Based on the formulation for the FAM by Hane et al. (1995) and Belobaba et al. (2009), the following multi-commodity network flow **FAM formulation** is presented:

$$\min \sum_{f \in F} \sum_{k \in K} c_f^k x_f^k \quad (3.1)$$

subject to:

$$\sum_{k \in K} x_f^k = 1 \quad \forall f \in F \quad (3.2)$$

$$y_n^k + \sum_{f \in O(k,n)} x_f^k - y_n^k - \sum_{f \in I(k,n)} x_f^k = 0 \quad \forall n \in N^k, k \in K \quad (3.3)$$

$$\sum_{a \in CG(k)} y_a^k + \sum_{f \in CL(k)} x_f^k \leq M^k \quad \forall k \in K \quad (3.4)$$

$$x_f^k \in \{0, 1\} \quad \forall i \in F, k \in K \quad (3.5)$$

$$y_a^k \geq 0 \quad \forall a \in G^k, k \in K \quad (3.6)$$

$$(3.7)$$

The objective is to minimize the operating cost. Constraint (3.2) forces each flight leg to be flown by one fleet type. Constraint (3.3) balances number of aircraft arriving and departing at each airport and constraint (3.4) ensures that the number of assigned aircraft does not exceed the amount of aircraft in the fleet of that type. The binary constraint (3.5) and the non-negativity constraint (3.6) complete the FAM.

3.2.5. Aircraft Routing

The output from the fleet assignment is the type of aircraft to be flown for each flight leg. However, individual aircraft have not yet been taken into account. This assignment of individual aircraft within each fleet to flight legs is done during aircraft routing. Individual aircraft are also referred to as tail numbers and aircraft routing is commonly referred to as aircraft rotation, aircraft assignment or tail assignment. According to Bazargan (2010) aircraft routing is ultimately intended to maximize revenue or to minimize operating cost while taking into account the following points: (1) flight coverage, (2) aircraft load balance and (3) maintenance requirements (Clarke, Johnson, Nemhauser and Zhu, 1996; Papadakos, 2009).

Several methods for aircraft routing exist. Papadakos (2009) gives an integrated model for airline scheduling optimization (combining fleet assignment, aircraft maintenance routing and crew scheduling). Talluri (1998) proposes a solution for the routing problem while having the aircraft required to stay overnight at a maintenance station after at most four days. Argüello et al. (1997) and Bard et al. (2001) both discuss how aircraft routings should adapt when groundings or delays occur (disruption management). Desaulniers et al. (1997) dives deeper into daily aircraft routing and scheduling. Aircraft routings are based on network flow models, such as the set-partitioning model and the travelling salesman problem. The airline planning is currently moving towards integrated models. The combination of fleet assignment and aircraft routing is proposed in several articles (Barnhart et al., 1998; Papadakos, 2009; Sherali et al., 2006).

3.2.6. Crew Scheduling

The crew scheduling is the final part of the schedule development process and starts when the aircraft are assigned to all flight legs in the schedule. It involves designating cockpit and cabin crews to aircraft. The crew scheduling is generally split up in two parts, the *crew pairing* and the *crew rostering*. Crew pairing deals with creating a "sequence of flight legs that starts and ends at the same crew base" (Bazargan, 2010). Crew rostering focuses on combining the pairings into larger rosters. Individual crew member's needs and preferences are taken into account in this process. The objective of the crew scheduling is to minimize the cost of the crew, while maximizing crew satisfaction and adhering to union and government rules as well as operational constraints (Barnhart, 2008). Desrosiers et al. (1995) and Klabjan et al. (2001) provide solutions to solving large scale crew scheduling problems. Crew scheduling has been incorporated in the FAM successfully and is being researched further by Barnhart et al. (1998); Clarke, Hane, Johnson and Nemhauser (1996); Cohn and Barnhart (2003); Cordeau et al. (2001); Klabjan et al. (2002).

3.3. WFP UNHAS case

The Humanitarian Flight Optimization Model that will be addressed in this paper has to be tested against a real specific industry problem. For this matter the case of WFP's UNHAS is taken as the test subject. The stakeholders involved, the current situation and the methods used to solve the daily problem will be assessed and the operational constraints for the flight scheduling will be analyzed. UNHAS provides "air services to all humanitarian actors in some of the world's most remote and challenging locations." (WFP, 2017a). They also state to ensure "the provision of safe, reliable, effective and cost-efficient passenger and light cargo transport to the wider humanitarian community, development actors and donors" (WFP, 2018a). The air service they provide is for all organisations that support people in sudden onset emergencies or long-lasting humanitarian crises, like South Sudan.

The UNHAS case revolves around their flight operations in South Sudan. It is UNHAS' biggest country served in terms of user organisations served, destinations flown and passengers and cargo transported (WFP, 2018a). Figure 3.4 gives an overview of the overall users of UNHAS operated flights in the first half of 2018 and Table 3.3 gives an overview of UNHAS field operations globally and in South Sudan. With 56 destinations served and more than forty-two thousand passengers flow in six months (roughly 1600 passengers a week, 320 passengers a day), the logistics involving the transport of all the requests in a cost-efficient way is complicated. According to WFP (2017a), UNHAS managed to serve 96% of all incoming passenger requests in 2017. Up until this point all the routing and scheduling has been done manually by flight planners, of which the procedure will be shown in Section 3.3.1



Figure 3.4: UNHAS user categories (January-June 2018) (WFP, 2018a)

UNHAS Global	UNHAS South Sudan
15 countries	282 user organisations served
286 destinations	56 destinations
173,652 passengers	42,504 passengers
1,341 mt of cargo	537 mt cargo

Table 3.3: UNHAS aviation field operations performance (January-June 2018) (WFP, 2018a)

3.3.1. Manual Flight Routing and Scheduling procedure

The Flight Tasking Officer (FTO) is in charge of creating a daily flight routing plan. The decision making process is a heuristic step by step approach, but heavily depends on the experience of the FTO. Most FTOs use the following decision steps or a similar heuristic (the flowchart of the process can be found in Appendix A, Figure A.1):

1. Check fleet availability. (Are aircraft grounded due to technical issues, maintenance, crew availability)
2. Check daily Passenger Requests (Pax numbers on all Origin-Destination requests)
3. Check airfield availability. (Are airports unavailable due to bad runway conditions?, recent pilot reports, security related issues)
4. If requested, schedule dedicated special flights (VIP, charter, Medevac)
5. (a) Identify potential connection passengers and O-D with a potential connection in Rumbek. (typical Airfield include: Leer, Nyal Ganyiel, Pagil, Koch, Mayendit, Jiech, Gorwai, Mogok, Old Fangak, Agok)

- (b) Schedule first rotation of Fokker 50 for bringing connections to Rumbek. Fill up Fokker 50 with terminating pax for Wau, Aweil, Rumbek.
 - (c) Schedule Rumbek flights of the two Mil Mi8's and the two Cessna 208's to serve the connecting passengers and pick up connections. Inbound Juba connections are often picked up on return leg Fokker 50 from Wau/Aweil or picked up separately by Dash 8 bringing remaining terminating passengers for Rumbek.
 - (d) Determine whether 2nd rotation with Fokker 50 is required, depending on pax demand.
 - (e) If needed schedule 2nd rotation of Fokker 50.
6. (a) Identify potential routing combination to be served via Bor. Typical destinations include Bor, Mingkaman, Pibor, Mabior.
 - (b) Schedule Mil Mi8 from Juba (typically 1 or 2 rotations due to high pax demand around Mingkaman and travel between Bor and Mingkaman)
 - (c) Schedule Mil Mi8 or Cessna 208.
7. (a) Check for potential passengers combination for Dash 8 flights.
 - (b) Identify potential routing combination (Paloich, Maban, Malakal) for Dash 8 flights.
 - (c) Schedule Dash 8's first rotations.
 - (d) Identify potential routing combination Yambio Yei, for Dash 8, Dornier 228, Cessna 208 depending on demand.
 - (e) Schedule Dash 8 second rotation.
 - (f) Identify Rubkona direct flights, for Dash 8, Dornier 228 or Cessna 208 depending on demand.
8. (a) Identify potential routing combination Kapoeta, Torit for Dornier 228 or Cessna 208 depending on demand.
 - (b) Schedule Dornier 228 or Cessna 208.
9. (a) Identify potential routing combination for Motot, Lankien and Waat, most likely served by Mil Mi8 from Juba.
 - (b) Schedule Mil Mi8 or Cessna 208.
10. (a) Identify potential routing combination Yida, Ajuong Thok for Cessna 208 or Dornier 228 depending on demand.
 - (b) Schedule Cessna 208 or Dornier 228.
11. (a) Identify Renk direct flight depending on payload range. The flight would require a fuel stop in either Bor or Malakal depending on price of fuel in Bor and Malakal.
 - (b) Schedule Cessna 208.
12. Aircraft Availability check. Check whether the intended flight is covered by a specific tail number and whether by his time the aircraft did not develop a technical grounding to the next day. If needed adjust the schedule and/or tail number.
13. (a) Adjust Scheduled Departure Timings (SDT)
 - (b) Adjust SDT timings of Dash 8 with 15 minutes separation after Fokker 50 flight, because Dash 8 are often used in 2 rotations.
 - (c) Adjust SDT timings for Rumbek based flight to the Scheduled Arrival Timings (SAT) of Fokker 50 to ensure connection.
 - (d) Adjust SDT Fokker 50 on Rumbek-Juba return leg to accommodate inbound connections.
 - (e) Overall check morning SDT's for Juba and Rumbek on 15 minutes separation.
 - (f) Overall check Afternoon SDT's for 2nd rotations on 15 minutes separation.
 - (g) Overall check SAT late afternoon.
14. (a) Check payload, range and fuelling constraints.
 - (b) Check depending on Payload Range Diagram if all aircraft can carry the scheduled pax loads.
 - (c) Adjust routing for fuelling stops and/or reduce passenger numbers on specific flight legs. This step is often already taken into account during scheduling steps for the Fokker 50 and long Mil Mi8 flight from Juba and Rumbek.

- Optional: check tailnumbers on monthly minimum guaranteed hours. Adjust the tail assignment accordingly.

The above steps are to be seen as general guidelines to construct a feasible schedule quicker. They are not set in stone but require often an iterative approach until a feasible schedule is found.

3.3.2. UNHAS Requirements

The objective of the flight scheduling is to minimize pax and cargo requests not served (maximize demand satisfaction) at lowest routing cost (minimize cost). The requirements for this are given below:

- Daily changing O-D demand:** Schedule has to be made 48-24hrs before departure.
- Maximum flight time and minimum turnaround time (TAT):** All aircraft have 10 hours of operation daily with a minimum of 20 minutes ground time. Each aircraft has to start and end at one of the two hubs.
- Operational constraints of the aircraft and airports:** Runway required, capacity and range for the aircraft and timing and runway length for the airports.
- Multiple stop flights:** Passengers can board and deboard.
- Request prioritization:** Request not served on given day will have priority on coming days.
- Monthly aircraft utilization:** Contracting constraints require leased aircraft to be utilized a certain amount of block hours in a month, minimum guaranteed hours (MGH).

3.3.3. UNHAS South Sudan Fleet and Operational Constraints

The fleet that is used in South Sudan is given in Table 3.4. Each aircraft has a specific (1) cruising speed, (2) cost, (3) capacity, (4) range, (5) runway required and (6) location where the aircraft is stationed.

Aircraft	Aircraft type	Cruising speed [nm/hr]	Cost [unit/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
Dash 8_2	DHC8-202	200	17	37	1020	2000	Juba
Dornier 228	Dornier 228	220	11	15	1000	1000	Juba
Cessna 208_1	C-208B	180	10	10	1070	1000	Juba
Cessna 208_2	C-208B	180	9.2	10	1070	1000	Juba
Cessna 208_3	C-208B	180	9.5	10	1070	1000	Juba
Cessna 208_4	C-208B	180	10.2	10	1070	1000	Juba
Cessna 208_5	C-208B	180	9.8	10	1070	1000	Juba
Mil Mi8_1	Mi8-T	120	32	17	355	50	Juba
Mil Mi8_2	Mi8-T	120	33	17	355	50	Juba
Cessna 208_1R	C-208B	180	11	10	1070	1000	Rumbek
Cessna 208_2R	C-208B	180	10.5	10	1070	1000	Rumbek
Mil Mi8_1R	Mi8-T	120	32	17	355	50	Rumbek
Mil Mi8_2R	Mi8-T	120	31	17	355	50	Rumbek

Table 3.4: Fleet used by UNHAS in South Sudan

The runway is virtual because it the runways differ in length and it is unnecessary to know to exact length in South Sudan. What is of vital importance is to know what aircraft can land on a specific airport. For that reason virtual runway lengths are introduced that define what aircraft types can land and take-off. A list of the virtual runway lengths is given below:

- 3000m - These airports can handle all aircraft types.
- 2000m - These airports can handle aircraft types up to the Dash 8, excluding the Fokker 50.
- 1000m - These airports can handle aircraft types up to the Dornier 228 and Cessna 208, excluding the Fokker 50 and Dash 8.
- 50m - These airports can only handle helicopters, specifically the Mil Mi8.

4

Methodology

The opportunity for humanitarian airlines is clearly stated and the next step is to transform this opportunity or problem in a method or model that can tackle this problem. This translation will be discussed in this Chapter. In order to optimize the humanitarian transportation, the problem will be defined as a Vehicle Routing Problem, specifically a Multi-Depot Heterogeneous Pickup and Delivery Problem with Time Windows. Together with the operational constraints that hold for the humanitarian setting the model will be called the Humanitarian Flight Optimization Model (referred to as HFOM). The problem will be formulated to be solved by an exact method (branch and bound) and a metaheuristic, the Genetic Algorithm (GA).

4.1. Vehicle Routing Problem

The route planning for a commercial airline starts with a fleet plan and has a selection of routes to be flown as output. The route planning process is ongoing and constantly evaluated. The 'vision' of the airline is vital in this part and ultimately decides which direction should be followed. Whether an airline will do long-haul or short-haul, domestic or international flights is decided in this process. However, as the route planning is an ongoing process, the economic assessment of routes can have an impact on which routes are flown.

Demand, operating cost and revenue forecasts are used to identify possible markets. Route profitability models use the forecasts as inputs and select routes that maximize airline profits. The accuracy of such model is highly dependent on the estimates of the future demand and revenue, as well as the assumptions that are used. Another drawback is that competitive effects are difficult to implement (Belobaba et al., 2009). The humanitarian setting has a big impact on how planning is executed. The route planning and schedule development are done simultaneously and updated on a daily basis. The aircraft routing switches to a vehicle routing problem.

The vehicle routing problem (VRP) is a generalization of the Travelling Salesman Problem (TSP) and dates back to 1959, when Dantzig and Ramser proposed an "optimum routing of gasoline delivery trucks between a bulk terminal and a large number of service stations supplied by the terminal" (Dantzig and Ramser, 1959). A linear programming formulation was used to come up with a near optimal solution. Since that innovative paper, the vehicle routing problem has been covered extensively in the literature. A quick search using "vehicle routing problem" on Scopus yields more than 6,000 results and on Google Scholar even more than 60,000. It is of no surprise that the vehicle routing problem is one of the most successfully tackled problems in the field of operations research.

A division can be made in VRP literature based on the type of problem and on the optimization algorithm used to solve the problem, e.g. exact, heuristics or metaheuristics. Exact algorithms will find an optimal solution in a finite amount of time, but the more difficult or large the optimization problem is, the more time that will be needed to find a solution. Heuristics will generally return a suboptimal solution, but will do this in a reasonable amount of time by executing a limited search. In metaheuristics the solution space is explored more thoroughly. This in turn gives a solution closer to the optimal, but increases the computing time compared to classical heuristics. The routing problem for the humanitarian needs is not a 'large' problem (approximately 30 origin-destination requests and 40 airports to serve), so the focus of this thesis will lie on exact algorithms. Figure 4.1 gives the most used formulations and the solution methods for the VRP. Note that this overview is not complete, but illustrates the most used methods.

As Toth and Vigo (2002) reports, no exact algorithm is capable of consistently solving VRP instances with more than 50 customers. Eksioglu et al. (2009) provides a taxonomy for VRP problems to classify the literature. However, for the scope of this thesis that taxonomy is too detailed. For each variant of VRP shown below a small section will be used to cover the heuristic and metaheuristic solution methods, but the focus will lie on the integer programming formulation and the exact solution method.

According to Toth and Vigo (2002) three basic modelling approaches exists, being the (1) *vehicle flow formulation*, the (2) *commodity flow formulation* and the (3) *set-partitioning formulation*. This thesis will focus on commodity flow formulations of the different VRP.

The optimization models that will be looked at are linear programs (LP). The next section will present several adaptations of the VRP and provide their LP formulation. A taxonomy of the VRP is given by Bodin (1975); De Jaegere et al. (2014); Eksioglu et al. (2009), a bibliography is presented by Laporte and Osman (1995) and more in-depth surveys can be found in Bodin (1975); Desrochers et al. (1990); Fisher (1995); Laporte (1992, 2009); Laporte et al. (2000); Raff (1983); Solomon and Desrosiers (1988). The book by Toth and Vigo (2002) covers most versions of the VRP, provides an insight in the solution methods (exact, heuristics and metaheuristics) and also presents real-world applications. Other books which deal with VRP are written by Cordeau and Laporte (2007); Golden et al. (2008); Pereira and Tavares (2009); Siarry (2016).

Heuristics and metaheuristics

Classical heuristics are used to quickly compute solutions to large instances or to initialize metaheuristics. The savings algorithm published by Clarke and Wright (1964) is the most widely known heuristic. Classical heuristics have mostly been developed between 1960 and 1990, whereafter the focus was switched to metaheuristics. Classical heuristics have been heavily researched (Altinkemer and Gavish, 1991; Christofides, 1976; Desrochers and Verhoog, 1989; Fisher and Jaikumar, 1981; Foster and Ryan, 1976; Gillett and Miller, 1974; Lin and Kernighan, 1973; Wark and Holt, 1994) and Laporte et al. (2000) provides a survey on classical and modern heuristics for the VRP. Metaheuristics are divided in categories based on solution strategy. Tabu Search (Barbarosoglu and Ozgur, 1999; Gendreau et al., 1994; Jia et al., 2013; Osman, 1991; Rego, 1998, 2001; Taillard, 1993; Toth and Vigo, 2003; Xu and Kelly, 1996), Genetic Algorithm (Baker and Ayechev, 2003; Berger and Barkaoui, 2003; Wang and Lu, 2009), Simulated Annealing (Osman, 1991), Ant Colony (Bullnheimer et al., 1999; Reimann et al., 2004) and Greedy Randomized Adaptive Search Procedure (GRASP) (Marinakis, 2012; Prins, 2009) are all being used in the VRP.

4.2. VRP Adaptations

The VRP is a generalization of the Traveling Salesman Problem (TSP), where the shortest route needs to be determined while passing through n points once. Generalizations of the TSP come in the form of conditions, e.g. specified deliveries q_i made a every point P_i (Dantzig and Ramser, 1959). The adaptation made by Dantzig and Ramser is commonly referred to as the Classical (or Capacitated) VRP, or just VRP. Since then, more variants to the VRP have been studied. The most prominent ones are shown in Figure 4.1. The most recent research in the field of VRP is on Green Vehicle Routing (GVRP), which incorporates environmental costs with economic costs by selecting routes that meet environmental requirements. A survey on the GVRP is available (Lin et al., 2014). All adaptations are NP-hard as they generalize the Traveling Salesman Problem (TSP) (Garey and Johnson, 1979).

1. Classical VRP (CVRP)
2. Heterogeneous VRP (HVRP)
3. VRP with Time Windows (VRPTW)
4. VRP with Pickup and Delivery (VRPPD)
5. Multi-Depot VRP (MDVRP)
6. Green VRP (GVRP)

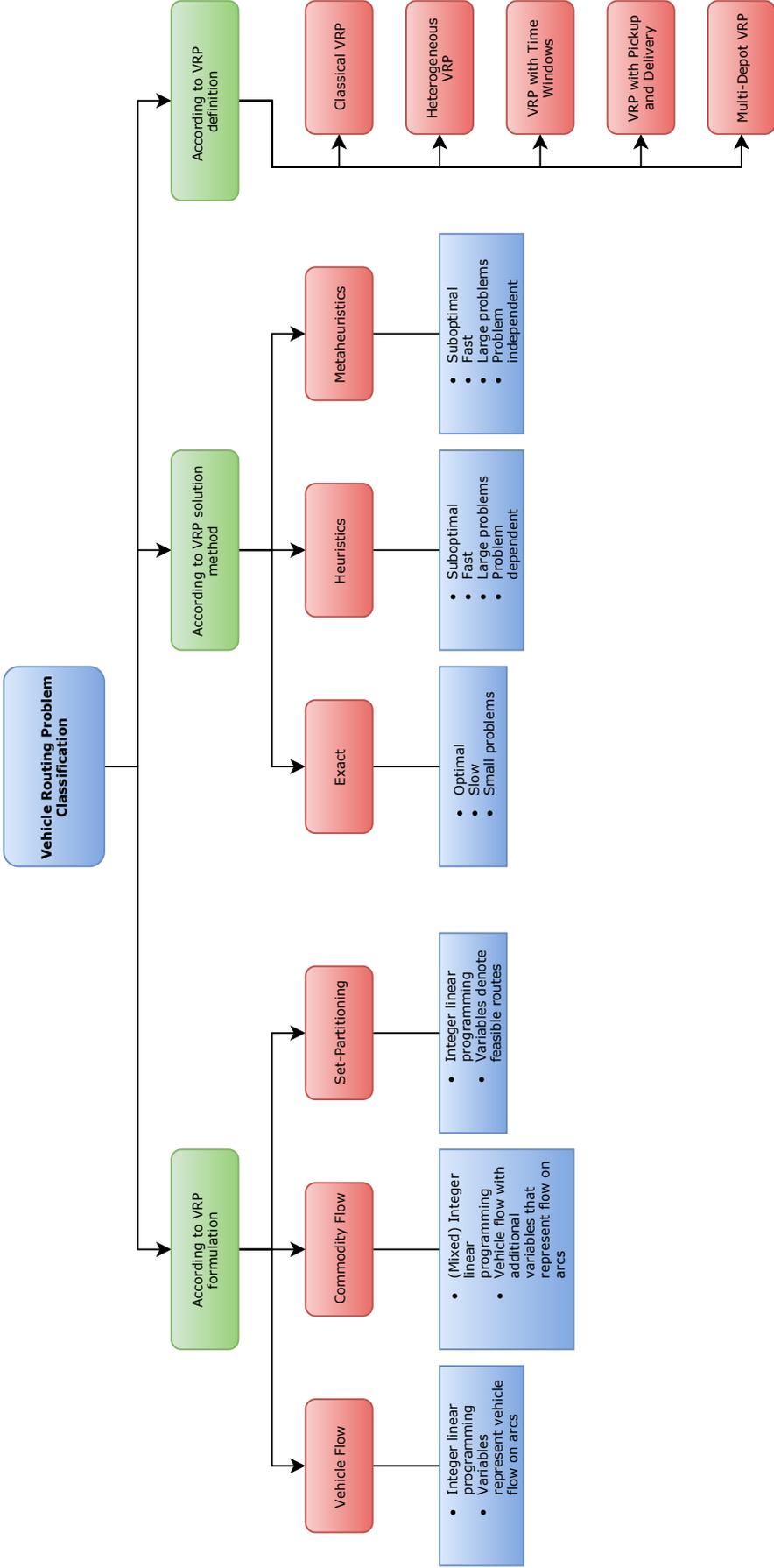


Figure 4.1: Vehicle routing problem classification

CVRP

The Classical Vehicle Routing Problem is the simplest form of the VRP and has also been studied the most (De Jaegere et al., 2014; Eksioğlu et al., 2009). An optimal route is to be found for a homogeneous fleet with a single depot. The objective is to find the least cost (least distance) routes to be traversed such that the demand for each customer is satisfied, each customer is visited once, each vehicle starts and ends at a depot and the capacity of the vehicles is not exceeded. Several exact algorithms can be found in Baldacci et al. (2004, 2012); Laporte and Nobert (1983); Letchford and Salazar-González (2015); Toth and Vigo (2002).

Definition

Based on Toth and Vigo (2002) and Parragh et al. (2008) the following definition for the VRP will be used throughout the thesis. Let $G = (V, A)$ be a complete graph where $V = \{0, \dots, n+1\}$ (or $V = \{0, n+1\} \cup N$) is the set of nodes and A is the arc set. Set $N = \{1, \dots, n\}$ (or $V \setminus \{0, n+1\}$) correspond to the airports and nodes 0 and $n+1$ correspond to the start and end hub respectively. A non-negative cost c_{ij} is associated with each arc $(i, j) \in A$ and represents the cost to go from node i to node j . Each airport i is associated with a known non-negative demand q_i and the vehicle capacity is equal to C . The arc set is reduced to $A = \{(i, j) : i, j \in V, i \neq j, i \neq n+1, j \neq 0\}$ for practicality. The complete nomenclature for the CVRP can be found in Table 4.1.

<i>Sets</i>		<i>Parameters</i>	
N	set of airports, with $N = \{1, \dots, n\}$ or $N = V \setminus \{0, n+1\}$	n	number of nodes
V	set of nodes, with $V = \{0, \dots, n+1\}$ or $V = \{0, n+1\} \cup N$	q_i	demand at node i , with $i \in V$
A	set of arcs, with $A = \{(i, j) : i, j \in V, i \neq j, i \neq n+1, j \neq 0\}$	c_{ij}	cost to traverse arc (i, j) , with $(i, j) \in A$
		C	vehicle capacity
<i>Variables</i>			
x_{ij}	$= \begin{cases} 1, & \text{if arc } (i,j) \text{ is traversed in the optimal solution} \\ 0, & \text{else} \end{cases}$		
u_i	load of vehicle after visiting airport i		

Table 4.1: Nomenclature CVRP

The following two-index MILP formulation for the CVRP is given based on formulations by Toth and Vigo (2002) and Parragh et al. (2008):

Two-index MILP formulation

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (4.1)$$

subject to:

$$\sum_{i:(i,j) \in A} x_{ij} = 1 \quad \forall j \in V \setminus \{0\} \quad (4.2)$$

$$\sum_{j:(i,j) \in A} x_{ij} = 1 \quad \forall i \in V \setminus \{n+1\} \quad (4.3)$$

$$x_{ij}(u_i + q_j - u_j) \leq 0 \quad \forall (i, j) \in A \quad (4.4)$$

$$q_i \leq u_i \leq C \quad \forall i \in N \quad (4.5)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (4.6)$$

The objective function (4.1) minimizes the total routing cost. Constraints (4.2) and (4.3) ensure that one arc enters and leaves each node associated with an airport (each airport is visited once). Capacity and connectivity requirements are imposed by (4.4) and (4.5). Subtours are eliminated by these constraints as well (Desrochers and Laporte, 1991; Miller et al., 1960). Equation (4.4) is non-linear, but can be linearised by using the binary conditions of (4.6) (Cordeau, 2006; Cordeau et al., 2002):

$$u_i + q_j - u_j \leq M_{ij}(1 - x_{ij}) \quad \forall (i, j) \in A \quad (4.7)$$

where M_{ij} are large constants, which can be replaced by $\min\{C^k, Q^k + q_i\}$.

4.3. Mathematical Formulation of the Linear Program

The HFOM is a Multi-Depot Heterogeneous Pickup and Delivery Problem with Time Windows further constrained by humanitarian needs. The goal of the model is to construct "optimal routes to satisfy transportation requests, each requiring pickup at an origin and delivery at a destination, under vehicle capacity, time windows and precedence constraints" (Jung and Haghani, 2000). The HFOM has added constraints that ensure that (1) vehicles can have an origin in different hubs (multi-depot problem), (2) the operational time of the vehicles is not exceeded, (3) the distance flown in a routing is bounded by the range of a vehicle, (4) the maximum runway length is adhered to and (5) the maximum legs in a routing is limited. The HFOM will be described below and the mathematical formulation will be given and thoroughly explained.

4.3.1. Problem Description

The characteristics of the HFOM can be described by the pickup and delivery nodes, the demand between those nodes, the distance between nodes and the vehicle capacity. A set of transportation requests has to be satisfied by a given fleet of vehicles. Each request has a pickup location (origin node), a delivery location (destination node) and a load that has to be transported. In addition, every node has a time windows in which a request has to be served. A vehicle can arrive at a node (pickup or delivery) before the time window, but it would have to wait to serve that request. Each vehicle has a certain capacity, a block time and an operational range. All vehicles have to return to their hub at the end of a route. The objective of the HFOM is to minimize the total cost obtained when serving all requests. An example of the problem can be found in Figure 4.2.

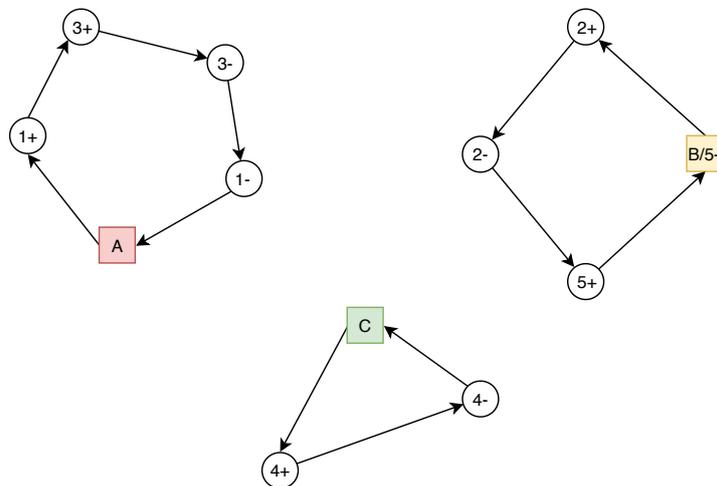


Figure 4.2: Example routing solution

In this example three hubs or depots exist, being A, B and C and at every hub two vehicles are stationed. Five requests exist, consisting of 10 nodes (5 pickup and 5 delivery nodes). A pickup node is illustrated as a + and a delivery node as a -, e.g. 1+ and 1- are the pickup and delivery node for request 1, respectively. An optimal routing schedule that can cover all the requests could be the routing schedule that uses only three vehicles. Vehicle 1, that has its origin in hub A, moves to the pickup location of request 1 and then to the pickup location of request 3. After that the delivery location of request 3 and 1 follow respectively. Vehicle 2 starts in hub B and moves to the pickup of request 2, then the delivery of request 2 and followed by the pickup and delivery of request 5. The delivery of request 5 is at the hub. Finally, vehicle 3 starts in hub C, and does the pickup and delivery of request 4, ultimately ending in hub C.

4.3.2. Mathematical Formulation

This section will give the complete formulation and nomenclature of the HFOM as an overview. The successive sections will further elaborate on the objective function, the constraints and the decision variables used in the mathematical formulation.

The linear programming formulation is based on the formulation by Parragh et al. (2008), adapted for humanitarian operations:

HFOM, Three-index MILP formulation

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k x_{ij}^k \quad (4.8)$$

subject to:

$$\sum_{k \in K} \sum_{i \in P} \sum_{j: (i,j) \in A} q_i \cdot x_{ij}^k \geq q_{tot} \cdot S \quad (4.9)$$

$$\sum_{k \in K} \sum_{j: (i,j) \in A} x_{ij}^k \leq 1 \quad \forall i \in P \quad (4.10)$$

$$\sum_{j: (h_{start}, j) \in A} x_{h_{start}, j}^k = 1 \quad \forall k \in K \quad (4.11)$$

$$\sum_{i: (i, h_{end}) \in A} x_{i, h_{end}}^k = 1 \quad \forall k \in K \quad (4.12)$$

$$\sum_{j: (j, i) \in A} x_{ji}^k - \sum_{j: (i, j) \in A} x_{ij}^k = 0 \quad \forall i \in P \cup D, k \in K \quad (4.13)$$

$$\sum_{j: (i, j) \in A} x_{ij}^k - \sum_{j: (n+i, j) \in A} x_{n+i, j}^k = 0 \quad \forall i \in P, k \in K \quad (4.14)$$

$$\text{if } x_{ij}^k = 1 \quad w_i^k + s_i^k + t_{ij}^k \leq w_j^k \quad \forall (i, j) \in A, k \in K \quad (4.15)$$

$$a_i^k \leq w_i^k \leq b_i^k \quad \forall i \in V, k \in K \quad (4.16)$$

$$w_i^k + t_{i, n+i}^k \leq w_{n+i}^k \quad \forall i \in P, k \in K \quad (4.17)$$

$$\text{if } x_{ij}^k = 1 \quad u_i^k + q_j \leq u_j^k \quad \forall (i, j) \in A, k \in K \quad (4.18)$$

$$u_i^k \leq Q^k \quad \forall i \in V, k \in K \quad (4.19)$$

$$\text{if } x_{ij}^k = 1 \quad v_i^k + d_{ij} \leq v_j^k \quad \forall (i, j) \in A, k \in K \quad (4.20)$$

$$v_i^k \leq r^k \quad \forall i \in V, k \in K \quad (4.21)$$

$$r w_r^k \cdot x_{ij}^k \leq r w_j \quad \forall (i, j) \in A, k \in K \quad (4.22)$$

$$\sum_{j: (i, j) \in A} e_{ij} \cdot x_{ij}^k \leq 3, 4 \text{ or } 5 \quad \forall k \in K \quad (4.23)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall (i, j) \in A, k \in K \quad (4.24)$$

The objective function (4.8) minimizes the total routing cost and is paired with the demand satisfaction constraint given by equation 4.9. Constraint (4.10) guarantees that every node is visited once. Constraints (4.11) and (4.12) ensure that each vehicle starts and ends at the hub. Note that not all vehicles have to be utilized. Constraint (4.13) guarantees flow conservation and constraint (4.14) makes sure that a request is served by the same vehicle. Constraints (4.15), (4.16) and (4.17) are to ensure the time windows are adhered to. The capacity of the aircraft should not be exceeded and to control this, constraints (4.18) and (4.19) are added. The range constraint is contained in equations (4.20) and (4.21) and the runway constraint can be found in equation (4.22). The final constraint (4.23) is the connection constraint.

The decision variables, objective function and the constraints will be touched upon in detail in the next sections.

Sets		Parameters	
P	set of pickup nodes, with $P = \{1, \dots, n\}$	n	number of pickup nodes
D	set of delivery nodes, with $D = \{n+1, \dots, 2n\}$	\tilde{n}	number of delivery nodes, paired pickups $\tilde{n} = n$
H	set of hub nodes, with $H = \{2n+1, \dots, 2n+2h\}$	h	number of hub nodes
V	set of nodes, with $V = \{1, \dots, 2n+2h\}$ or $V = P \cup D \cup H$	a_i^k	earliest time to begin service at node i for vehicle k
A	set of arcs, with $A = \{(i, j) : i, j \in V, i \neq j, i \neq 2n + h_{end}, j \neq 2n + h_{start}\}$	b_i^k	latest time to begin service at node i for vehicle k
K	set of vehicles, with $K = \{1, \dots, k\}$	k	number of vehicles
		q_i	demand at node i , with $i \in V$
		q_{tot}	total demand
		Q^k	capacity for vehicle k
		c_{ij}^k	cost to traverse arc (i, j) for vehicle k , with $(i, j) \in A, k \in K$
		d_{ij}	distance between node i and j
		e_{ij}	binary parameter, 1 if $d_{ij} > 0$, 0 if $d_{ij} = 0$
Variables			
x_{ij}^k	$= \begin{cases} 1, & \text{if arc } (i, j) \text{ is traversed for vehicle } k \\ 0, & \text{else} \end{cases}$	s_i^k	TAT at node i for vehicle k
u_i^k	load of vehicle k after visiting node i	S	percentage of demand satisfied
w_i^k	time travelled at node i for vehicle k	t_{ij}^k	travel time from i to j for vehicle k
v_i^k	distance travelled at node i for vehicle k	$r w_i$	runway at node i
		$r w r^k$	runway required by vehicle k

Table 4.2: Nomenclature HFOM

4.3.3. Decision Variables

In order to find a solution to a linear program, decisive factors have to be changed and a combination of variables has to be found to result in an optimal solution to the problem. The variables chosen to be decisive in this process are called decision variables. Each linear program has different decision variables, which together with the objective function and the constraints make it unique. The HFOM uses four different decision variables, one binary and three continuous:

- **Binary**

1. x_{ij}^k is the main binary decision variable, which equals 1 if the leg with origin i and destination j is traversed by vehicle k

- **Continuous**

1. u_i^k is a continuous decision variable, representing the load of vehicle k after visiting node i
2. v_i^k is a continuous decision variable, representing the distance travelled at node i for vehicle k
3. w_i^k is a continuous decision variable, representing the time travelled at node i for vehicle k

4.3.4. Objective Function

This section will cover two methods that are commonly used in multi-objective optimization, being the weighted-sum and ϵ -constraint method.

The weighted-sum method revolves around scalarizing a set of objectives into a single objective by giving user-defined weights (w_i) to each objective (F_i) and summing them (Chircop and Zammit-Mangion, 2013). The weight of an objective defines the relative importance of that objective. The main problem of this method is finding the appropriate weighting for the coefficients. The decision maker has to decide a-priori. The formulation (based on the one given by Chircop and Zammit-Mangion (2013)) can be found in equation (4.25).

$$\begin{aligned} \min \sum_{i=1}^m w_i F_i(x) & \quad (4.25) \\ \text{subject to} \quad & g(x) \leq 0 \\ & h(x) = 0 \end{aligned}$$

In the ϵ -constraint method one of the objectives is selected to be optimized. The remaining objectives are added as additional constraints. This method was first introduced by Haimes et al. (1971). The mathematical formulation for the ϵ -constraint method can be expressed by equation (4.26), again adopted from Chircop and Zammit-Mangion (2013).

$$\begin{aligned} \min F_{i_p}(x) & \quad (4.26) \\ \text{subject to} \quad & F_i(x) \leq \epsilon_i \quad \text{for } i = 1, \dots, m \text{ with } i \neq i_p \\ & g(x) \leq 0 \\ & h(x) = 0 \end{aligned}$$

In this particular case, the cost minimization is chosen as the main objective, while the demand satisfaction is fixed as a constraint. Since the nature of this optimization problem has only two objectives and at least 90% demand satisfaction on a daily basis needs to be achieved, the choice was made to go for the ϵ -constraint method over the weighted-sum method. The ϵ -constraint method is easy to implement and widely applicable. Pareto optimal solutions can be identified with the ϵ -constraint method that are not obtainable using the weighted-sum technique (Knowles, 2010). Even though hard constraints are introduced with this method, it is still preferable over the weighted-sum technique.

The objective function for the simple HFOM can now be defined with the following set of equations (4.27) and (4.28).

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij}^k x_{ij}^k \quad (4.27)$$

$$\text{subject to} \quad \sum_{k \in K} \sum_{i \in P} \sum_{j: (i,j) \in A} q_i \cdot x_{ij}^k \geq q_{tot} \cdot S \quad (4.28)$$

In which:

- x_{ij}^k is the binary decision variable, which equals 1 if the leg with origin i and destination j is traversed by vehicle k
- c_{ij}^k is the cost to traverse leg i to j with vehicle k
- q_i is the amount of passengers to be transported at node i (positive for origin, negative for destination), q_{tot} is the total amount of passengers to be transported and S is the percentage of passengers to be served
- A is the set of arcs of flight legs that exist
- P is the set of origin nodes
- K is the set of vehicles

4.3.5. Constraints

The mixed integer linear program is subject to several linear constraints, which define the feasible region. The constraints for this model are hard constraints, meaning that they cannot be violated and have to be satisfied in any case. Violating a hard constraint will result in an infeasible solution. The constraints for the HFOM are for a typical humanitarian flight operation, where time windows, load, range of the aircraft, an runway required by the aircraft are examples of constraining factors. This section covers all the constraints of the HFOM and discusses them individually

Customers served constraint

The first constraint is the customers served constraint, given by the inequality constraint equation (4.29). It states that every request has to be served a maximum of one time. Together with the demand satisfaction

constraint, expressed by the inequality equation (4.28), it is defined how many of the requests have to be served.

$$\sum_{k \in K} \sum_{j: (i,j) \in A} x_{ij}^k \leq 1 \quad \forall i \in P \quad (4.29)$$

Start and end at hub constraint

In order to define where an aircraft has to start and where it has to end its tour, the equality constraints (4.30) and (4.31) are needed. An aircraft has to start in hub h_{start} and has to end in hub h_{end} .

$$\sum_{j: (h_{start}, j) \in A} x_{h_{start}, j}^k = 1 \quad \forall k \in K \quad (4.30)$$

$$\sum_{i: (i, h_{end}) \in A} x_{i, h_{end}}^k = 1 \quad \forall k \in K \quad (4.31)$$

Flow conservation constraint

The flow conservation constraint ensures that if an aircraft arrives at a node j , it also leaves that node j . In this way the flow of the aircraft is conserved. This constraint is valid for all nodes i that belong to the set of the origin (P) and destination (D) nodes. This flow conservation is covered by equality constraint (4.32).

$$\sum_{j: (j,i) \in A} x_{ji}^k - \sum_{j: (i,j) \in A} x_{ij}^k = 0 \quad \forall i \in P \cup D, k \in K \quad (4.32)$$

Same vehicle served constraint

The following constraint guarantees that a request is served by the same vehicle. Since a request is linked with a pickup and a delivery point, it has to be assured that the origin and destination are paired and accessed by the same aircraft. Equality constraint (4.33) incorporates this into the linear program.

$$\sum_{j: (i,j) \in A} x_{ij}^k - \sum_{j: (n+i,j) \in A} x_{n+i,j}^k = 0 \quad \forall i \in P, k \in K \quad (4.33)$$

Timing constraints

The timing of each aircraft is covered by inequality constraints (4.34), (4.35) and (4.36). The time for each aircraft k at node i is depicted by w_i^k . If leg x_{ij}^k is flown, the time w_i^k added with the turnaround time s_i^k and the flight time t_{ij}^k has to be lower than time w_j^k . The arrival and departure times are bounded by a_i^k and b_i^k respectively. In order to ensure that the destination of a request is visited after the origin, precedence constraint (4.36) is added. The timing constraints also ensure that subtours are broken, given that $t_{ij}^k + s_i^k > 0$ (Desrochers and Laporte, 1991; Miller et al., 1960; Parragh et al., 2008).

$$\text{if } x_{ij}^k = 1 \quad w_i^k + s_i^k + t_{ij}^k \leq w_j^k \quad \forall (i,j) \in A, k \in K \quad (4.34)$$

$$a_i^k \leq w_i^k \leq b_i^k \quad \forall i \in V, k \in K \quad (4.35)$$

$$w_i^k + t_{i,n+i}^k \leq w_{n+i}^k \quad \forall i \in P, k \in K \quad (4.36)$$

Load constraints

The load constraints given by equations (4.37) and (4.38) follow the same principle as the timing constraints. The upper bound of load u_i^k is now given by the capacity of the vehicle Q^k .

$$\text{if } x_{ij}^k = 1 \quad u_i^k + q_j \leq u_j^k \quad \forall (i,j) \in A, k \in K \quad (4.37)$$

$$u_i^k \leq Q^k \quad \forall i \in V, k \in K \quad (4.38)$$

Range constraints

Constraints (4.39) and (4.40) restrict a vehicle in the amount of distance it can traverse in a tour. The distance travelled by a vehicle v_i^k is bounded by the range of that vehicle r^k .

$$\text{if } x_{ij}^k = 1 \quad v_i^k + d_{ij} \leq v_j^k \quad \forall (i,j) \in A, k \in K \quad (4.39)$$

$$v_i^k \leq r^k \quad \forall i \in V, k \in K \quad (4.40)$$

Runway constraint

The next constraint of the HFOM is the runway constraint. The representation of this constraint is given by equation (4.41). The runway required by a vehicle rw_r^k has to be lower than or equal to the runway travelled to rw_j .

$$rw_r^k \cdot x_{ij}^k \leq rw_j \quad \forall (i, j) \in A, k \in K \quad (4.41)$$

Connection constraint

The final constraint of the HFOM is the connection constraint, which limits the amount of legs in a routing. This constraint representation is given by equation (4.42). The parameter e is binary and takes a 1 if the connection i to j has a positive value for the distance between i and j . If the distance is zero, the value of e is also zero.

$$\sum_{j:(i,j) \in A} e_{ij} \cdot x_{ij}^k \leq 3, 4 \text{ or } 5 \quad \forall k \in K \quad (4.42)$$

4.3.6. Genetic Algorithm

A genetic algorithm (GA) has been developed to solve the multi-vehicle, multi-depot pickup and delivery problem. This was done to test if the GA could be used for solving the problem for instances where the exact solution procedure is too time-consuming. The genetic algorithm is a metaheuristic that is able to solve the pickup and delivery problem in an short time compared to the exact algorithm (Jung and Haghani, 2000). GAs are based on the mechanism of natural selection and natural genetics to search stochastically for a solution to a problem (Gen and Cheng, 1997). GAs start with a random feasible solution and they can handle any sort of objective function and constraints. The general procedure of the GA is described below using the formulation by Jung and Haghani (2000):

1. Generate an initial population of N number of random solutions
2. Evaluate fitness value for each individual in the population
3. While convergence criterion is not met or generation limit is reached:
 - Select parents
 - Recombine parents to create children
 - Evaluate fitness value of children
 - Select new population from parents and children

Chromosome representation

The chromosomes in traditional GA problems are represented by binary strings. However for the VRP this representation is not well suited. Jung and Haghani (2000) proposes a random key representation. This method will be presented below.

The random key representation for the VRP with multiple vehicles uses a four-digit number gene representation where the first digit is the vehicle number and the last three digits are used to generate the timings. A chromosome is built up with genes and each gene is a pickup or a delivery node (Jung and Haghani, 2000). This random key chromosome representation can be found in Table 4.3. This shows a solution to a 3 request routing problem. The multi-vehicle VRP has two main constraints related to sequence, being the coupling and the precedence constraints. The coupling constraint states that a request consisting of a pickup and a delivery should be handled by the same vehicle and the precedence constraint ensures that a pickup is always done before the delivery.

Index	1	2	3	4	5	6
Request	1+	1-	2+	2-	3+	3-
Value	1345	1895	1452	1632	2153	2359

Table 4.3: Random Key chromosome representation

A pickup point in this chromosome representation is depicted as a + and a delivery as a -. For example request 1+ is the origin node for request 1 and 1- is the delivery node for request 1. The value of the pickup should always be lower than the delivery. The first digit in this value represents the vehicle number and the next three are involved in the timings. After decoding, the route of this particular example would be as given in Table 4.4.

Vehicle 1	1+	2+	2-	1-
Vehicle 2	3+	3-		

Table 4.4: Decoded route

After the generation of the routes in this manner, the routes are checked for capacity or time violations. If those have not been violated and the route is unique, the fitness value is calculated for each chromosome in the population.

Selection and replacement

After the initial population N has been created and the fitness has been evaluated, the selection and replacement procedure can be started. Firstly, the offspring is created by genetic operators, which will be analysed in the following section, and the new generation is selected from a enlarged sampling space which contains the parents and the offspring. The selection is done according to the fitness value of the chromosomes and by elitist selection. The best parents are chosen as elites and will be automatically moved to the new generation, unaltered. The selection for the next ($N - elites$) chromosomes is done based on a selection probability. The selection probability p_i is determined using equation (4.43).

$$p_i = \frac{\text{eval}(i)}{\sum_{i=1}^N \text{eval}(i)} \quad (4.43)$$

Genetic operators

As briefly mentioned in the previous section, the offspring is created by two genetic operators, crossover and mutation, as given by Jung and Haghani (2000).

Crossover

The crossover operator works with two steps on a two-point basis:

1. Create two crossover points
2. Swap segments of the parents and create children

In order to make sure that the coupling of the pickup and delivery is not broken, a two-point crossover is needed and each crossover point has to be an odd number. An example of this is given in Table 4.5. In this example the crossover points are at numbers 3 and 5. This coincides with the point before the pickup of request 2 and after the delivery of request 2, respectively. Once these points have been chosen, the children can be created from the parents, ma and pa , by performing the crossover. Two children are generated, in this case called child 1 and child 2. After this, the crossover is completed and the children are checked for feasibility. If a child contains an infeasible route (either by range violation or capacity violation), the crossover is disregarded and a new crossover is performed. The routes that are represented by the parents and the children are given Table 4.6.

Request	1+	1-	2+	2-	3+	3-
Ma	1345	1895	1452	1632	2153	2359
Pa	2133	2695	2304	2949	1004	1854
Child 1	1345	1895	2304	2949	2153	2359
Child 2	2133	2695	1452	1632	1004	1854

Table 4.5: Two-point crossover at 3 and 5

Mutation

The mutation operator is again based on two steps:

1. Generate uniform random number between 0 and 1 for each request in each chromosome
2. If the random number of request i is lower than the mutation rate, the first digit (vehicle number) is changed for the pickup and delivery of that request.

Ma	Vehicle 1	1+	2+	2-	1-
	Vehicle 2	3+	3-		
Pa	Vehicle 1	3+	3-	0	0
	Vehicle 2	1+	2+	1-	2-
Child 1	Vehicle 1	1+	1-		
	Vehicle 2	3+	2+	3-	2-
Child 2	Vehicle 1	3+	2+	2-	3-
	Vehicle 2	1+	1-		

Table 4.6: Routes of parents and children

Again an example is given for this genetic operator, which can be found in Table 4.7. In this case the random number generated for request 1 is apparently lower than the mutation rate. This first digit of this request is now changed to a different vehicle number.

Request	1+	1-	2+	2-	3+	3-
Parent	1345	1895	1452	1632	2153	2359
Child	2345	2895	1452	1632	2153	2359

Table 4.7: Mutation for request 1

The Genetic Algorithm in this form was not able to optimally solve a 6 request, 5 vehicle, 3 hub problem, while the exact method could solve this problem within two seconds. This method for solving is therefore discarded, but the methodology is shown to illustrate that heuristics and metaheuristics can be utilised to generate solutions to this problem.

5

Model

The physical problem has been translated to a mathematical model called the Humanitarian Flight Optimization Model (HFOM) in Chapter 4. The next step is to generate a solution to this model for a case study on the humanitarian routing and scheduling for UNHAS in South Sudan. This will be done by creating a software application that takes the daily flight requests and delivers the routing and scheduling for that day. This Chapter covers the analysis of the process involved in getting from an input to an output. The flow of the process will be divided in three parts, namely the pre-processing, the optimization and the post-processing. Each part will be explained thoroughly in this Chapter.

5.1. Process Flow

The flow of the software application is depicted in Figure 5.1 Since the Humanitarian Flight Optimization Model is a sub-problem of the Vehicle Routing Problem, determining the optimal solution is NP-hard. The size of the problems than can be solved optimally using mathematical programming may be limited. In practice this means that problems with many variables and constraints take an unacceptable amount of time to solve optimally. From testing it was found that the HFOM was unable to find a solution for problems of over 15 requests on a ‘regular’ computer (8 GB RAM) using Cplex and Gurobi. Since the HFOM generally takes problems that have 20+ requests, this would make this method of solving impossible. To solve this issue a heuristic approach is adopted to decrease the size of the problem and make it solvable in an acceptable time frame. The pre-processing block covers this heuristic and shall be further elaborated upon in Section 5.1.1. The optimization for the sub-problems is done with the mathematical formulation designed in Chapter 4 and is discussed in Section 5.1.2. Finally, after the routes have been optimized, the schedule is made and the routes are plotted for easy interpretation. This post-processing block will be touched upon in Section 5.1.3.

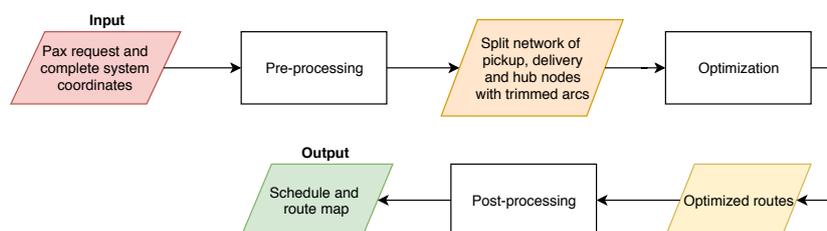


Figure 5.1: Process Flow diagram

5.1.1. Pre-processing

The pre-processing is the first step of the model and takes the daily flight requests and the coordinates of the airports as an input. The daily requests are collected by UNHAS one to two days in advance, following the breakdown tree provided in Section 3.3. The bookings are created in the TakeFlite system and the responsible Flight Tasking Officer uses that data to create the routing and scheduling for that day. The bookings from TakeFlite can be exported in the form of a table. An example of a daily request table can be found in Appendix

B, Figure B.1. The table in its current form is however unusable as an input for the linear program. Therefore, it has to be transformed into a functional format. The process of loading, dividing, transforming and filtering this data into a usable format for the linear program will be discussed below, starting with the loading of the relevant data. The division of the flight requests is followed and after that arcs (or edges) that can not be accessed by definition are removed from the complete set of arcs. And finally, decision variables that are always unused for a particular reason are deleted from the potential set of solutions.

1. Loading data
2. Division heuristic, dividing the requests
3. Creating nodes
4. Removing impossible edges
5. Removing unused decision variables
6. Creating parameters

1. Loading data

The daily flight requests are loaded from the database as well as the full airport list with names and coordinates (latitude and longitude) of the airports, which UNHAS is allowed to access. This airport list also includes the virtual runway length of the airports. A local dataset is made from both these datasets, so that the set of requests is expanded with the airport data (location and runway length). The fleet data is loaded from a different database and stored as a different local dataset. The fleet data consists of the aircraft's (1) cruising speed, the (2) cost to fly per nautical mile, the (3) capacity, the (4) range, the (5) runway required and the (6) hub where the aircraft is located. The reader is referred to Table 3.4 for this information. Moreover, for each day the block times of the aircraft are set at 10 hours. Since the problem is split up in five sub-problems (discussed in the following section), these block times have to be tracked and updated after each sub-problem is optimized in order to ensure that the maximum time is not exceeded. This updated block time is saved in a file and re-loaded every time a sub-problem optimization is being performed.

2. Division heuristic

The division heuristic splits the daily requests in manageable and solvable problem sizes of less than 15 requests since Cplex and Gurobi are not able to get results within a reasonable time if the problem size gets over 15 requests. Since the time it takes to find an optimal solution has to be within one hour to compete with the expert planners, it was decided to come up with a heuristic that divides the problem in sub-problems that can be solved sequentially. The requests are divided in six primary regions that have been identified by close examination of the daily routing, trial-and-error testing and by the use of expert opinions. Most passengers have their origin in Juba or their destination in Juba, making it the main hub. The 6 regions that have been distinguished are shown in Figure 5.2.

1. **'F50' region** This region deals with airports that can accommodate a Fokker 50, which is the largest aircraft in the fleet in terms of capacity. Juba, Rumbek, Wau and Aweil have a landing strip that is long enough to for the Fokker 50 to land on. Not only does this region handle the requests for Juba, Rumbek, Wau and Aweil, but it also connects passengers that have to travel from Juba to the 'Rumbek'-region and vice versa.
2. **'Rumbek 1' region** The 'Rumbek 1' region is a region that connects passengers that have their origin or destination in Juba to or from a helipad close to Rumbek. The hub Rumbek has two Cessna 208's and two Mil Mi8's to transport passengers to their destinations. Most of the airports in the 'Rumbek' area are actually helipads, only accessible by the Mil Mi8's.
3. **'Bor' region** The area surrounding the airport of Bor will be labelled as the 'Bor' domain. Requests are often coming in from Juba to Bor and back. The connection from Bor to Pibor is also a highly requested transfer, making this relatively low dense area an active one in terms of flights offered.
4. **'Juba North' region** The 'Juba North' area are all the airports that do not fall in any of the mentioned regions. Since all of the requests have either their origin or their destination in Juba and all airports lie north of Juba, this was appropriately deemed as the 'Juba North' region.

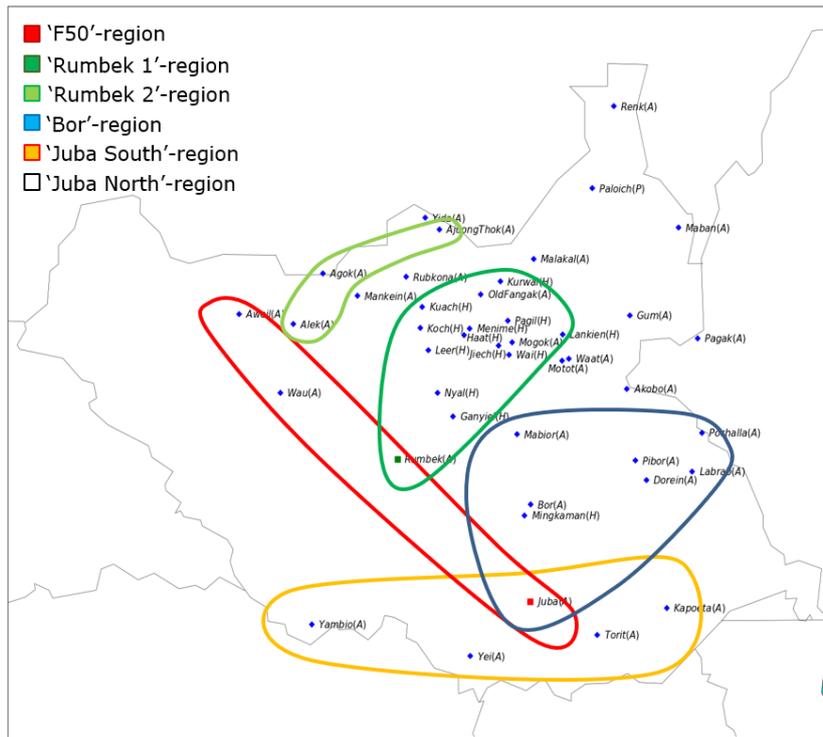


Figure 5.2: Division of South Sudan regions

5. **'Juba South' region** The area south of Juba is suitably referred to as the 'Juba South' region. Both Yambio and Yei can handle aircraft up to a Dash 8 (requiring a virtual runway of 2000m). Both the Dornier 228 and the Cessna 208 can land at Torit and Kapoeta.
6. **'Rumbek 2' region** The 'Rumbek 2' region is a region that connects passengers that have their origin or destination in Juba to or from a small landing strip close to Rumbek. The hub Rumbek has two Cessna 208's and two Mil Mi8's to transport passengers to their destinations. Only Agok, Ajuong Thok, Alek, Mogok and Old Fangak have a landing strip that is suited for a Cessna 208 to land on.

Now that the dataset of the requests is complete and filled with all the necessary information, the requests are split up in the different regions. This decision is made by cross-referencing the names of the origin and destination airports to airport names that are part of one of the six areas. However, for the 'F50' and 'Rumbek' region a different approach is needed, since connections take place of flights going to and from Juba. If a request exists between Juba and an airport that is located in the 'Rumbek' area, a virtual origin and destination of Rumbek is created for both the 'F50' and 'Rumbek' areas. For example, if the request Juba-Nyal is present a request of Juba-Rumbek is added to 'F50' and a request of Rumbek-Nyal is added to 'Rumbek'. The quantity of both requests is equal to that of the original request. While the requests are being divided in the different regions, the requests are also checked whether they do not exceed the capacity of the largest aircraft, in terms of capacity, that can land at origin or destination. If the request does exceed this quantity, the difference is seen as 'spilled' passengers and that request is saved to be used on a later date where more passengers are willing to fly on that particular origin-destination. The spillage decision can be found in Table 5.1 An

Airport runway length [m]	Aircraft possible to land	Spilled when
3000	up to Fokker 50	request $q_i > 50$
2000	up to Dash 8	request $q_i > 37$
1000	up to Dornier 228	request $q_i > 15$
50	Mil Mi8	request $q_i > 17$

Table 5.1: Decision on spillage

airport that has a runway length of 3000m is able to accommodate aircraft up to the Fokker 50. A request that exceeds the capacity of the Fokker 50 (50 seats) will be unable to be handled and are automatically seen as spilled passengers. A similar decision goes for airports that have a runway of 2000m. If the request exceeds the capacity of the Dash 8 (37 seats), the difference is seen as spilled. For the airports that have a runway of 1000m, the capacity of the Dornier 228 (15 seats) is leading. However, when a request is willing to fly to an airport that has a virtual runway of 1000m and the request consists of 20 or more passengers, the request is split in two requests of 10 to allow for two Cessna 208's to accommodate these. Airports with a helipad can only be accessed by the Mil Mi8 and are therefore limited by the capacity of that helicopter (17 seats). For the 'Rumbek 2' area, only two Cessna 208's are available, so the spillage for those requests happens when $q_i > 10$.

For the 'Bor', 'Juba North' and 'Juba South' region another decision is made. Since both the Cessna 208 and Dornier 228 are active aircraft in these regions, it is decided to split the requests that are between 10 and 15 into two requests of 10 and one of the difference of the request and the capacity of the Cessna 208 (10 seats). This is done in order to let the optimization program decide whether it is less costly to use two Cessna 208's or one Dornier 228 (the Dornier 228 might not be available due to timing constraints).

Requests to Rubkona, Malakal and Maban have another heuristic involved. A Malakal request is split in two requests if the request exceeds 15 to allow for triangle flights with Rubkona or Maban. A capacity check is also done for a routing with Rubkona. If the combined requests for Malakal-Rubkona would exceed 37 (capacity of Dash 8) and stay below 40, the difference would be seen as spillage for Malakal.

Once the requests have been divided and the 'spilled' requests have been filtered out, each sub-problem is ready to be processed further. The next step of the pre-processing involves the creation of the nodes in the network, discussed in the following segment.

3. Node creation

The daily requests that are processed consist of an origin, a destination and an amount of passengers to be transported between the given origin-destination. The origins are labelled as pickup nodes (P) and the destinations are tagged as delivery nodes (D). The hubs or depots are divided in starting and ending hub nodes. A numerical example is given in Figures 5.3 and 5.4 and Tables 5.2 and 5.3. In this test instance three hubs exists, being locations 6, 4 and 11. Each request is split up a pickup and a delivery node with a certain request quantity. A positive request is created for the pickup node and an equal but negative request is created for the delivery node. Finally, the hub nodes are added starting with the hubs where a tour is started and ending with a hub where the tour is ended. This order of pickups-destinations-starting hubs-ending hubs is chosen for convenience in programming. Once the nodes have been created, a distance matrix is computed using the coordinates of all the nodes. This matrix is stored in such a way that distances between nodes can be easily accessed when required. After the node creation has been completed, the network can be trimmed by removing edges or arcs that are never going to be traversed.

4. Edge removal

The next part of the pre-processing is the removal of edges (or arcs) that are impossible to be traversed by an aircraft. This is done in order to decrease the network as a whole and by that gaining a decrease in the time to find a solution to the optimization problem. The edges that are removed from the total set of arcs A (refer to Table 4.2) are summarized below:

- Edges where origin (i) is ending hub
- Edges where destination(j) is starting hub
- Edges where i equals j
- Edges where origin (i) is the starting hub and destination (j) is a delivery node
- Edges where origin (i) is a pickup node and destination (j) is the ending hub
- Edges of the type $(n + i, i) \quad \forall i \in P$, edges where destination is before origin
- Edges of the type $(2n + h_{start}, n + i) \quad \forall i \in P$ and $(i, 2n + h_{end}) \quad \forall i \in P$, edges where destination is right after starting hub and edges where ending hub is straight after origin

This removal ultimately leads to a fully trimmed set of arcs A that will be used to in the model and to create the decision variables x_{ij}^k .

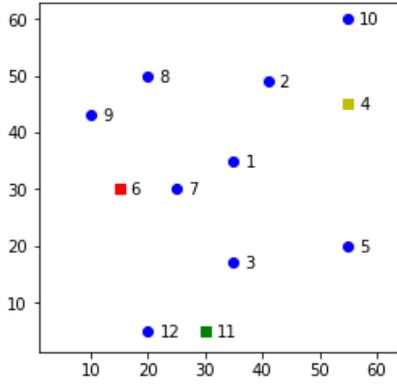


Figure 5.3: Test instance plot

Request number	Origin	Destination	Request quantity
1	9	8	3
2	8	7	10
3	7	1	20
4	3	6	5
5	1	3	2
6	4	10	9

Table 5.2: Test instance data

	Node	Location	Request quantity
Origin (P)	1	9	3
	2	8	10
	3	7	20
	4	3	5
	5	1	2
	6	4	9
Destination (D)	7	8	-3
	8	7	-10
	9	1	-20
	10	6	-5
	11	3	-2
	12	10	-9
Hubs (start)	13	6	0
	14	4	0
	15	11	0
Hubs (end)	16	6	0
	17	4	0
	18	11	0

Table 5.3: Test instance data, processed

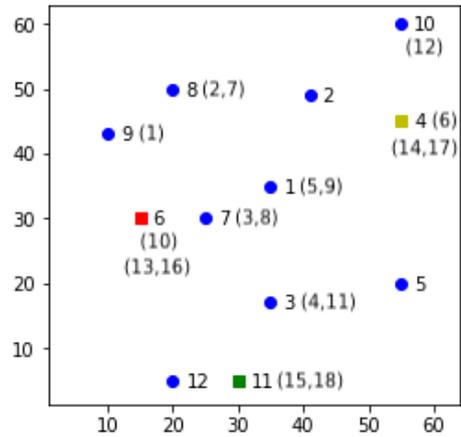


Figure 5.4: Test instance plot, processed

5. Decision variable removal

Now that the network is trimmed of any impossible and unused edges, the following step is to remove decision variables that are unused as well. The binary decision variables x_{ij}^k are created by copying the arcs from the trimmed set A and creating the variables for the different aircraft available. For the ‘F50’ region the decision variables are further trimmed. In order to accommodate the passengers that connect from the ‘Rumbek’ region to Juba, the Fokker 50 is solely used. To force this, the decision variables for the other aircraft have the edges removed that allow for this connection. Furthermore, passengers that connect from Juba to the ‘Rumbek’ region have to be transported there in the morning to allow for the Mil Mi8’s and the Cessna 208’s that are located in Rumbek to use 8 hours of operational time. This is done by setting the upper bound of the time window b_i^k (refer to Table 4.2) to 1.5 for those requests for all aircraft k , meaning that one of the aircraft in the ‘Juba’ region have to serve those requests within 1.5 hours.

6. Parameter creation

Finally, in order to complete the input for the optimizer the parameters and other decision variables given in Table 4.2 are created and stored. The parameters are created by using the data that was loaded earlier in step 1. With this the pre-processing is finished and optimization can be started.

5.1.2. Optimization

Once the data is pre-processed into a working format for the linear program, the optimization of the sub-problems can be executed. Every region has its own set of requests to be handled and a specific set of vehicles to cover these requests. A loop is initiated that handles each region and loads the required data for the region. The utilization of each vehicle is also updated after each iteration in order to ensure that the maximum time is not exceeded. For each sub-problem the objective function, the constraints and decision variables are created and used as input values for the optimizer. A linear program (lp) file is created and this file is sent to the optimization engine. The optimizer uses a dynamic search algorithm to find a mathematically feasible solution and returns this solution in a format that has to be decoded and filtered in an understandable and easily readable manner.

5.1.3. Post-processing

The post-processing is initiated once a solution to the routing problem has been generated. The optimization block creates several files after each sub-problem has been optimized. These files include the LP file, the solution file in XML format and a log file to track the progress. The time an aircraft has been used is also tracked and a file is updated that contains these aircraft utilisations. The routings of each sub-problem are generated as well and saved in a separate file. The post-processing uses these output files to generate a plot of the routing for a specific day and a timetable that goes along with that routing.

6

HFOM, UNHAS case

The model is transformed into an application and can now be used to create more efficient and effective flight in South Sudan. The test case used is based on a typical 5 day week of operations of UNHAS in South Sudan. This Chapter will discuss the test data and the tuning of the parameters. Finally, the results that are produced will be presented.

6.1. Test Data

In order for the HFOM to get a result, the test data is needed. This data is used to construct the objective function, the decision variables, the parameters and the constraints. The data that is needed must contain the following information:

1. Five consecutive days of demand data
2. Airport coordinates and constraints
3. Fleet data
4. Operational constraints

As mentioned five days of data are used to test the HFOM. This data can be found in Appendix B, Figure B.1. The airport coordinates can be found in Figure B.2 and B.3. The fleet data is given in Chapter 3, Figure 3.4, together with the operational constraints.

6.2. Tuning of Parameters

While testing the HFOM, it was found that the computing time to solve the full problem would exceed 10 hours. In order to improve the performance of the application, the heuristic as mentioned in Chapter 5 is applied and the optimization engine parameters are tweaked. The optimization engine parameters will be discussed below:

Optimization Engine parameters

The optimization of a problem is always a trade off between finding the optimal solution and the time it takes to get that optimal solution. Using optimization software like CPLEX and Gurobi, many settings can be changed that force the program to find a solution more quickly, but in return optimization towards the optimal solution might be increased. The optimization engine is mostly a black box in terms of the processes involved in getting to a solution to the optimization problem. Once the objective, constraints and decision variables have been correctly loaded in the software, several heuristics take over in finding solutions. What is known is that for mixed integer linear programming models, mixed integer optimizers, like branch-and-cut search. The optimizer works in such a way that it "builds a tree with the linear relaxation of the original MIP at the root and subproblems to optimize at the nodes of the tree." (*IBM ILOG CPLEX Optimization Studio*, 2019). Every node represents a linear programming subproblem to be solved and checked for integrality. Nodes are processed until no more nodes are available or a time limit has been reached.

Two parameters are being altered when running the optimization, being the (1) time limit and the (2) MIP emphasis or focus. More parameters exist that influence the performance of optimization engines, however these were found to be of greatest interest.

Time limit: The time limit is simply a limit on how much time the optimization engine should take to find a solution. If no time limit is set, the optimization engine will run until an optimal solution has been found. If a time limit is set, the best feasible solution generated up to that limit will be passed as the solution. The HFOM is split up in sub-problems and each problem has a set time limit. In order to find a results within an hour, it was empirically found that by firstly running a 'short' 300 second, 3-leg problem and using that as a starting point for a 'long' 600 second, 4/5-leg problem, the HFOM is capable of finding a good and quick solution.

MIP emphasis/focus: Another parameter that is used to help generate better solutions is the MIPemphasis (for CPLEX) or the MIPFocus (for Gurobi). This parameter influences the emphasis on feasibility and optimality. The default setting aims to find a balance between finding new feasible solutions and proving optimality. However, more attention can also be put on finding feasible solutions quickly or on proving optimality. In the case of the HFOM, the 'short' 300 second, 3-leg problem has an emphasis on feasibility and the 'long' 600 second, 4/5-leg problem has emphasis on balancing feasibility and optimality. It was found by testing that these options worked best for this specific problem.

6.3. HFOM Results

The Humanitarian Flight Optimization Model is now able to produce results after loading the input data and setting the performance tuning settings. The results produced are both in tabular form and in graphical format. The tabular form shows the schedule with the timings of the individual aircraft and the graphical format shows a plot of the area with the flights for each aircraft. A full week of scheduling (five days) will be presented and the first day will have an in-depth discussion of the results, including an analysis of the results when parameters like the time limit, the leg connections and the demand satisfaction are changed. All these factors influence the results and having a good insight in these parameters can lead to better schedules and improved utilization of the aircraft. Section 6.3.1 shows an in-depth analysis of day 1, the following sections give the results for the other days, two through five. Only day 1 will be analysed intensively, providing details about the application and describing the benefit of using the model. The results are discussed based on the topics of aircraft routing cost and passenger demand satisfaction.

6.3.1. Day 1

This section will discuss the routing and scheduling of test day 1. The request for this day can be found in Appendix B, Figure B.1. The results for this day are assessed by the creation of the routes while changing certain parameters. These parameters include the time limit on the optimization of the sub-problems, the maximum amount of legs in a route of an aircraft and the demand satisfied per sub-problem. Each of these parameters have an impact on the cost and are therefore of major interest. The trade off curve, or Pareto front for the problem with 3-leg, 4-leg and 5-leg routings is shown in Figure 6.1. This graph will be analyzed in-depth, as this Pareto front shows the routes found when optimizing for different settings. Each point in the plot corresponds to a full day of routing and scheduling and shows the passengers transported for that routing along with the cost that is associated with it. All the runs have been done with a 300 second time limit for each sub-problem. If an optimal solution was found before that limit, the solution was passed as optimal and the next sub-problem would be started. These points are further broken down in Appendix C, Tables C.1, C.4 and C.7. These Tables show, for the 3-leg, 4-leg and 5-leg routings respectively, the routing cost when a change is made in how many people *have* to be transported in a certain region. An even more detailed breakdown of these routing costs is given in Appendix C, Tables C.2, C.5 and C.8. The differences in routings is given in these Tables, along with the difference in cost. Next, the routing and scheduling for a 4-leg routing will be given, as is received as an output from the application. This 4-leg routing is capable of getting a solution within a maximum of 25 minutes (300 seconds for each of the six sub-problems) and satisfies a demand that is based on the demand pre-determined by the heuristic given in Section 5.1.1. As mentioned, the Pareto front in Figure 6.1 shows the trade off in multi-objective optimization between, in this case, two objectives. The (1) demand satisfaction on the x-axis is plotted against the (2) cost of the total routing on the y-axis. These objectives are connected in such a way that both have to be traded off against each other. A high demand satisfaction might lead to a higher routing cost as more aircraft will be needed to transport the

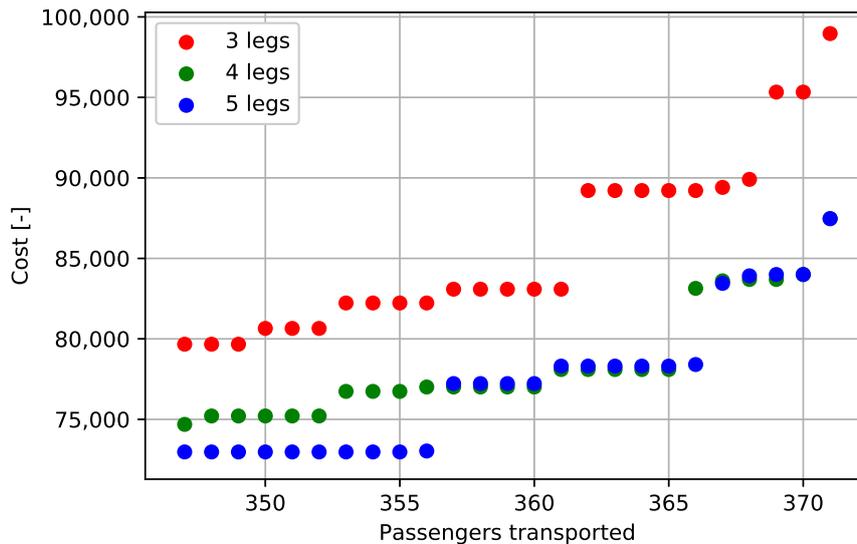


Figure 6.1: Pareto front plot, 3/4/5 legs, day 1

increase in passengers. However, it could be that the capacity of an aircraft is filled to exactly the maximum with the higher demand satisfaction. In that case, the routing cost will not have increased but stayed the same, albeit with a higher demand satisfaction. Those points are of the highest interest on the Pareto front, as these points give the best routing cost with the highest demand satisfaction. All the points correspond to a unique flight schedule. The main conclusion to be drawn from this plot is that the 4-leg routing is the best overall. The cost for the 3-leg routing is higher for every demand satisfaction (by roughly 7,000) and although the 5-leg routing is lower in most cases, the application could not always find a solution in the time-frame that was given. The solution of the 4-leg routing was always used as a starting point. For this to be used in real life by a Flight Tasking Officer (FTO), it would be best to run a 4-leg routing first and finally check with a 5-leg routing if a better route can be found. The Pareto front shows the strength of the application in that it can find multiple routings for multiple settings. A decision on how many passengers are transported can now be made with more knowledge on the impact of such a decision. The optimiser advises the planner on the cost effects when aiming for a certain demand satisfaction and to select the corresponding flight schedule.

Finally, the routing for this day is given by the routing and scheduling provided in Table 6.2 and Figure 6.2. This routing is based on a maximum of 5 legs for the 'F50' and 'Rumbek' regions and a maximum of 4 legs for the 'Juba' and 'Bor' regions. The division heuristic given in Section 5.1.1 filtered out several requests to be spilled up front, leading to a 99% demand satisfaction. The requests that have been spilled and the requests that have been transported can be found in Table 6.3. Table 6.1 shows the different costs associated with different time limits and routings based on the division that has been made.

		CPLEX		Gurobi	
		Cost [-]	Time [s]	Cost [-]	Time [s]
Day 1	300s, 3 legs	89789	903	89089	903
	600s, 4/5 legs	79974	1803	79958	1803
Day 2	300s, 3 legs	96191	1201	96191	1062
	600s, 4/5 legs	91918	2105	91942	1870
Day 3	300s, 3 legs	81477	448	81477	532
	600s, 4/5 legs	72519	733	72519	691
Day 4	300s, 3 legs	97589	911	108418	975
	600s, 4/5 legs	87414	1865	94030	1824
Day 5	300s, 3 legs	102842	490	102842	347
	600s, 4/5 legs	92773	1326	92773	831

Table 6.1: Results HFOM using CPLEX and Gurobi

	Origin	Destination	Departure time	Arrival time	Leg cost
Fokker 50	Juba	Wau	07:15	08:27	5518
	Wau	Aweil	08:47	09:06	1477
	Aweil	Rumbek	09:26	10:13	3631
	Rumbek	Juba	11:20	12:03	3276 (13902)
Dash 8_1	Juba	Rubkona	07:15	08:41	5165
	Rubkona	Juba	09:01	10:27	5165 (10330)
Dash 8_2	Juba	Rumbek	07:15	08:04	2785
	Rumbek	Juba	08:24	09:13	2785
	Juba	Malakal	09:33	10:57	4771
	Malakal	Juba	11:17	12:42	4771
	Juba	Yambio	13:02	13:59	3242
	Yambio	Juba	14:19	15:16	3242 (21598)
Dornier 228	Juba	Pibor	07:15	07:55	1620
	Pibor	Juba	08:15	08:55	1620 (3240)
Cessna 208_1	-	-	-	-	-
Cessna 208_2	Juba	Bor	07:15	07:41	728
	Bor	Pibor	08:01	08:34	900
	Pibor	Bor	08:54	09:26	900
	Bor	Juba	09:46	10:13	728
	Juba	Yida	10:33	12:22	3012
	Yida	Juba	12:42	14:31	3012 (9280)
Cessna 208_3	Juba	Yida	07:15	09:04	3110
	Yida	Rubkona	09:24	09:41	486
	Rubkona	Mankien	10:01	10:16	423
	Mankien	Juba	10:36	12:13	2771 (6790)
Cessna 208_4	-	-	-	-	-
Cessna 208_5	-	-	-	-	-
Mil Mi8_1	Juba	Mingkaman	07:15	07:50	2252
	Mingkaman	Juba	08:10	08:45	2252 (4504)
Mil Mi8_2	-	-	-	-	-
Cessna 208_1R	-	-	-	-	-
Cessna 208_2R	Rumbek	Agok	08:30	09:25	1735
	Agok	Rumbek	09:45	10:40	1735 (3470)
Mil Mi8_1R	-	-	-	-	-
Mil Mi8_2R	Rumbek	Koch	08:30	09:24	3379
	Koch	Leer	09:44	09:54	595
	Leer	Rumbek	10:14	11:00	2888 (6862)

Table 6.2: Flight schedule for day 1

Origin	Destination	Request	Transported by	Spill
Agok	Juba	9	Fokker 50, Cessna 208_2R	
Aweil	Juba	14	Fokker 50	
Aweil	Rumbek	2	Fokker 50	
Bor	Juba	6	Cessna 208_2	
Bor	Pibor	8	Cessna 208_2	
Juba	Agok	11	Dash 8_2: 10, Cessna 208_2R: 10	1
Juba	Aweil	17	Fokker 50	
Juba	Bor	6	Cessna 208_2	
Juba	Koch	4	Dash 8_2, Mil Mi8_2R	
Juba	Leer	4	Dash 8_2, Mil Mi8_2R	
Juba	Malakal	35	Dash 8_2	
Juba	Mingkaman	15	Mil Mi8_1	
Juba	Mankien	1	Cessna 208_3	
Juba	Pibor	16	Dornier 228: 15	1
Juba	Rubkona	28	Dash 8_1	
Juba	Rumbek	13	Fokker 50	
Juba	Wau	27	Fokker 50	
Juba	Yambio	12	Dash 8_2	
Juba	Yida	16	Cessna 208_2: 10, Cessna 208_3: 5	1
Koch	Juba	1	Fokker 50, Mil Mi8_2R	
Leer	Juba	5	Fokker 50, Mil Mi8_2R	
Malakal	Juba	16	Dash 8_2	
Mingkaman	Juba	7	Mil Mi8_1	
Pibor	Juba	9	Dornier 228	
Rubkona	Juba	15	Dash 8_1	
Rubkona	Mankien	2	Cessna 208_3	
Rumbek	Juba	20	Dash 8_2	
Wau	Juba	20	Fokker 50	
Wau	Rumbek	2	Fokker 50	
Yambio	Juba	20	Dash 8_2	
Yida	Juba	10	Cessna 208_2	

Table 6.3: Transport breakdown for day 1

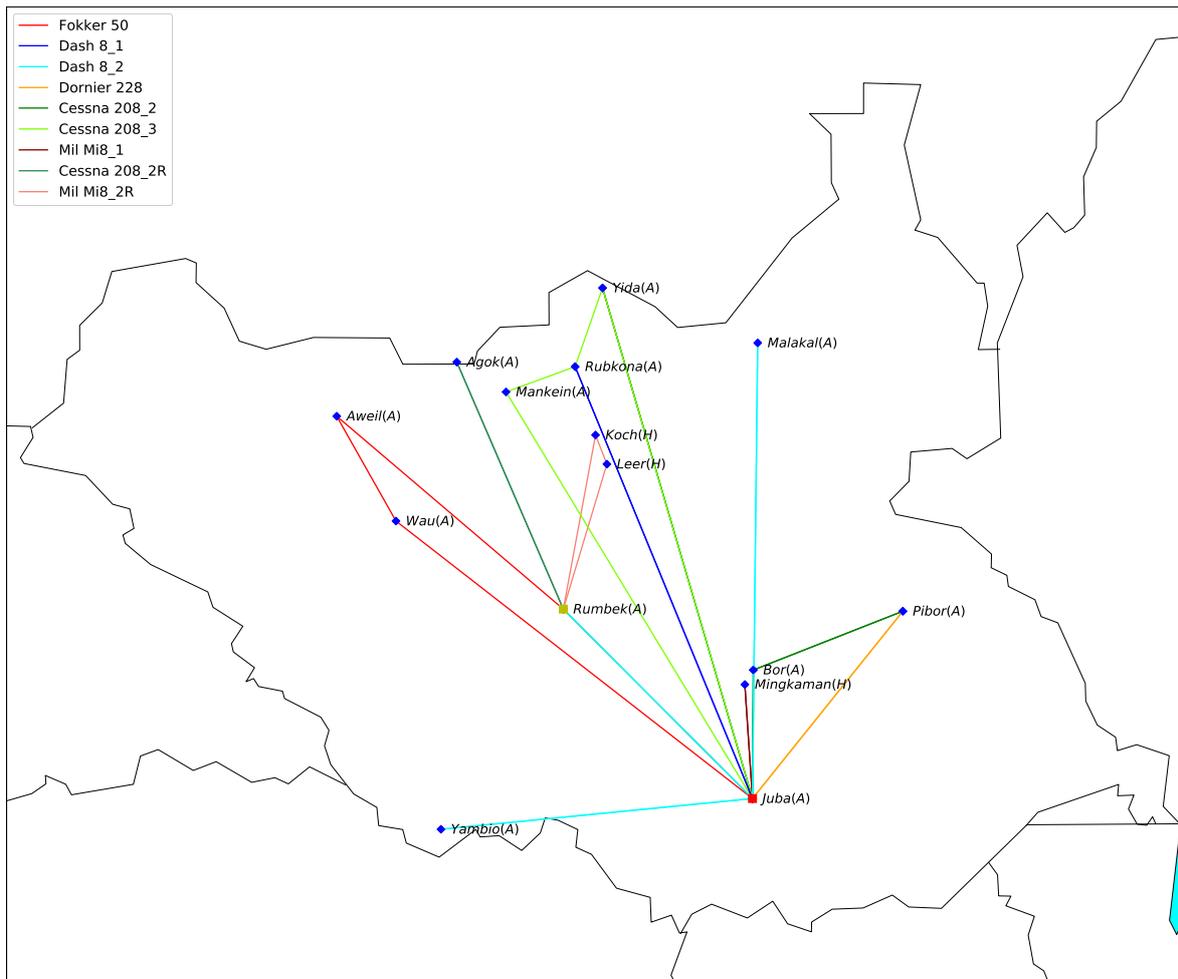


Figure 6.2: Results of day 1, objective 79974, pax 368/371

6.3.2. Day 2

The routing and scheduling for day 2 can be found in Appendix E, Figure E.1 and Tables E.1 and E.2. A brief overview of the routing cost is given in Table 6.1, along with the routing cost for other parameters. This is done in order to provide a clear perspective of the impact that the time limit or the maximum amount of legs in a routing might have on the total routing cost.

6.3.3. Day 3

The routing and scheduling for day 3 can be found in E, Figure E.2 and Tables E.3 and E.4. A brief overview of the routing cost is given in Table 6.1, along with the routing cost for other parameters.

6.3.4. Day 4

The routing and scheduling for day 4 can be found in E, Figure E.3 and Tables E.5 and E.6. A brief overview of the routing cost is given in Table 6.1, along with the routing cost for other parameters.

6.3.5. Day 5

The routing and scheduling for day 5 can be found in E, Figure E.4 and Tables E.7 and E.8. A brief overview of the routing cost is given in Table 6.1, along with the routing cost for other parameters.

7

Verification and Validation

This Chapter will focus on the verification and validation of the model. The results of the model be checked to ensure that they are correct (verification) and if they are valuable (validation). The computational verification is done in Section 7.1 and Section 7.2 validates the model by comparing the results of different simulation days to the schedules made by expert planners.

7.1. Model Verification

The verification is focused on the verifying the computational correctness of the optimization model and the heuristics that have been introduced in the Humanitarian Flight Optimization Model.

The heuristic is checked by tracing the test data set of day 1 (see Figure B.1 in Appendix B). Firstly, the loading of the data and the division heuristic will be checked, followed with the node creation and edge removal process. The optimization block is checked by unit testing the writing of the constraints and objective function in the lp file. The data that will be used is given in Table 7.3 and consists of 5 requests.

7.1.1. Heuristic Verification

The heuristic as described in Section 5.1.1 will be verified by tracing test day 1, given in Appendix B. The loading of the data is checked by inspecting the lists and dictionaries created in Python. The main loop that is used to process the data and which connects the airports to coordinates and runway lengths is inspected and deemed to function properly. Since the local dataset that is created will be used in the following heuristic steps no further verification of the data loading is necessary (the confirmation that it is indeed correct will be given by the tracing). The division heuristic is described in Section 5.1.1 will be traced using the requests of test day 1. Table 7.1 shows the outcome of this division heuristic.

In Table 7.1a requests marked with the asterisks * after JUB and RUM are there to indicate that these request are virtual and have been added by the heuristic. The JUB-RUM connection of 8 passengers is actually the JUB-KOCH and JUB-LER connection added together (4 and 4 passengers respectively). The RUM-JUB connection of 6 passengers is the KOCH-JUB and LER-JUB connection added together (1 and 5 passengers respectively). These requests can be found in Table 7.1d.

The requests with asterisks a number within parentheses in Table 7.1b and 7.1c are requests that have been split and might have spilled passengers. For example, the JUB-PIBR request has 16 passengers and has been split in one of 10 and one of 5 passengers. The spillage for this request is equal to 1 passenger. Since MINGK is a helipad, up to 17 passengers can be transported. Therefore, a request to and from MINGK is not split up.

Table 7.1 shows the lists that the heuristic has created. The regions are correctly split and the requests have been split correctly and the spilled requests are handled in the correct manner.

The node creation, edge removal, decision variable removal and parameter creation are all traced for the data of test day 1. All the aforementioned actions are performed correctly. Table 7.2 gives an overview of the tracing that has been performed. With this tracing, the heuristic analysis is finished and the verification is considered completed.

(a) 'F50' region				(b) 'Bor' region				(c) 'Juba North' region			
#	Origin	Destination	Request	#	Origin	Destination	Request	#	Origin	Destination	Request
1	AWL	JUB	14	1	BOR	JUB	6	1	JUB	MAK	18 (35)
2	AWL	RUM	2	2	BOR	PIBR	8	2	JUB	MNK	1
3	JUB	AWL	17	3	JUB	BOR	6	3	JUB	RUB	28
4	JUB	RUM	13	4	JUB	MINGK	15	4	JUB	YIDA	10 (16)
5	JUB	WAU	27	5	JUB	PIBR	10 (16)	5	MAK	JUB	8 (16)
6	RUM	JUB	20	6	MINGK	JUB	7	6	RUB	JUB	15
7	WAU	JUB	20	7	PIBR	JUB	9	7	RUB	MNK	2
8	WAU	RUM	2	8	JUB	PIBR	5 (16)	8	YIDA	JUB	10
9	JUB	RUM (*)	18					9	JUB	MAK	17 (35)
10	RUM (*)	JUB	15					10	JUB	YIDA	5 (16)
								11	MAK	JUB	8 (16)

(d) 'Rumbek 1' region				(e) 'Rumbek 2' region				(f) 'Juba South' region			
#	Origin	Destination	Request	#	Origin	Destination	Request	#	Origin	Destination	Request
1	RUM (JUB*)	KOCH	4	1	AGOK	RUM (JUB*)	9	1	JUB	YAM	12
2	RUM (JUB*)	LER	4	2	RUM (JUB*)	AGOK	10 (11)	2	YAM	JUB	20
3	KOCH	RUM (JUB*)	1								
4	LER	RUM (JUB*)	5								

Table 7.1: Pre-processing, dividing and splitting requests, day 1

Expected	LP file
Correct division	True, checked ✓
Correct split and spilled requests	True, checked ✓
Correct node creation	True, checked ✓
Correct edge removal	True, checked ✓
Correct decision variable removal	True, checked ✓
Correct parameter creation	True, checked ✓

Table 7.2: Heuristic verification

7.1.2. Linear Program writing Verification

The writing of the objective function and the constraints is checked by taking the problem given in Table 7.3 and checking whether the the linear program is written as intended. The node creation was checked in the previous section and for convenience the nodes that are created for this problem are given in 7.4. The nodes that will be traced are nodes 2 (Rumbek, origin), 9 (Yida, destination) and 11 (Jub, starting hub) and 12 (Jub, ending hub) for the Fokker 50 (vehicle $k = 1$) and the Cessna 208 (vehicle $k = 4$). As can be seen a pickup node, a delivery node and a hub node will be checked. Only possible connections between these nodes will be checked.

Request number	Origin	Destination	Request quantity
1	JUB	RUM	35
2	RUM	JUB	49
3	JUB	RUB	10
4	RUB	YIDA	6
5	YIDA	JUB	9

Table 7.3: Test instance used for verification

	Node	Location	Request quantity
	1	JUB	35
	2	RUM	49
Origin (P)	3	JUB	10
	4	RUB	6
	5	YIDA	9
	6	RUM	-35
	7	JUB	-49
Destination (D)	8	RUB	-10
	9	YIDA	-6
	10	JUB	-9
Hubs (start)	11	JUB	0
Hubs (end)	12	JUB	0

Table 7.4: Test instance processed

Objective function

The objective function is given in 4.27 and consists of the decision variable x_{ij}^k and the cost of flying on leg i to j , c_{ij}^k (distance i to j times cost of vehicle k). Table 7.5 shows the results of the verification. The results check out and therefore the objective function is written correctly. It is also checked that the lp file indeed correctly states that the problem is a minimization problem. It should be noted that the edges from 2 (origin) to 11 (starting hub) and 12 (ending hub), 8 (destination) to 11, 11 to 8 and 12 to 2, 8 and 11 do not exist and are therefore not present in the objective function or in Table 7.5.

Arc, start	Arc, end	Fokker 50		Cessna 208	
		Expected	LP file	Expected	LP file
Node 2 (Rumbek)	9 (Yida)	$20 \cdot 199.235 = 3984.7 x_{2,9}^1$	$3984.7 x_{2_9_1}$	$10 \cdot 199.235 = 1992.3 x_{2,9}^4$	$1992.3 x_{2_9_4}$
Node 9 (Yida)	2 (Rumbek)	$3984.7 x_{9,2}^1$	$3984.7 x_{9_2_1}$	$1992.3 x_{9,2}^4$	$1992.3 x_{9_2_4}$
	12 (Juba)	$20 \cdot 327.339 = 6546.8 x_{9,12}^1$	$6546.8 x_{9_12_1}$	$10 \cdot 327.339 = 3273.4 x_{9,12}^4$	$3273.4 x_{9_12_4}$
Node 11 (Juba)	2 (Rumbek)	$20 \cdot 163.828 = 3276.6 x_{11,2}^1$	$3276.6 x_{11_2_1}$	$10 \cdot 163.828 = 1638.3 x_{11,2}^4$	$3276.6 x_{11_2_4}$
	12 (Juba)	$0 x_{11,12}^1$	$0 x_{11_12_1}$	$0 x_{11,12}^4$	$0 x_{11_12_4}$

Table 7.5: Verification of the objective function

Customers served constraint unit test

The customers served constraint (given in 4.10) is now checked with the same five requests test set. The constraints should be written in such a way that the origin i is taken from the set of P (number of origins), so in this case five constraints need to be written. Then the sum of all the edges that exists for the specific i in set A and all the vehicles k in set K has to be lower than 1. Table 7.6 gives a short overview of this unit test.

Expected	LP file
5 constraints	5 constraints
sum of all $x_{1,j}^k$ for all j that belong to $i=1$ in A and all k in K lower than 1	True, checked ✓

Table 7.6: Verification of customers served constraint

Start and end hub unit test

Both the start and end hub constraint (given in 4.11 and 4.12) are tested. The constraints should be written in such a way that each vehicle k is taken from the set of K , so in this case five constraints need to be written for both constraints. Then the sum of all the edges that exists for the $i = 11$ in set A and $j = 12$ in set A for a certain k in K needs to be equal to 1, for the start and end hub constraint respectively. Table 7.7 shows that this is indeed true.

Expected	LP file
5 constraints	5 constraints
sum of all $x_{11,j}^1$ for all j that belong to $i=11$ in A and $k=1$ equal to 1	True, checked ✓
sum of all $x_{i,12}^1$ for all j that belong to $j=12$ in A and $k=1$ equal to 1	True, checked ✓

Table 7.7: Verification of start and end hub constraint

Flow conservation unit test

The next unit test is for the flow conservation constraint (given in 4.13). For each origin i from the set P and D and each vehicle k in K , the constraint should be made. The sum of all $x_{j,i}^k$ for all j that belong to the specific i and vehicle k minus sum of all $x_{i,j}^k$ for all j that belong to i and vehicle k should be equal to 0. It can be seen from Table 7.8 that this holds.

Expected	LP file
10·5=50 constraints sum of all $x_{j,1}^1$ for all j that belong to $i=1$ in A and $k=1$ minus sum of all $x_{1,j}^1$ for all j that belong to $i=1$ in A and $k=1$ equal to 0	50 constraints True, checked ✓

Table 7.8: Verification of flow conservation constraint

Same vehicle served unit test

The next unit test is for the flow conservation constraint (given by equation 4.14). For each origin i from the set P and each vehicle k in K , the constraint should be made. The sum of all $x_{i,j}^k$ for all j that belong to the specific i and vehicle k minus sum of all $x_{n+i,j}^k$ for all j that belong to $n+i$ and vehicle k should be equal to 0. Table 7.9 holds verification.

Expected	LP file
5·5=25 constraints sum of all $x_{1,j}^1$ for all j that belong to $i=1$ in A and $k=1$ minus sum of all $x_{6,j}^1$ for all j that belong to $i=6$ in A and $k=1$ equal to 0	25 constraints True, checked ✓

Table 7.9: Verification of same vehicle served constraint

Timing unit test

The timing is controlled by three constraints (refer to equations 4.15, 4.16 and 4.17). The constraints will be checked by looking at again the Fokker 50 and Cessna 208, whom have different speeds (230 nm/hr and 180 nm/hr respectively) and nodes 2 (Rumbek), 9 (Yida), 11 (Juba) and 12 (Juba), as was done for the objective function. The confirmation of constraint 4.15 can be found in Table 7.10. The time t_{ij}^k it takes to get from i to j with vehicle k is simply the distance from i to j divided by the speed of vehicle k . The TAT s_i^k is added, which is equal to 20 minutes for each vehicle, but zero when the nodes lie on top of each other.

Arc, start	Arc, end	Fokker 50		Cessna 208	
		Expected	LP file	Expected	LP file
Node 2 (Rumbek)	9 (Yida)	$w_2^1 - w_9^1 \leq -(199.235/230) - 20/60 = -1.199$	$w_{2_1} - w_{9_1} \leq -1.199$	$w_2^4 - w_9^4 \leq -(199.235/180) - 20/60 = -1.44$	$w_{2_4} - w_{9_4} \leq -1.44$
Node 9 (Yida)	2 (Rumbek)	$w_9^1 - w_2^1 \leq -(199.235/230) - 20/60 = -1.199$	$w_{9_1} - w_{2_1} \leq -1.199$	$w_9^4 - w_2^4 \leq -(199.235/180) - 20/60 = -1.44$	$w_{9_4} - w_{2_4} \leq -1.44$
	12 (Juba)	$w_9^1 - w_{12}^1 \leq -(327.339/230) - 20/60 = -1.76$	$w_{9_1} - w_{12_1} \leq -1.76$	$w_9^4 - w_{12}^4 \leq -(327.339/180) - 20/60 = -2.15$	$w_{9_4} - w_{12_4} \leq -2.15$
Node 11 (Juba)	2 (Rumbek)	$w_{11}^1 - w_2^1 \leq -(163.828/230) - 20/60 = -1.04$	$w_{11_1} - w_{2_1} \leq -1.04$	$w_{11}^4 - w_2^4 \leq -(163.828/180) - 20/60 = -1.24$	$w_{11_4} - w_{2_4} \leq -1.24$
	12 (Juba)	$w_{11}^1 - w_{12}^1 \leq -(0/230) = 0$	$w_{11_1} - w_{12_1} \leq 0$	$w_{11}^4 - w_{12}^4 \leq -(0/180) = 0$	$w_{11_4} - w_{12_4} \leq 0$

Table 7.10: Verification of first timing constraint

Constraint 4.16 is the constraint that deals with the time windows and in this case the check for the block times. A vehicle k should arrive at and depart from node i between time a_i^k and b_i^k . The check of this with a time window of 10 hours for the Fokker 50 and a time windows of 7 hours for the Cessna 208 is given in Table 7.11.

The final timing constraint (4.17) will be checked for node 2 and again the Fokker 50 and the Cessna 208. The destination of request 2 is node 7 (Juba). For each origin i from the set P and each vehicle k in K , the constraint should be made. Table 7.12 verifies that this constraint is correctly being written.

Node	Time	Fokker 50		Cessna 208	
		Expected	LP file	Expected	LP file
Node 2 (Rumbek)	start	$w_2^1 \geq 0$	$w_{_2_1} \geq 0$	$w_2^4 \geq 0$	$w_{_2_4} \geq 0$
	end	$w_2^1 \leq 10$	$w_{_2_1} \leq 10$	$w_2^4 \leq 7$	$w_{_2_4} \leq 7$
Node 9 (Yida)	start	$w_9^1 \geq 0$	$w_{_9_1} \geq 0$	$w_9^4 \geq 0$	$w_{_9_4} \geq 0$
	end	$w_9^1 \leq 10$	$w_{_9_1} \leq 10$	$w_9^4 \leq 7$	$w_{_9_4} \leq 7$
Node 11 (Juba)	start	$w_{11}^1 \geq 0$	$w_{_11_1} \geq 0$	$w_{11}^4 \geq 0$	$w_{_11_4} \geq 0$
	end	$w_{11}^1 \leq 10$	$w_{_11_1} \leq 10$	$w_{11}^4 \leq 7$	$w_{_11_4} \leq 7$
Node 12 (Juba)	start	$w_{12}^1 \geq 0$	$w_{_12_1} \geq 0$	$w_{12}^4 \geq 0$	$w_{_12_4} \geq 0$
	end	$w_{12}^1 \leq 10$	$w_{_12_1} \leq 10$	$w_{12}^4 \leq 7$	$w_{_12_4} \leq 7$

Table 7.11: Verification of second timing constraint

Arc, start	Fokker 50		Cessna 208	
	Expected	LP file	Expected	LP file
Node 2 (Rumbek)	$w_2^1 - w_7^1 \leq -(163.828/230) = -0.71$	$w_{_2_1} - w_{_7_1} \leq -0.71$	$w_2^4 - w_7^4 \leq -(163.828/180) = -0.91$	$w_{_2_4} - w_{_7_4} \leq -0.91$

Table 7.12: Verification of third timing constraint

Loading unit test

The loading is controlled by two constraints (refer to equations 4.18 and 4.19). The same unit test will done as the timing constraint. The confirmation of constraint 4.18 can be found in Table 7.13. The difference between the load at node i and the load at node j for vehicle k , given by u_i^k and u_j^k respectively has to be lower or equal to the request q_j . The reader is referred to Table 7.4 for the requests for this test.

Arc, start	Arc, end	Fokker 50		Cessna 208	
		Expected	LP file	Expected	LP file
Node 2 (Rumbek)	9 (Yida)	$u_2^1 - u_9^1 \leq 6$	$u_{_2_1} - u_{_9_1} \leq 6$	$u_2^4 - u_9^4 \leq 6$	$u_{_2_4} - u_{_9_4} \leq 6$
Node 9 (Yida)	2 (Rumbek)	$u_9^1 - u_2^1 \leq -49$	$u_{_9_1} - u_{_2_1} \leq -49$	$u_9^4 - u_2^4 \leq -49$	$u_{_9_4} - u_{_2_4} \leq -49$
	12 (Juba)	$u_9^1 - u_{12}^1 \leq 0$	$u_{_9_1} - u_{_12_1} \leq 0$	$u_9^4 - u_{12}^4 \leq 0$	$u_{_9_4} - u_{_12_4} \leq 0$
Node 11 (Juba)	2 (Rumbek)	$u_{11}^1 - u_2^1 \leq -49$	$u_{_11_1} - u_{_2_1} \leq -49$	$u_{11}^4 - u_2^4 \leq -49$	$u_{_11_4} - u_{_2_4} \leq -49$
	12 (Juba)	$u_{11}^1 - u_{12}^1 \leq 0$	$u_{_11_1} - u_{_12_1} \leq 0$	$u_{11}^4 - u_{12}^4 \leq 0$	$u_{_11_4} - u_{_12_4} \leq 0$

Table 7.13: Verification of first loading constraint

Constraint 4.19 is very similar to the second timing constraint and will be tested identically. The difference is that only an upper bound exists and no lower bound. Table 7.14 gives that verification that the constraint holds true. The load for the Fokker 50 has to be lower than the capacity of 50 and the same goes for the Cessna 208, which has a capacity of 10.

Node	Fokker 50		Cessna 208	
	Expected	LP file	Expected	LP file
Node 2 (Rumbek)	$u_2^1 \leq 50$	$u_{_2_1} \leq 50$	$u_2^4 \leq 10$	$u_{_2_4} \leq 10$
Node 9 (Yida)	$u_9^1 \leq 50$	$u_{_9_1} \leq 50$	$u_9^4 \leq 10$	$u_{_9_4} \leq 10$
Node 11 (Juba)	$u_{11}^1 \leq 50$	$u_{_11_1} \leq 50$	$u_{11}^4 \leq 10$	$u_{_11_4} \leq 10$
Node 12 (Juba)	$u_{12}^1 \leq 50$	$u_{_12_1} \leq 50$	$u_{12}^4 \leq 10$	$u_{_12_4} \leq 10$

Table 7.14: Verification of second loading constraint

Range constraint unit test

To ensure that the range of the different aircraft is not exceeded during a tour, the range constraints are added (refer to equations 4.20 and 4.21). The same unit test will done as the loading constraint. The confirmation of constraint 4.20 can be found in Table 7.13. The distance travelled at node j for vehicle k (v_j^k) has to be higher than or equal to the distance travelled at node i for vehicle k (v_i^k) plus the distance between i and j (d_{ij}). The reader is referred to Table B for the distance matrix.

Table 7.16 gives the confirmation that the constraint 4.21 holds true. The range for the Fokker 50 has to be lower than maximum range of 1080 nm and the same goes for the Cessna 208, which has a maximum range of 1070.

Arc, start	Arc, end	Fokker 50		Cessna 208	
		Expected	LP file	Expected	LP file
Node 2 (Rumbek)	9 (Yida)	$v_2^1 - v_9^1 \leq -199.23$	$v_{_2_1} - v_{_9_1} \leq -199.23$	$v_2^4 - v_9^4 \leq -199.23$	$v_{_2_4} - v_{_9_4} \leq -199.23$
Node 9 (Yida)	2 (Rumbek)	$v_9^1 - v_2^1 \leq -199.23$	$v_{_9_1} - v_{_2_1} \leq -199.23$	$v_9^4 - v_2^4 \leq -199.23$	$v_{_9_4} - v_{_2_4} \leq -199.23$
	12 (Juba)	$v_9^1 - v_{12}^1 \leq -327.34$	$v_{_9_1} - v_{_12_1} \leq -327.34$	$v_9^4 - v_{12}^4 \leq -327.34$	$v_{_9_4} - v_{_12_4} \leq -327.34$
Node 11 (Juba)	2 (Rumbek)	$v_{11}^1 - v_2^1 \leq -163.83$	$v_{_11_1} - v_{_2_1} \leq -163.83$	$v_{11}^4 - v_2^4 \leq -163.83$	$v_{_11_4} - v_{_2_4} \leq -163.83$
	12 (Juba)	$v_{11}^1 - v_{12}^1 \leq 0$	$v_{_11_1} - v_{_12_1} \leq 0$	$v_{11}^4 - v_{12}^4 \leq 0$	$v_{_11_4} - v_{_12_4} \leq 0$

Table 7.15: Verification of first range constraint

Node	Fokker 50		Cessna 208	
	Expected	LP file	Expected	LP file
Node 2 (Rumbek)	$v_2^1 \leq 1080$	$v_{_2_1} \leq 1080$	$v_2^4 \leq 1070$	$v_{_2_4} \leq 1070$
Node 9 (Yida)	$v_9^1 \leq 1080$	$v_{_9_1} \leq 1080$	$v_9^4 \leq 1070$	$v_{_9_4} \leq 1070$
Node 11 (Juba)	$v_{11}^1 \leq 1080$	$v_{_11_1} \leq 1080$	$v_{11}^4 \leq 1070$	$v_{_11_4} \leq 1070$
Node 12 (Juba)	$v_{12}^1 \leq 1080$	$v_{_12_1} \leq 1080$	$v_{12}^4 \leq 1070$	$v_{_12_4} \leq 1070$

Table 7.16: Verification of second range constraint

Runway constraint unit test

The runway constraint from equation 4.22 is now assessed and it is ensured that it is correctly pushed to the linear program. The end of the arc should have a runway that a specific vehicle k can land on. The runway of Rumbek, Juba and Yida are virtually 3000m, 3000m and 1000m respectively. The runway required by the Fokker 50 and the Cessna 208 are 3000m and 1000m respectively.

Arc, start	Arc, end	Fokker 50		Cessna 208	
		Expected	LP file	Expected	LP file
Node 2 (Rumbek)	9 (Yida)	$3000x_{2,9}^1 \leq 1000$	$3000x_{_2_9_1} \leq 1000$	$1000x_{2,9}^4 \leq 1000$	$1000x_{_2_9_4} \leq 1000$
Node 9 (Yida)	2 (Rumbek)	$3000x_{9,2}^1 \leq 3000$	$3000x_{_9_2_1} \leq 3000$	$1000x_{9,2}^4 \leq 3000$	$1000x_{_9_2_4} \leq 3000$
	12 (Juba)	$3000x_{9,12}^1 \leq 3000$	$3000x_{_9_12_1} \leq 3000$	$1000x_{9,12}^4 \leq 3000$	$1000x_{_9_12_4} \leq 3000$
Node 11 (Juba)	2 (Rumbek)	$3000x_{11,2}^1 \leq 3000$	$3000x_{_11_2_1} \leq 3000$	$1000x_{11,2}^4 \leq 3000$	$1000x_{_11_2_4} \leq 3000$
	12 (Juba)	$3000x_{11,12}^1 \leq 3000$	$3000x_{_11_12_1} \leq 3000$	$1000x_{11,12}^4 \leq 3000$	$1000x_{_11_12_4} \leq 3000$

Table 7.17: Verification of runway constraint

7.2. Model Validation

Two manual planners are used to validate the optimization model. Both planners have had multiple years of experience in creating daily schedules in South Sudan as Flight Tasking Officers. **Planner 1** is a former Colonel of the Fuerza Aerea Uruguaya and in this role he worked as a Logistic Officer, Pilot Instructor, Air Operations Director and ATM director. In 2001 he worked as a Chief Planning Officer at the Uruguayan Air Force and in 2005 he was Air Operations Director. In this role he planned, managed and conducted the air operations requested in support of the National Security, Medical Requirements, Disaster Support and all the administrative mission's requirements. From 2013 till 2014 he worked as a consultant for WFP Aviation, stationed in Juba. His duties were air asset management and logistics coordination. One of the tasks in this job involved the creation of the daily schedules for UNHAS in South Sudan. He has done this job for over a year and this has led to him having an excellent understanding of the situation in South Sudan and gave him the relevant experience needed to be called an expert planner. He currently works as a Flight Instructor at the Academia Latino Americana de Aviacion Superior ALAS. Furthermore, he is the author of the book 'Nyasala, una historia de aviadores y humanitarios en Sudan del Sur'. **Planner 2** holds a Master degree in Aerospace Engineering with a specialization in Design, Operation and Integration of Aircraft and Rotorcraft. From 2014 till 2015 he was stationed in South Sudan as a Flight Tasking Officer, managing and coordinating the daily flight operations focusing on the planning of all passenger flights. Currently he is active as an independent Aviations Operations Consultant, training and coaching professionals to improve their business processes to deliver more customer value and enhance the operations efficiency to achieve significant cost reductions.

Their expertise in creating these schedules will be tested against the Humanitarian Flight Optimization Model. This is the biggest test for the application as this is the first step in proving that humanitarian operations can be made more effective and efficient. It should be noted that the simulating a same environment as the one in South Sudan is impossible. No distractions were present during the making of these schedules

and all schedules were made with zero stress. In the field many more distractions are present, either through phone, e-mail, VHF radio communications or Air Transport Management wanting information on the coming flights. All these distractions have a negative impact on the schedule made by the planners as such events have an effect on the focus of a person. A loss of concentration leads to inefficient scheduling or even errors in the scheduling. A computer obviously does not have these distractions and will always produce the same results, even though the phone might be ringing. The checking of these schedules was done manually and took roughly 60 minutes per daily schedule. All schedules had some form of error, whether it was a missed request, a constraint violation or a wrong addition which made the flight exceed the capacity of the aircraft. These errors have been reported and Table 7.18 shows the results of the comparison of the Humanitarian Flight Optimization Model with the daily schedules created by the two expert planners. Each day will be discussed below. The results of the planners can be found in Appendix F. Figures 7.1, 7.2, 7.3, 7.4 and 7.5 show the results of the HFOM, the Planner 1 and the Planner 2 for the five days plotted with a Pareto front.

Schedule differences between HFOM and Planners

Day 1

Planner 1 had a difference of almost 10,000 in cost with 2 more passengers transported. This difference was mainly due to flying Juba-Yambio with two Cessna 208's (3624 cost per route) and flying to Aweil with a Fokker 50 and to Wau with a Dash 8 from Juba. The HFOM did a tour with the Fokker 50 to Rumbek, Aweil and Juba and did a direct flight to Rumbek with a Dash 8 saving roughly 4,000 this way.

Planner 2 A difference of roughly 3,000 in cost was found with 1 passenger extra being transported. Planner 2 also flew to Aweil with the Fokker 50 and to Wau with a Dash 8.

Day 2

Planner 1 had more than 4,000 higher cost with the same number of people being transported compared to the HFOM. The difference was mostly in flying directly to Paloich from Juba with a Cessna 208 and flying directly to Malakal from Juba with a Dash 8. The HFOM did both those destinations together with other requests.

Planner 2 had a difference of roughly 2,000 in cost with 1 passenger less transported. Planner 2 arranged a flight to Motot and Lankien with a Mil Mi8 from Juba. The HFOM connected the Lankien passengers in Rumbek and flew to Motot with a Cessna 208.

Day 3

Planner 1 had a difference of 8,000 in cost with 6 passengers more transported. Planner 1 did Juba-Rubkona & Rubkona-Juba, Juba-Malakal & Malakal-Juba and Juba-Maban & Maban-Juba all with separate Dash 8's.

Planner 2 had a difference of 1,500 in cost with 4 more passengers being transported. Planner 2 outperformed the HFOM by providing Juba-Wau and Wau-Juba with Cessna 208, where the HFOM is incapable of this routing (HFOM can only provide flights to and from Wau with a Fokker 50 or Dash 8). This is a limitation of the model.

Day 4

Planner 1 had a schedule with a 12,000 higher cost and all passengers transported. Old Fangak was connected with a Mil Mi8 from Rumbek. The HFOM did this connection with a Cessna 208. Juba-Pagak was flown directly with a Dornier 228, where the HFOM flew a double Juba-Akobo-Pagak-Juba tour with two Cessna 208's.

Planner 2 had a 3,000 higher cost in routing with an equal amount of passengers being transported. The difference was mainly from a Juba-Waat direct flight with a Mil Mi8. The HFOM did a Juba-Renk-Waat-Juba routing.

Day 5

Planner 1 created a schedule with 5,000 higher cost and the same amount of passengers transported as the HFOM. Planner 1 did direct flights to Rubkona, Malakal and Maban with three Dash 8's from Juba. The HFOM found routings that transported those requests with two Dash 8's.

Planner 2 had a 500 lower cost with two passengers less transported. Planner 2 found a routing with a Dash 8 that flew to both Yambio and Rumbek. The HFOM is incapable of this routing as no connection can be made between regions, only within.

		HFOM	Planner 1	Planner 2	Diff. HFOM Planner 1	Diff. HFOM Planner 2
Day 1 (Pax: 371)	Cost [-]	79974	89147	82964	-10.3%	-3.6%
	Time [hr]	0.75	2.0	3.5	-62.5%	-78%
	Demand satis.	368 (99.1%)	370 (99.7%)	369 (99.4%)	-0.5%	-0.2%
Day 2 (Pax: 298)	Cost [-]	91918	96290	94242	-4.5%	-2.4%
	Time [hr]	0.66	3.25	3.25	-79.7%	-79.7%
	Demand satis.	291 (97.6%)	291 (97.6%)	289 (96.9%)	0%	+0.5%
Day 3 (Pax: 293)	Cost [-]	72519	80846	74048	-10.3%	-2.1%
	Time [hr]	0.33	1.5	2.25	-78%	-85%
	Demand satis.	287 (97.9%)	293 (100%)	291 (99.3%)	-2.0%	-1.4%
Day 4 (Pax: 305)	Cost [-]	87414	99353	90495	-12.8%	-3.4%
	Time [hr]	0.75	2.25	2.0	-66.6%	-62.5%
	Demand satis.	305 (100%)	305 (100%)	302 (99.0%)	0%	+0.9%
Day 5 (Pax: 420)	Cost [-]	92773	98032	92390	-5.3%	+0.4%
	Time [hr]	0.5	3.5	2.5	-85%	-80%
	Demand satis.	417 (99.3%)	417 (99.3%)	415 (98.8%)	0%	+0.4%
Total week	Cost [-]	424598	463668	434138	-8.4%	-2.2%
	Time [hr]	3.0	12.5	13.5	-76%	-78%
	Demand satis.	1668 (98.9%)	1676 (99.3%)	1666 (98.7%)	-0.4%	+0.1%

Table 7.18: Overview of expert system validation

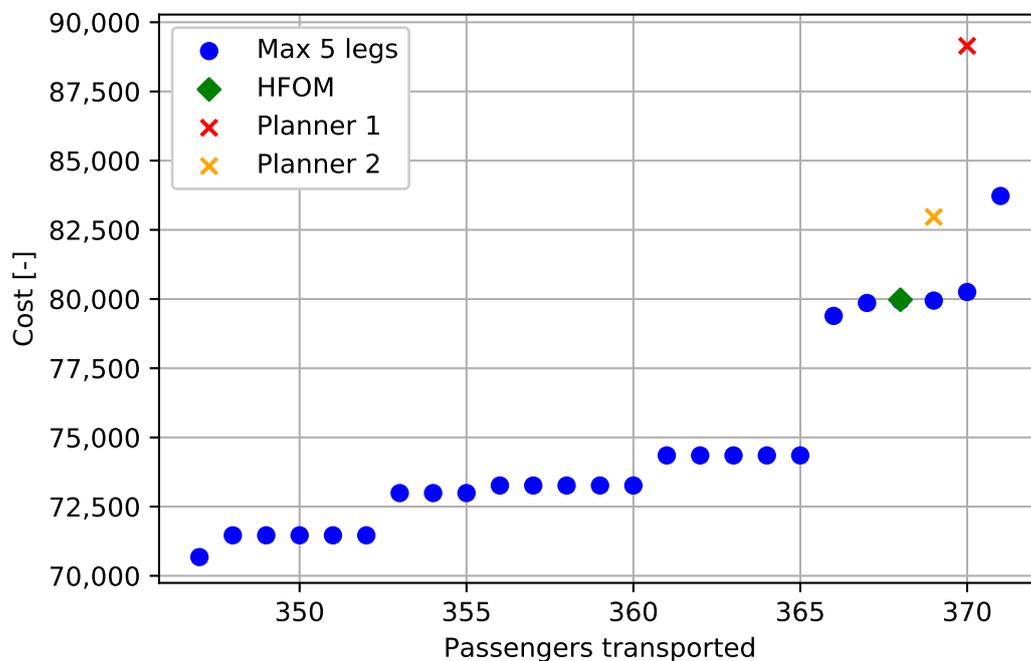


Figure 7.1: Pareto front plot, day 1

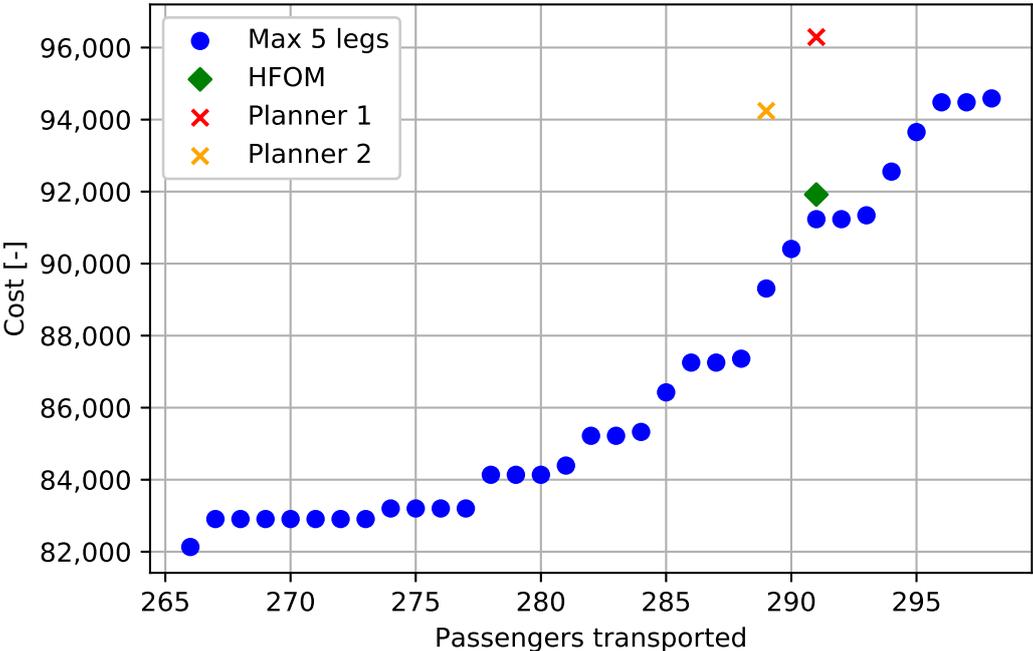


Figure 7.2: Pareto front plot, day 2

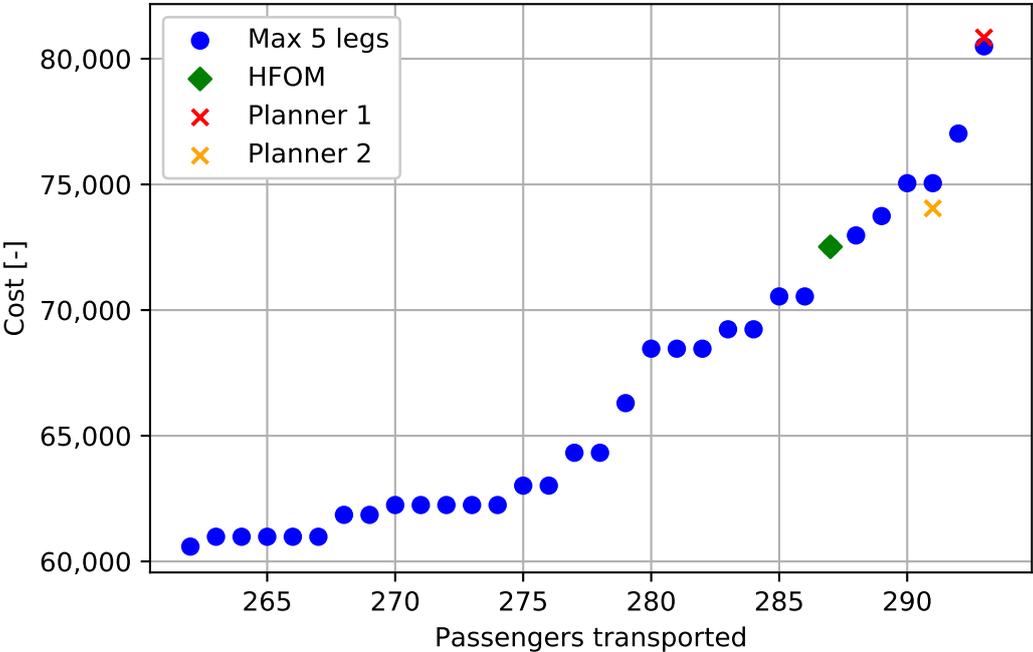


Figure 7.3: Pareto front plot, day 3

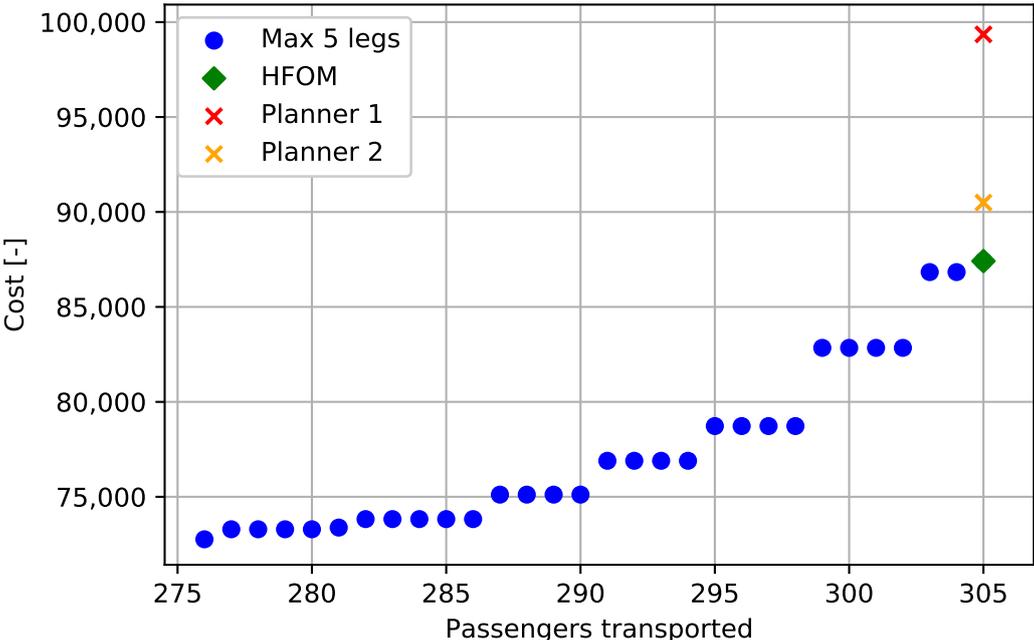


Figure 7.4: Pareto front plot, day 4

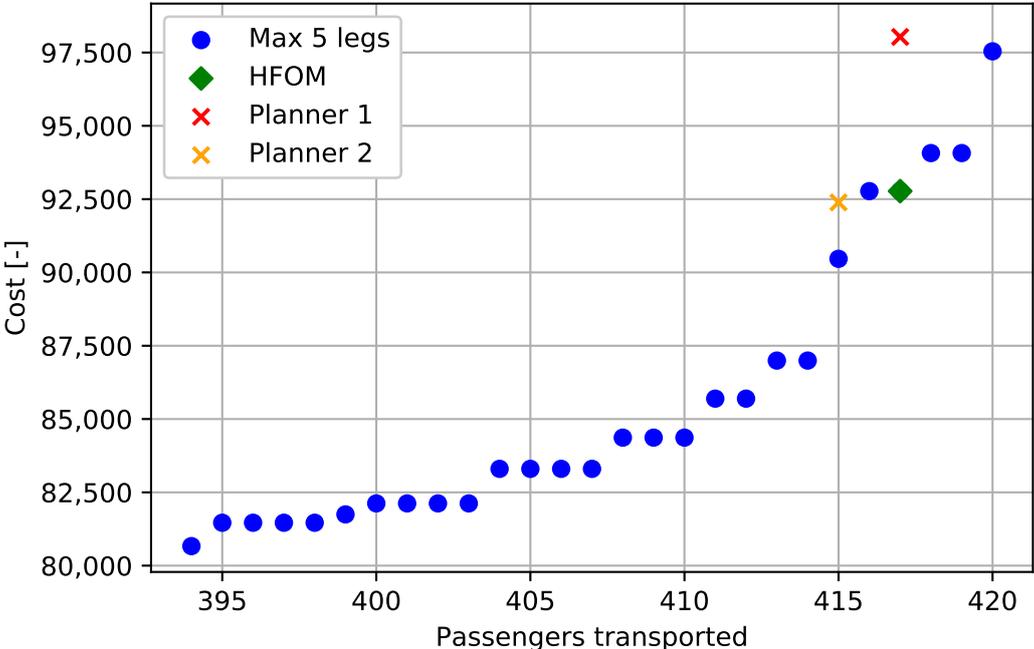
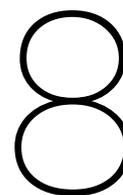


Figure 7.5: Pareto front plot, day 5



Conclusion

Now that the humanitarian flight optimization model has been designed and validated, the results can be discussed and conclusions can be drawn. Firstly, the applicability in the field will be touched upon. The next Section will revolve around the model limitations and the final Section will answer the main research question and give recommendations for future research.

8.1. Applicability

What can be concluded from this comparison between the expert planners and the Humanitarian Flight Optimization Model is that the model is able to create a more cost efficient schedule almost five times faster. From Table 7.18, it can be concluded that the model achieves cost saving between 2.2% to 8.4% in a typical week of operations, while maintaining similar levels in demand satisfaction when compared to manual planners. On top of that, the scheduling time can be reduced by approximately 77%. These savings are significant considering the high cost of aviation operations in comparison to others modes of transport like road transport and shipping. By using a Pareto front the flight planner can make better judgements on what decisions to make when choosing a schedule for a given day, whereas the planners make just one schedule per day and they can only use their own experience in making a 'perfect' schedule. This schedule might be good, but a lack of different schedules to compare it with makes it impossible to know how well it actually performs.

In short the **Flight Schedule Optimization Model** is capable of:

- Optimizing effectiveness of the flight schedule by improving demand satisfaction
- Optimizing the cost efficiency of the flight schedule
- Providing tasking officers with a decision support tool
- Giving insight in network effects and route dependencies

The results of this model are **twofold**:

1. Improved flight schedule for optimised Demand Satisfaction and Cost Efficiency
The algorithm selects the most cost-efficient flight schedule for a given demand satisfaction.
2. Improved strategic decision support
Strategic decision support to help with trading off demand satisfaction and flight routing cost.

8.2. Model Limitations

1. Request splitting
The model is currently unable to split requests. Therefore it is incapable of deciding whether it is more efficient to fly a single person or not. If, for example, a request Juba-Yida (Cessna 208 (10 capacity) and Dornier 228 (15 capacity) are able to land) consists of 11 people, it may be more efficient to 'spill' a single person and move that person to a later date.

2. Sub-problem splitting

The model is unable to handle the full amount of requests that come in daily. The model divides the main problem into sub problems as a solution to this obstacle. Due to this subdivision, a possible global solution might be lost.

8.3. Research Question and Recommendations

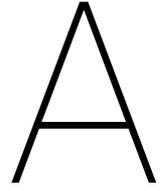
Now that the model has been finalized and the limitations have been identified, the main research question will be answered that has been defined in Chapter 2.

Can a daily flight scheduling model be developed that improves the efficiency and effectiveness of flight routing and scheduling in a humanitarian setting?

Yes. A flight scheduling model has been built that is indeed capable of improving the efficiency in terms of cost for a given demand satisfaction as was shown in Section 7.2. The validation with two expert planners provided the necessary proof that the daily flight scheduling can indeed be improved by using the Humanitarian Flight Optimization Model.

The **recommendations** are as follows:

1. The Humanitarian Flight Optimization Model shows promising results in regard to making cost efficient flight schedules. The five schedules that were obtained were all either better in terms of routing cost, or very close to the solutions found by the planners. However, more testing needs to be done in the field to ensure that the heuristic used will be able to cope with all the different requests that are present.
2. The heuristic divides the main problem into sub-problems in order to ensure that the exact algorithm will be able to find a result. Other algorithms should be researched that might be able to handle the full-scale problem in a timely manner.
3. Weekly schedule development for long-standing missions like South Sudan based on historical demand.
4. More research is required on a fleet planning assistance for long-standing missions by simulating fleet type changes.
5. Training FTOs in balancing efficiency and effectiveness.
6. More research needs to be done on the effect of pushing over demand to next flight in the week.
7. The monthly utilization of aircraft needs to be optimised further. At this moment the minimum guaranteed hours (MGH) required for the contracts are not being looked at. For optimal efficiency the daily utilization needs to be linked to the monthly utilization.



Appendix A

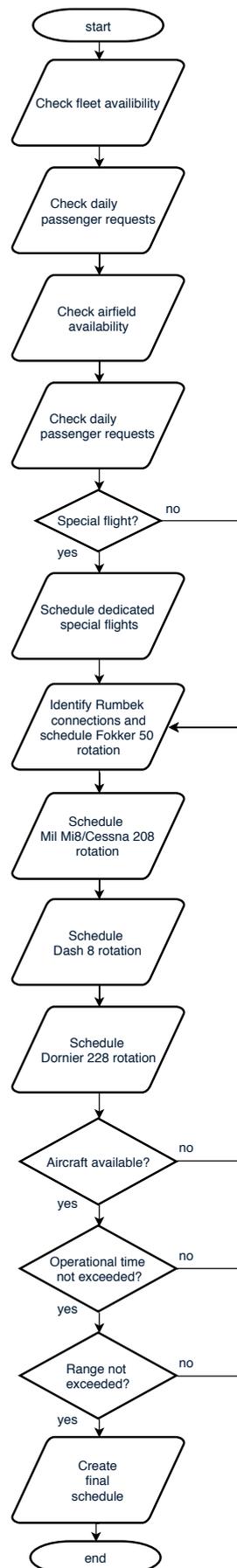


Figure A.1: Flight Tasking Officer decision flowchart

B

Appendix B

	Agek (A)	Aluong Th Akobo (A)	Alek (A)	Awel (A)	Bor (A)	Dorein (A)	Ganyiel (H)	Gum (A)	Haat (H)	Jiech (H)	Juba (A)	Kapoeta (K)	Koch (H)	Kuach (H)	Kurwai (H)	Labrab (A)	Lankien (L)	Leer (H)	Maban (A)	Mabior (A)	Malakal (L)	
	0	106,6297	278,9014	48,87509	79,79114	261,1578	326,7014	162,3378	266,5502	31,16603	162,6142	323,1246	405,0633	95,0335	89,9382	152,7465	358,1341	212,3321	110,295	307,9541	213,4001	181,7893
Aluong The	106,6297	0	207,9363	148,0564	186,2752	239,198	272,2148	153,8052	178,2715	108,2515	314,9408	363,8022	82,84993	65,25656	67,52878	295,1242	136,4336	99,54553	205,1909	181,0073	84,58601	133,406
Akobo (A)	48,87509	207,9363	0	293,0334	340,7816	76,52428	152,0871	60,31484	147,5626	116,4636	193,5009	183,2904	185,487	189,0357	140,4129	71,50657	174,8095	139,3075	102,1102	133,406	133,406	133,406
Alek (A)	79,79114	148,0564	340,7816	0	47,74821	253,1081	397,7153	337,1009	147,5456	147,5456	306,5054	398,1827	109,4453	112,2416	181,7019	365,6216	232,295	118,2244	340,8172	213,5195	213,5195	213,5195
Awel (A)	148,0564	293,0334	47,74821	337,1009	0	296,9457	377,1153	337,1009	147,5456	147,5456	306,5054	398,1827	109,4453	112,2416	181,7019	365,6216	232,295	118,2244	340,8172	213,5195	213,5195	213,5195
Bor (A)	186,2752	340,7816	47,74821	147,5456	296,9457	0	102,3966	98,82651	177,4428	133,239	74,17006	145,7078	145,7078	187,3782	142,9003	142,9003	260,5598	384,9179	259,999	258,0085	201,4835	201,4835
Dorein (A)	272,2148	76,52428	152,0871	397,7153	102,3966	102,3966	0	175,2485	197,4254	168,6008	141,8129	106,7831	321,823	204,0519	205,894	40,51614	139,6862	216,7627	208,629	117,9371	205,5402	205,5402
Ganyiel (H)	326,7014	162,3378	152,0871	397,7153	197,4254	197,4254	175,2485	0	173,571	67,28961	165,7166	243,2935	77,64995	93,63547	117,6869	212,0763	115,8643	115,8643	48,46647	248,2549	57,37705	146,4822
Gum (A)	266,5502	178,2715	60,31484	293,0334	337,1009	177,4428	135,7129	173,571	0	143,87	115,9156	249,2022	242,3759	181,1074	178,8895	115,0219	139,0763	60,16183	176,3173	83,12849	137,7823	94,68901
Haat (H)	131,6603	89,24129	147,5456	147,5456	143,87	150,6325	195,0312	150,6325	143,87	0	31,1021	225,7058	284,8283	38,28083	53,78811	127,0653	53,78811	127,0653	33,3731	204,4734	93,51153	86,54164
Jiech (H)	162,6142	108,2515	116,4636	177,9994	225,6228	133,239	168,6008	69,84673	115,9156	31,1021	0	211,2534	260,0322	69,21297	73,07621	52,70218	196,8164	55,88701	182,6477	74,34485	77,33559	77,33559
Juba (A)	323,1246	314,9408	193,5009	306,0547	345,1336	79,17006	141,8129	165,7166	249,2022	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 (A)	367,8022	183,2904	398,1827	442,4728	145,7078	106,7831	243,2935	242,3759	284,8283	260,0322	119,2228	119,2228	0	313,8092	325,5502	304,3203	114,2278	242,1968	296,1348	312,2128	192,9544	308,6864
Koch (H)	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 (H)	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	132,8835	103,5531
Kurwai (H)	152,7465	67,52878	140,4129	181,7019	226,8076	184,8841	205,894	117,6869	115,0219	52,70218	263,44	304,3203	78,89783	70,1244	0	227,6895	68,90718	83,77764	159,4981	126,1201	34,27794	34,27794
Labrab (A)	358,1341	295,1242	88,31348	365,6216	412,88	142,9013	40,51614	212,0763	139,0763	27,0245	196,8164	176,6284	114,2278	263,1468	269,6629	127,6895	0	159,0379	249,3999	200,2996	154,7915	221,6524
Lankien (H)	212,3321	136,4336	71,50657	232,295	279,6179	142,4093	139,6862	115,8645	60,16183	84,88228	55,88701	220,6881	242,1968	122,8507	122,8241	68,90718	159,0379	0	116,6365	132,7318	90,81738	66,49042
Leer (H)	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	83,77764	249,3999	116,6365	0	237,773	103,4975	117,7299	117,7299	
Maban (A)	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	200,2996	154,7915	221,6524	221,6524
Mabior (A)	181,0073	181,0073	102,1102	121,35195	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 (A)	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
Mankien (A)	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	323,3363	323,3709	179,5389	75,4379	281,7458	178,7255	154,7836
Menime (H)	134,328	85,308	144,5736	152,2915	199,5837	153,9345	196,8549	73,54387	138,3604	7,422981	229,8304	286,1335	42,86112	46,42744	225,4723	80,03457	40,17321	197,6092	96,09067	79,12182	79,12182	79,12182
Mingkam (A)	264,3148	246,0647	136,6716	254,3922	297,4914	10,30093	109,5042	187,7417	157,1253	141,0641	70,39952	145,0779	178,3355	49,27559	193,0894	149,8759	152,1918	159,4509	270,8597	66,97759	210,5325	210,5325
Mogok (A)	174,4773	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
Motor (A)	217,9894	150,581	60,97367	233,647	281,3362	124,3203	122,1953	104,4699	69,37096	86,96428	55,91608	199,3599	222,6727	125,1288	128,088	83,74477	145,0955	21,45708	115,7835	148,1271	71,77856	86,63371
Nyal (H)	139,1165	134,1256	163,9148	183,3347	172,1814	194,3844	23,52803	177,4579	91,57465	105,4483	99,42627	22,21371	231,1178	273,4403	75,69473	74,27327	32,50159	202,2041	48,65054	72,75672	165,5263	93,78663
Old Fangak	137,1407	64,07752	148,0401	163,6297	209,3519	177,895	208,832	102,9315	129,4098	56,63899	44,95734	255,3611	303,7431	59,39419	51,49354	19,54207	223,5181	77,54628	64,62062	178,3874	119,082	53,81931
Pagok (A)	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 (H)	163,6823	94,96552	117,4213	184,8227	231,8319	152,4395	177,4579	91,57465	105,4483	99,42627	22,21371	231,1178	273,4403	75,69473	74,27327	32,50159	202,2041	48,65054	72,75672	165,5263	93,78663	55,15338
Paloich (P)	210,4811	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	122,2564	200,9233	194,5411	104,6138	187,1337
Piber (A)	352,5116	281,3476	74,29359	364,366	412,0331	159,9965	61,95911	215,9554	114,8177	208,8882	189,8981	208,5505	147,1704	258,388	262,8295	173,3663	32,94724	145,2665	246,4723	169,3483	160,1524	203,5
Pochalla (A)	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	232,3338	173,4704	306,8681	191,9465	255,5692	113,7554	281,3429	142,6495
Renk (A)	210,4811	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	122,2564	200,9233	194,5411	104,6138	187,1337
Rubkosa (A)	352,5116	281,3476	74,29359	364,366	412,0331	159,9965	61,95911	215,9554	114,8177	208,8882	189,8981	208,5505	147,1704	258,388	262,8295	173,3663	32,94724	145,2665	246,4723	169,3483	160,1524	203,5
Rumbek (A)	165,2453	191,8392	206,8133	142,5895	181,5435	121,3208	126,2942	59,54371	232,7546	116,7823	127,4599	163,8282	263,9799	109,0012	126,652	170,3798	202,2041	48,65054	72,75672	165,5263	93,78663	55,15338
Tort (A)	379,8103	359,4401	203,2995	366,4403	407,0741	121,3478	133,9078	218,5254	263,4704	271,5905	251,9199	64,68725	64,30852	294,5794	308,7081	301,5757	228,1756	275,7445	341,043	178,4589	312,9404	312,9404

Mankeîn (Menime (Mingklam: Mogok (A) Motot (A) Nyal (H) Old Fangas Paqak (A) Paqil (H) Paloch (P) Pibor (A) Pochalla (P) Rent (A) Rubkona (Rumbek (T) Torit (A) Waat (A) Wai (H) Wau (A) Yambio (A) Yei (A) Yida (A)	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	233.0908	173.8323	104.5737	287.9093	338.7518	98.95271	Agok (A)	
	89.31861	85.308	246.0647	111.172	150.581	134.1256	64.0752	239.3661	94.96552	135.3915	254.0769	180.0117	48.63825	359.4401	153.7466	118.9605	191.9091	342.2885	350.9891	15.63019	Ajuong Th			
	244.8898	144.5726	136.6716	106.4487	60.97367	163.9148	148.0041	73.60555	117.4212	167.0364	59.31318	74.29359	321.7535	208.9133	208.8133	309.2995	56.18518	105.6146	299.8106	334.7609	257.9067	223.5607	Akobo (A)	
	59.97028	152.2915	254.3922	189.2923	233.647	136.641	163.6297	348.8589	184.8237	280.3297	316.1977	364.366	321.7597	104.4586	147.5895	366.4404	239.3036	187.997	57.33096	246.6062	312.4077	143.1224	Alek (A)	
	103.3271	199.5837	297.4914	236.8379	281.3362	183.3347	209.3519	396.1177	321.8319	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	Aweil (A)	
	227.5364	156.9845	103.30093	134.2024	121.3203	122.1814	177.895	198.7287	152.4395	264.8294	97.79482	159.9991	334.3034	215.7667	121.3208	121.3478	124.156	124.5722	235.4279	213.8706	134.6731	282.3714	Bor (A)	
	291.852	193.3504	109.5042	161.9676	122.1953	194.3844	208.832	124.3102	177.4579	243.5169	18.43022	61.95991	327.4386	266.2751	116.2942	133.9078	119.9694	324.841	324.841	210.3332	282.57409	Dorein (A)		
	128.78	138.3604	187.7417	103.9471	69.37096	177.7516	129.4098	61.32453	105.4483	108.9278	119.242	114.8177	171.7519	195.4985	232.7546	263.4304	63.65578	108.854	308.3473	374.2892	311.1719	193.4737	Ganyiel (H)	
	97.36865	7.422991	157.1253	41.84764	86.96428	52.63189	36.50968	20.4439	39.42627	163.2496	180.4609	187.6056	213.9663	173.4704	81.34295	170.3798	301.5757	86.53483	60.32001	210.5193	324.9627	308.1639	Kuach (H)	
	128.441	28.66634	141.0641	11.97301	55.91608	65.32726	44.95734	171.7159	152.2501	151.2096	189.8981	219.7529	97.9276	127.4599	251.9109	61.41189	11.64643	192.4502	279.9256	255.568	122.6177	Jiech (H)		
	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	190.8867	68.62929	327.3386	Juba (A)	
	370.7911	286.1335	145.0779	256.135	222.6727	266.1132	303.7431	223.0273	273.4408	350.2812	124.1423	147.1704	413.9802	353.4902	263.9799	248.97	379.0421	175.4851	382.5021	175.4851	382.5021	Kapoeta (f)		
	56.72458	44.52714	192.7559	82.5872	128.088	71.74136	51.49354	238.5964	74.2722	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	Koch (H)	
	123.5363	46.42744	193.0894	50.80368	83.74477	106.0496	19.54207	176.2664	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	Kurwai (H)	
	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	255.1573	157.2321	141.4882	185.246	362.5282	353.4332	245.1562	310.7008	Labrab (A)	
	179.5389	80.9437	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	322.0802	275.61	152.0563	Lankien (H)	
	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	132.4007	246.4529	253.5855	108.5201	Leer (H)		
	178.7255	96.09027	66.97759	75.62494	71.77856	76.90143	119.082	174.6056	93.57863	211.821	104.6138	160.1524	281.3429	161.0014	105.5941	178.4539	76.27568	65.80946	207.7689	236.7441	186.311	194.566	Mabior (A)	
	154.7836	79.12182	210.5325	70.75296	86.63371	137.5514	53.81981	155.262	55.15338	76.76869	187.1337	203.5	142.6495	111.057	201.7718	312.9404	87.10276	81.07716	244.5644	355.8802	330.1358	99.37902	Malak (A)	
	100.4904	0	160.7588	38.17488	83.6784	59.7876	29.57569	196.5689	33.20866	155.8266	179.6209	218.3044	219.8971	69.43698	124.9337	274.5049	88.86581	40.23562	171.9878	278.6707	268.7403	98.51503	Mankeîn (f)	
	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	Mogok (A)	
	184.1148	83.6784	131.1975	45.51225	0	110.6328	88.42275	118.5151	57.18578	143.5653	103.8756	135.0539	212.9918	150.9853	163.4471	226.9711	6.062246	45.86634	306.2301	255.0031	166.062	Motot (A)		
	106.4496	29.57569	185.4524	47.64364	88.42275	89.12477	0	89.12477	228.9554	84.75002	129.2183	180.0825	231.3187	279.4124	180.92973	64.35346	242.0534	116.695	69.47841	138.9124	218.9174	217.7362	143.9493	Nyal (H)
	295.2461	196.5689	208.6289	160.0576	118.5151	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	Pagak (A)	
	130.9962	33.20866	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	178.3197	366.0275	141.1825	153.8704	316.7578	431.9814	397.7419	145.6002	Paloch (P)
	275.5966	179.6209	106.0825	144.2264	103.8756	180.0825	190.8527	113.8129	159.4408	226.2783	278.3197	286.0199	278.0818	249.0558	206.0837	146.5586	101.5478	139.5639	312.3353	311.3766	215.0323	269.4401	Pibor (A)	
	318.2294	218.3044	168.2136	180.3392	135.0539	231.3187	222.2727	77.60593	191.7144	221.5752	62.20165	0	62.20165	280.9409	249.0558	206.0837	146.5586	101.5478	139.5639	312.3353	311.3766	215.0323	269.4401	Pochalla (f)
	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	222.2932	370.5306	498.197	467.3203	185.5944	185.5944	Rent (A)	
	44.50754	69.43698	221.1153	106.0975	150.9853	88.92973	66.25446	256.3445	94.6761	175.9542	249.0558	286.0199	226.5304	0	149.5162	337.0951	155.5616	109.5643	144.1052	296.3198	315.9257	51.21292	Rubkona (f)	
	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	Rumbek (f)	
	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	0	227.674	242.1543	339.1555	248.5602	171.9086	373.0108	373.0108	Torit (A)	
	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	155.6116	169.2911	227.674	0	51.40572	250.6958	311.4432	258.2618	169.289	Waat (A)	
	139.5056	40.23562	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	0	200.2755	279.7122	249.5524	133.538	133.538	Wat (H)	
	272.2306	278.6707	205.2951	289.1477	306.2301	218.9124	307.7879	408.3636	301.4337	331.9814	311.3766	373.4732	498.197	296.3198	154.6562	248.5602	311.4432	339.1555	250.6958	200.2755	279.7122	249.5524	Wau (A)	
	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	238.2618	249.5524	271.6674	140.3731	0	361.5423	Yei (A)	
	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	Yida (A)	

Figure B.3: Distance matrix, b

C

Appendix C

-Passengers	Cost per region						Total routing cost
	1	2	3	4	5	6	
0	28278	6861	11638	38218	6865	7106	98966
1	28278	6861	11638	38218	6865	3470	95330
2	28278	6861	11638	32796	6865	3470	89909
3	28278	6861	11638	32298	6865	3470	89411
4	28278	6861	11638	32102	6865	3470	89214
5	28278	5775	10781	27061	6865	3470	82230
6	28278	5775	10781	27061	6865	3470	82230
7	28278	5775	10568	27061	6865	3470	82017
8	28278	5775	10568	27061	6865	3470	82017
9	28278	5775	9202	26079	6865	3470	79669

Table C.1: Routing cost for different demand satisfactions with a maximum of 3 legs in a routing for day 1

-Passengers	Cost difference per region					
	1	2	3	4	5	6
1	0	0	0	0	0	3635
2	0	0	0	5421	0	0
3	0	0	0	498	0	0
4	0	0	0	196	0	0
5	0	1086	857	5041	0	0
6	0	0	0	0	0	0
7	0	0	213	0	0	0
8	0	0	0	0	0	0
9	0	0	1367	982	0	0

Table C.2: Difference in routing cost for different demand satisfactions with a maximum of 3 legs in a routing

Change in routing, passengers less transported per region						
-Passengers	1	2	3	4	5	6
1	-	-	-	-	-	Juba-Agok: 1
2	-	-	-	Rubkona-Mankien: 2	-	-
3	-	-	-	Juba-Mankien: 1	-	-
4	-	-	-	Juba-Yida: 1	-	-
5	-	Rumbek-Koch: 5	Juba-Pibor: 5	efficiency	-	-
6	-	-	-	-	-	-
7	-	-	efficiency	-	-	-
8	-	-	-	-	-	-
9	-	-	Bor-Pibor: 8, Juba-Pibor: 1	Juba-Yida: 5	-	-

Table C.3: Difference in routing for different demand satisfactions with a maximum of 3 legs in a routing

Cost per region							
-Passengers	1	2	3	4	5	6	Total routing cost
0	22839	6861	11311	32654	6865	6940	87470
1	22839	6861	11001	32654	6865	3470	83690
2	22839	6861	11001	32570	6865	3470	83606
3	22839	6861	11001	32102	6865	3470	83138
4	22839	6861	11001	27061	6865	3470	78097
5	22839	5775	11001	27061	6865	3470	77011
6	22839	5775	10727	27061	6865	3470	76737
7	22839	5775	10727	27061	6865	3470	76737
8	22839	5775	10727	27061	6865	3470	76737
9	22839	5775	9202	26537	6865	3470	74688

Table C.4: Routing cost for different demand satisfactions with a maximum of 4 legs in a routing for day 1

Cost difference per region							
-Passengers	1	2	3	4	5	6	
1	0	0	310	0	0	3470	
2	0	0	0	85	0	0	
3	0	0	0	468	0	0	
4	0	0	0	5041	0	0	
5	0	1086	0	0	0	0	
6	0	0	273	0	0	0	
7	0	0	0	0	0	0	
8	0	0	0	0	0	0	
9	0	0	1526	524	0	0	

Table C.5: Difference in routing cost for different demand satisfactions with a maximum of 4 legs in a routing

Change in routing, passengers less transported per region						
-Passengers	1	2	3	4	5	6
1	-	-	Juba-Pibor: 1	-	-	Juba-Agok: 1
2	-	-	-	Rubkona-Mankien: 2	-	-
3	-	-	-	Juba-Mankien: 1	-	-
4	-	-	-	Juba-Yida: 1	-	-
5	-	Rumbek-Koch: 5	-	-	-	-
6	-	-	Bor-Juba: 6	-	-	-
7	-	-	-	-	-	-
8	-	-	-	-	-	-
9	-	-	Bor-Pibor: 8, Juba-Pibor: 1	Juba-Yida: 5	-	-

Table C.6: Difference in routing for different demand satisfactions with a maximum of 4 legs in a routing

-Passengers	Cost per region						Total routing cost
	1	2	3	4	5	6	
0	22839	6861	11311	32654	6865	6940	87470
1	22839	6861	11311	32654	6865	3470	84000
2	22839	6861	11311	32570	6865	3470	83916
3	22839	6861	11311	32102	6865	3470	83448
4	22839	6861	11213	27061	6865	3470	78309
5	22839	5775	11213	27061	6865	3470	77223
6	22839	5775	10922	27061	6483	3470	76047
7	19473	5775	10922	27061	6483	3470	73185
8	19473	5775	10771	27061	6483	3470	73034
9	19473	5775	10713	27061	6483	3470	72976

Table C.7: Routing cost for different demand satisfactions with a maximum of 5 legs in a routing for day 1

-Passengers	Cost difference per region						
	1	2	3	4	5	6	
1	0	0	0	0	0	3470	
2	0	0	0	85	0	0	
3	0	0	0	468	0	0	
4	0	0	98	5041	0	0	
5	0	1086	0	0	0	0	
6	5034	0	291	0	381	0	
7	2862	0	0	0	0	0	
8	0	0	150	0	0	0	
9	0	0	59	0	0	0	

Table C.8: Difference in routing cost for different demand satisfactions with a maximum of 5 legs in a routing

-Passengers	Change in routing, passengers less transported per region					
	1	2	3	4	5	6
1	-	-	-	-	-	Juba-Agok: 1
2	-	-	-	Rubkona-Mankien: 2	-	-
3	-	-	-	Juba-Mankien: 1	-	-
4	-	-	Juba-Pibor: 1	Juba-Yida: 1	-	-
5	-	Rumbek-Koch: 5	-	-	-	-
6	efficiency	-	Bor-Juba: 6	-	efficiency	-
7	efficiency	-	-	-	-	-
8	-	-	Juba-Pibor: 5	-	-	-
9	-	-	Juba-Pibor: 1	-	-	-

Table C.9: Difference in routing for different demand satisfactions with a maximum of 5 legs in a routing

D

Appendix D

Day 1

(a) 'F50' region				(b) 'Bor' region				(c) 'Juba North' region			
#	Origin	Destination	Request	#	Origin	Destination	Request	#	Origin	Destination	Request
1	AWL	JUB	14	1	BOR	JUB	6	1	JUB	MAK	18 (35)
2	AWL	RUM	2	2	BOR	PIBR	8	2	JUB	MNK	1
3	JUB	AWL	17	3	JUB	BOR	6	3	JUB	RUB	28
4	JUB	RUM	13	4	JUB	MINGK	15	4	JUB	YIDA	10 (16)
5	JUB	WAU	27	5	JUB	PIBR	10 (16)	5	MAK	JUB	8 (16)
6	RUM	JUB	20	6	MINGK	JUB	7	6	RUB	JUB	15
7	WAU	JUB	20	7	PIBR	JUB	9	7	RUB	MNK	2
8	WAU	RUM	2	8	JUB	PIBR	5 (16)	8	YIDA	JUB	10
9	JUB	RUM (*)	18					9	JUB	MAK	17 (35)
10	RUM (*)	JUB	15					10	JUB	YIDA	5 (16)
								11	MAK	JUB	8 (16)

(d) 'Rumbek 1' region				(e) 'Rumbek 2' region				(f) 'Juba South' region			
#	Origin	Destination	Request	#	Origin	Destination	Request	#	Origin	Destination	Request
1	RUM (JUB*)	KOCH	4	1	AGOK	RUM (JUB*)	9	1	JUB	YAM	12
2	RUM (JUB*)	LER	4	2	RUM (JUB*)	AGOK	10 (11)	2	YAM	JUB	20
3	KOCH	RUM (JUB*)	1								
4	LER	RUM (JUB*)	5								

Table D.1: Pre-processing, dividing and splitting requests, day 1

Table D.1 shows the division of the requests in the different regions. The * in the 'F50' region is to illustrate that the origin or destination is 'virtual' and the total request is the sum of the requests that are transported to and from the 'Rumbek 1' and 'Rumbek 2' regions. The number within parentheses shows the number of the original request and either a passenger has been spilled or a split has been done. The predetermined spill of this day, based on the heuristic, is Juba-Agok: 1, Juba-Pibor: 1, Juba-Yida: 1.

Day 2

(a) 'F50' region				(b) 'Bor' region				(c) 'Juba North' region			
#	Origin	Destination	Request	#	Origin	Destination	Request	#	Origin	Destination	Request
1	WAU	JUB	9	1	MBR	JUB	4	1	PCL	JUB	5
2	RUM	JUB	6	2	JUB	MBR	6	2	JUB	PCL	7
3	JUB	WAU	11	3	JUB	BOR	1	3	JUB	MABAN	27
4	JUB	RUM	18					4	JUB	MAK	7
5	JUB	AWL	12					5	JUB	MOTO	2
6	AWL	WAU	2					6	JUB	PGK	10 (23)
7	AWL	JUB	14					7	MABAN	JUB	16
8	JUB	RUM (*)	50					8	MAK	JUB	2
9	JUB	RUM (*)	3					9	MOTO	JUB	1
10	RUM (*)	JUB	34					10	PGK	JUB	4
								11	JUB	PGK	10 (23)

(d) 'Rumbek 1' region				(e) 'Rumbek 2' region				(f) 'Juba South' region			
#	Origin	Destination	Request	#	Origin	Destination	Request	#	Origin	Destination	Request
1	NYAL	RUM (JUB*)	5	1	OLDFG	RUM (JUB*)	10 (14)	1	KAP	JUB	10 (11)
2	LKEN	RUM (JUB*)	3	2	RUM (JUB*)	OLDFG	4	2	KAP	HSTR	1
3	RUM (JUB*)	NYAL	6	3	RUM (JUB*)	MOGOK	7	3	JUB	KAP	10 (13)
4	RUM (JUB*)	LKEN	9	4	RUM (JUB*)	ALEK	9	4	JUB	HSTR	4
5	RUM (JUB*)	KURWA	5	5	RUM (JUB*)	AJUON	5	5	HSTR	JUB	1
6	RUM (JUB*)	JCH	3	6	ALEK	RUM (JUB*)	4	6	KAP	JUB	1 (11)
7	RUM (JUB*)	GANY	5	7	AJUON	RUM (JUB*)	10	7	JUB	KAP	3 (13)
8	JCH	RUM (JUB*)	2								

Table D.3: Pre-processing, dividing and splitting requests, day 2

Table D.3 shows the division of the requests in the different regions. The * in the 'F50' region is to illustrate that the origin or destination is 'virtual' and the total request is the sum of the requests that are transported to and from the 'Rumbek 1' and 'Rumbek 2' regions. The number within parentheses shows the number of the original request and either a passenger has been spilled or a split has been done. The predetermined spill of this day, based on the heuristic, is Juba-Pagak: 3, Old Fangak-Juba: 4.

Day 3

(a) 'F50' region				(b) 'Bor' region				(c) 'Juba North' region			
#	Origin	Destination	Request	#	Origin	Destination	Request	#	Origin	Destination	Request
1	WAU	RUM	4	1	HSPA	JUB	6	1	JUB	MABAN	23
2	WAU	JUB	5	2	HSPA	BOR	5	2	JUB	MAK	13 (25)
3	RUM	JUB	8	3	MINGK	JUB	9	3	JUB	RUB	19
4	JUB	WAU	8	4	JUB	HSPA	9	4	JUB	YIDA	10 (13)
5	JUB	RUM	20	5	JUB	MINGK	17 (22)	5	MABAN	JUB	7
6	JUB	RUM (*)	17	6	JUB	BOR	1	6	MAK	JUB	12 (23)
7	RUM (*)	JUB	16	7	BOR	HSPA	5	7	RUB	JUB	25
								8	YIDA	JUB	10
								9	JUB	MAK	12 (25)
								10	JUB	YIDA	3 (13)
								11	MAK	JUBA	11 (23)
								12	YIDA	JUB	5
(d) 'Rumbek 1' region				(e) 'Rumbek 2' region				(f) 'Juba South' region			
#	Origin	Destination	Request	#	Origin	Destination	Request	#	Origin	Destination	Request
1	LER	RUM (JUB*)	6	1	RUM (JUB*)	AGOK	10 (11)	1	YEI	JUB	3
2	RUM (JUB*)	PAGL	1	2	AGOK	RUM (JUB*)	7	2	JUB	YEI	4
3	RUM (JUB*)	LER	5								
4	RUM (JUB*)	GANY	1								
5	HAAT	RUM (JUB*)	2								
6	GANY	RUM (JUB*)	1								

Table D.5: Pre-processing, dividing and splitting requests, day 3

Table D.5 shows the division of the requests in the different regions. The * in the 'F50' region is to illustrate that the origin or destination is 'virtual' and the total request is the sum of the requests that are transported to and from the 'Rumbek 1' and 'Rumbek 2' regions. The number within parentheses shows the number of the original request and either a passenger has been spilled or a split has been done. The predetermined spill of this day, based on the heuristic, is Juba-Agok: 1, Juba-Mingkaman: 5.

Day 4

(a) 'F50' region				(b) 'Bor' region				(c) 'Juba North' region			
#	Origin	Destination	Request	#	Origin	Destination	Request	#	Origin	Destination	Request
1	WAU	RUM	1	1	PIBR	JUB	2	1	HSAK	JUB	6
2	WAU	JUB	7	2	PIBR	BOR	3	2	PCL	JUB	4
3	WAU	AWL	1	3	JUB	PIBR	10 (14)	3	HSRN	JUB	1
4	RUM	JUB	8	4	JUB	BOR	10 (15)	4	JUB	HSAK	10 (20)
5	JUB	WAU	24	5	BOR	PIBR	4	5	JUB	PCL	11
6	JUB	RUM	5	6	BOR	JUB	3	6	JUB	HSRN	3
7	JUB	AWL	13	7	JUB	PIBR	4 (14)	7	JUB	MABAN	10
8	AWL	JUB	11	8	JUB	BOR	5 (15)	8	JUB	WAT	6
9	JUB	RUM (*)	24					9	MABAN	JUB	17
10	RUM (*)	JUB	32					10	PGK	JUB	10 (14)
								11	WAT	JUB	2
								12	JUB	HSAK	10 (20)
								13	PGK	JUB	4 (14)
(d) 'Rumbek 1' region				(e) 'Rumbek 2' region				(f) 'Juba South' region			
#	Origin	Destination	Request	#	Origin	Destination	Request	#	Origin	Destination	Request
1	NYAL	RUM (JUB*)	2	1	OLDFG	RUM (JUB*)	4	1	YAM	JUB	10
2	NYAL	GANY	1	2	RUM (JUB*)	ALEK	7	2	KAP	JUB	10 (11)
3	LKEN	RUM (JUB*)	10	3	RUM (JUB*)	AJUON	10	3	JUB	YAM	12
4	RUM (JUB*)	LKEN	5	4	ALEK	RUM (JUB*)	3	4	JUB	KAP	5
5	RUM (JUB*)	GANY	2	5	AJUON	RUM (JUB*)	9	5	HSTR	KAP	1
6	GANY	RUM (JUB*)	4					6	HSTR	JUB	4
								7	KAP	JUB	1 (11)

Table D.7: Pre-processing, dividing and splitting requests, day 4

Table D.7 shows the division of the requests in the different regions. The * in the 'F50' region is to illustrate that the origin or destination is 'virtual' and the total request is the sum of the requests that are transported to and from the 'Rumbek 1' and 'Rumbek 2' regions. The number within parentheses shows the number of the original request and either a passenger has been spilled or a split has been done.

Day 5

(a) 'F50' region			(b) 'Bor' region			(c) 'Juba North' region					
#	Origin	Destination	Request	#	Origin	Destination	Request	#	Origin	Destination	Request
1	WAU	JUB	16	1	MINGK	JUB	15	1	JUB	MABAN	22
2	RUM	WAU	2	2	MINGK	BOR	4	2	JUB	MAK	11 (21)
3	RUM	JUB	29	3	JUB	MINGK	6	3	JUB	RUB	16
4	JUB	WAU	27	4	JUB	BOR	10 (11)	4	JUB	YIDA	9
5	JUB	RUM	22	5	BOR	JUB	9	5	MABAN	JUB	34
6	JUB	AWL	20	6	JUB	BOR	1 (11)	6	MAK	JUB	14 (16)
7	AWL	JUB	12					7	RUB	JUB	23
8	JUB	RUM (*)	31					8	YIDA	JUB	10
9	RUM (*)	JUB	25					9	JUB	MAK	10 (21)
								10	YIDA	JUB	4
(d) 'Rumbek 1' region			(e) 'Rumbek 2' region			(f) 'Juba South' region					
#	Origin	Destination	Request	#	Origin	Destination	Request	#	Origin	Destination	Request
1	LER	RUM (JUB*)	7	1	RUM (JUB*)	AGOK	10	1	YAM	JUB	16
2	KOCH	RUM (JUB*)	8	2	AGOK	RUM (JUB*)	10 (11)	2	JUB	YAM	19
3	RUM (JUB*)	WAI	5								
4	RUM (JUB*)	MENIM	2								
5	RUM (JUB*)	LER	6								
6	RUM (JUB*)	KUACH	3								
7	RUM (JUB*)	KOCH	5								

Table D.9: Pre-processing, dividing and splitting requests, day 5

Table D.9 shows the division of the requests in the different regions. The * in the 'F50' region is to illustrate that the origin or destination is 'virtual' and the total request is the sum of the requests that are transported to and from the 'Rumbek 1' and 'Rumbek 2' regions. The number within parentheses shows the number of the original request and either a passenger has been spilled or a split has been done. The predetermined spill of this day, based on the heuristic, is Agok-Juba: 1, Malakal-Juba: 2.

E

Appendix E

Day 2

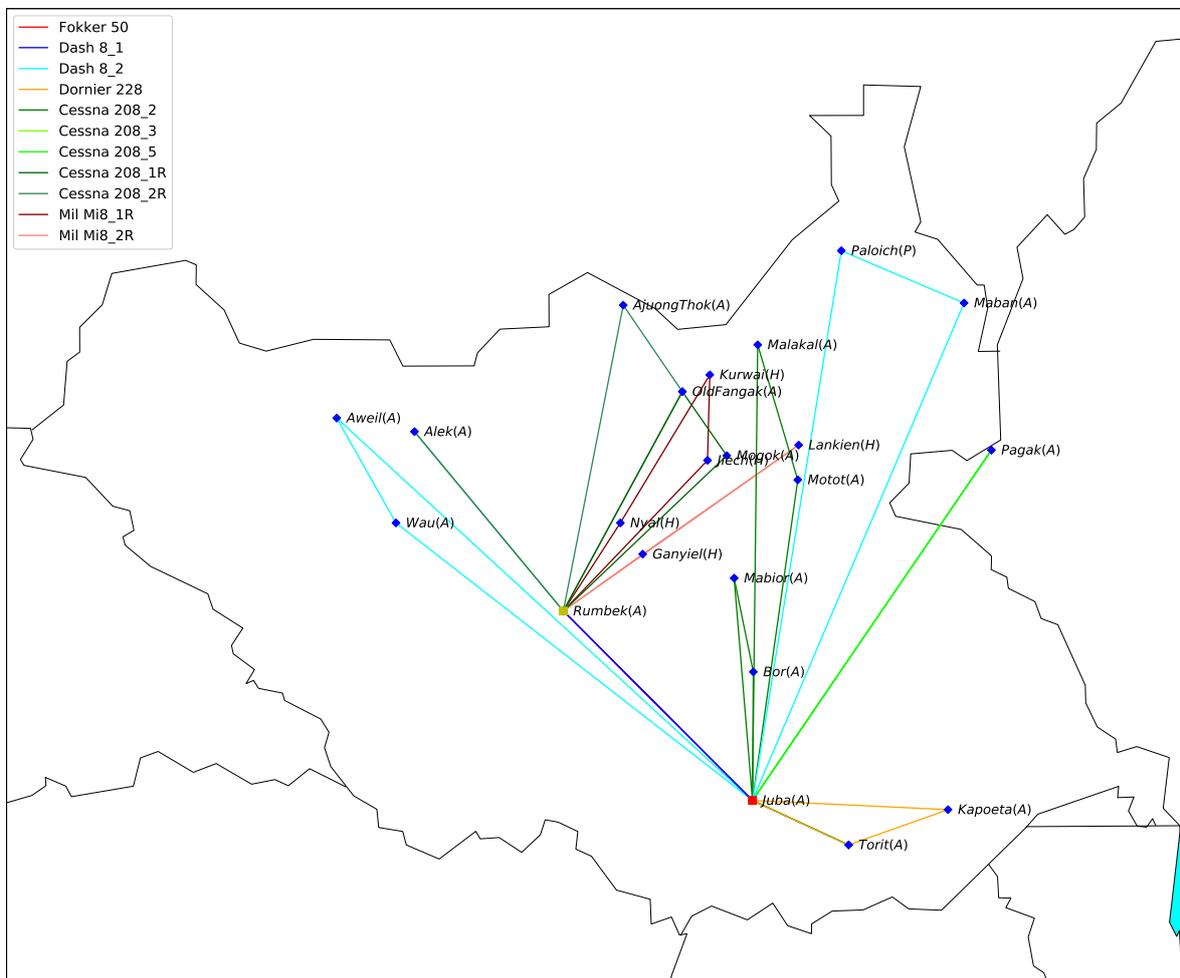


Figure E.1: Results of day 2, objective 91918, pax 291/298

	Origin	Destination	Departure time	Arrival time
Fokker 50	Juba	Rumbek	07:15	07:57
	Rumbek	Juba	14:01	14:44
Dash 8_1	Juba	Rumbek	07:15	08:04
	Rumbek	Juba	08:24	09:13
Dash 8_2	Juba	Aweil	07:15	08:58
	Aweil	Wau	09:18	09:40
	Wau	Juba	10:00	11:23
	Juba	Maban	11:43	13:23
	Maban	Paloich	13:43	14:07
	Paloich	Juba	14:27	16:10
Dornier 228	Juba	Kapoeta	07:15	07:47
	Kapoeta	Torit	08:07	08:25
	Torit	Juba	08:45	09:02
Cessna 208_1	-	-	-	-
Cessna 208_2	Juba	Bor	07:15	07:41
	Bor	Mabior	08:01	08:34
	Mabior	Juba	08:54	09:26
	Juba	Motot	09:46	10:53
	Motot	Malakal	11:13	11:42
	Malakal	Juba	12:02	13:35
	Juba	Torit	13:55	14:17
	Torit	Juba	14:37	14:58
Cessna 208_3	Juba	Pagak	07:15	08:41
	Pagak	Juba	09:01	10:28
Cessna 208_4	-	-	-	-
Cessna 208_5	Juba	Pagak	07:15	08:41
	Pagak	Juba	09:01	10:28
Mil Mi8_1	-	-	-	-
Mil Mi8_2	-	-	-	-
Cessna 208_1R	Rumbek	Mogok	08:30	09:15
	Mogok	Old Fangak	09:35	09:51
	Old Fangak	Rumbek	10:11	11:02
Cessna 208_2R	Rumbek	Old Fangak	08:30	09:21
	Old Fangak	Ajuong Thok	09:41	10:02
	Ajuong Thok	Rumbek	10:22	11:26
	Rumbek	Alek	11:46	12:33
	Alek	Rumbek	12:53	13:41
Mil Mi8_1R	Rumbek	Jiech	08:30	09:33
	Jiech	Kurwai	09:53	10:20
	Kurwai	Nyal	10:40	11:33
Mil Mi8_2R	Nyal	Rumbek	11:53	12:25
	Rumbek	Lankien	08:30	09:57
	Lankien	Ganyiel	10:17	11:15
	Ganyiel	Rumbek	11:35	12:05

Table E.1: Flight schedule, day 2

Origin	Destination	Request	Transported by	Spill
Ajuong Thok	Juba	10	Cessna 208_2R, Fokker 50	
Alek	Juba	4	Cessna 208_2R, Fokker 50	
Aweil	Juba	14	Dash 8_2	
Aweil	Wau	2	Dash 8_2	
Paloich	Juba	5	Dash 8_2	
Torit	Juba	1	Dornier 228	
Jiech	Juba	2	Mil Mi8_1R, Fokker 50	
Juba	Ajuong Thok	5	Fokker 50, Cessna 208_2R	
Juba	Alek	9	Fokker 50, Cessna 208_2R	
Juba	Aweil	12	Dash 8_2	
Juba	Bor	1	Cessna 208_2	
Juba	Ganyiel	5	Fokker 50, Mil Mi8_2R	
Juba	Paloich	7	Dash 8_2	
Juba	Torit	4	Cessna 208_2	
Juba	Jiech	3	Dash 8_1, Mil Mi8_1R	
Juba	Kapoeta	13	Dornier 228	
Juba	Kurwai	5	Fokker 50, Mil Mi8_1R	
Juba	Lankien	9	Fokker 50, Mil Mi8_2R	
Juba	Maban	27	Dash 8_2	
Juba	Malakal	7	Cessna 208_2	
Juba	Mabior	6	Cessna 208_2	
Juba	Mogok	7	Fokker 50, Cessna 208_1R	
Juba	Motot	2	Cessna 208_2	
Juba	Nyal	6	Fokker 50, Mil Mi8_1R	
Juba	Old Fangak	4	Fokker 50, Cessna 208_1R	
Juba	Pagak	23	Cessna 208_3: 10, Cessna 208_5: 10	3
Juba	Rumbek	18	Dash 8_1	
Juba	Wau	11	Dash 8_2	
Kapoeta	Torit	1	Dornier 228	
Kapoeta	Juba	11	Dornier 228	
Lankien	Juba	3	Mil Mi8_2R, Fokker 50	
Maban	Juba	16	Dash 8_2	
Malakal	Juba	2	Cessna 208_2	
Mabior	Juba	4	Cessna 208_2	
Motot	Juba	1	Cessna 208_2	
Nyal	Juba	5	Mil Mi8_1R, Fokker 50	
Old Fangak	Juba	14	Cessna 208_2R: 10, Fokker 50: 10	4
Pagak	Juba	4	Cessna 208_3	
Rumbek	Juba	6	Dash 8_1	
Wau	Juba	9	Dash 8_2	

Table E.2: Request transport, day 2

Day 3

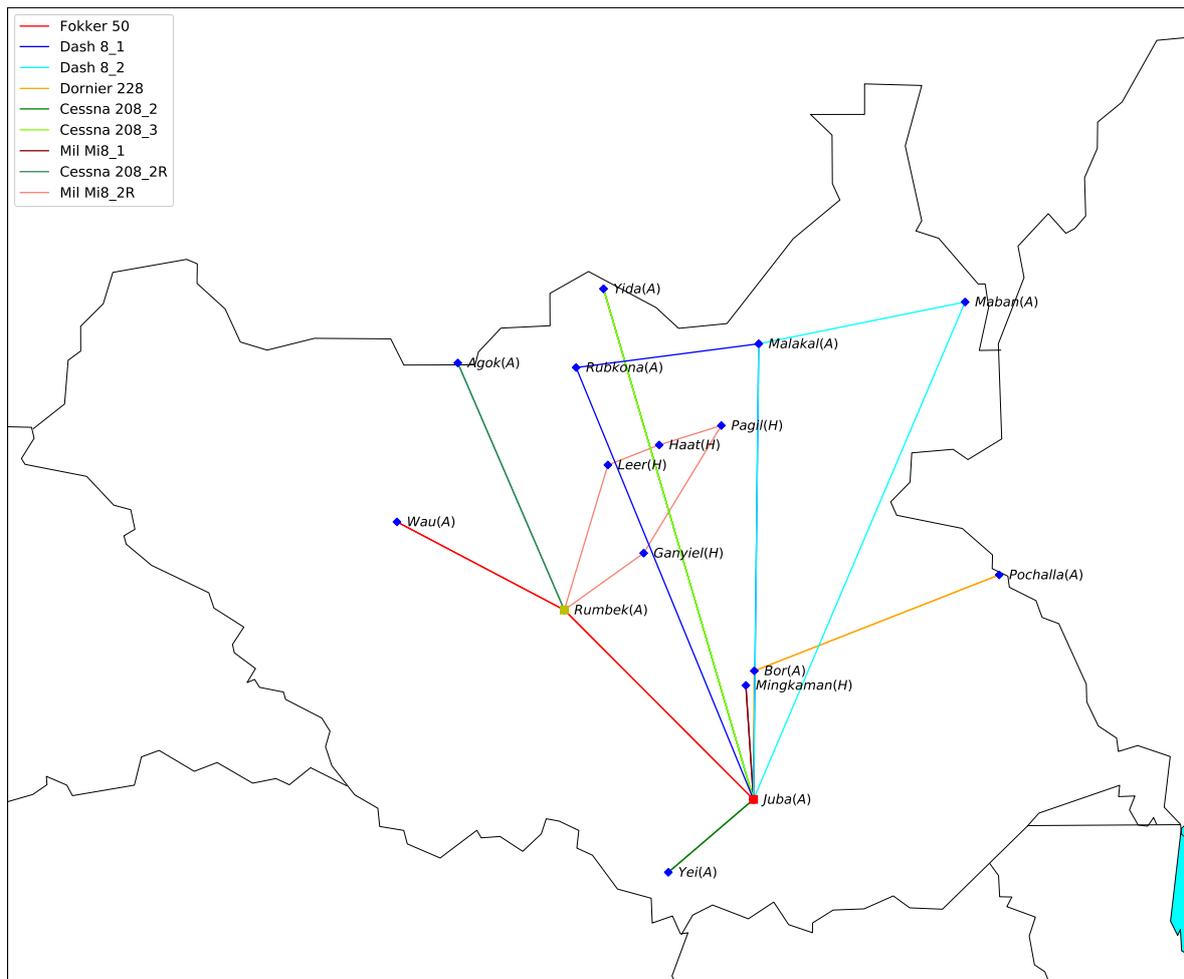


Figure E.2: Results of day 3, objective 72519, pax 287/293

	Origin	Destination	Departure time	Arrival time
Fokker 50	Juba	Rumbek	07:15	07:57
	Rumbek	Wau	08:17	08:47
	Wau	Rumbek	09:07	09:37
	Rumbek	Juba	12:48	13:31
Dash 8_1	Juba	Malakal	07:15	08:39
	Malakal	Rubkona	08:59	09:32
	Rubkona	Juba	09:52	11:18
Dash 8_2	Juba	Maban	07:15	08:54
	Maban	Malakal	09:14	09:52
	Malakal	Juba	10:12	11:37
Dornier 228	Juba	Bor	07:15	07:36
	Bor	Pochalla	07:56	08:40
	Pochalla	Bor	09:00	09:43
	Bor	Juba	10:03	10:25
Cessna 208_1	-	-	-	-
Cessna 208_2	Juba	Yida	07:15	09:04
	Yida	Juba	09:24	11:13
	Juba	Yei	11:33	11:56
	Yei	Juba	12:16	12:39
Cessna 208_3	Juba	Yida	07:15	09:04
	Yida	Juba	09:24	11:13
Cessna 208_4	-	-	-	-
Cessna 208_5	-	-	-	-
Mil Mi8_1	Juba	Mingkaman	07:15	07:50
	Mingkaman	Juba	08:10	08:45
Mil Mi8_2	-	-	-	-
Cessna 208_1R	-	-	-	-
Cessna 208_2R	Rumbek	Agok	08:30	09:25
	Agok	Rumbek	09:45	10:40
*Mil Mi8_1R	-	-	-	-
Mil Mi8_2R	Rumbek	Leer	08:30	09:16
	Leer	Haat	09:36	09:53
	Haat	Pagil	10:13	10:33
	Pagil	Ganyiel	10:53	11:39
	Ganyiel	Rumbek	11:59	12:28

Table E.3: Flight schedule, day 3

Origin	Destination	Request	Transported by	Spill
Agok	Juba	7	Cessna 208_2R, Fokker 50	
Bor	Pochalla	5	Dornier 228	
Ganyiel	Juba	1	Mil Mi8_2R, Fokker 50	
Haat	Juba	2	Mil Mi8_2R, Fokker 50	
Juba	Agok	11	Fokker 50: 10, Cessna 208_2R: 10	1
Juba	Bor	1	Dornier 228	
Juba	Ganyiel	1	Fokker 50, Mil Mi8_2R	
Juba	Leer	5	Fokker 50, Mil Mi8_2R	
Juba	Maban	23	Dash 8_2	
Juba	Malakal	25	Dash 8_1: 13, Dash 8_2: 12	
Juba	Mingkaman	22	Mil Mi8_1: 17	5
Juba	Pagil	1	Fokker 50, Mil Mi8_2R	
Juba	Pochalla	9	Dornier 228	
Juba	Rubkona	19	Dash 8_1	
Juba	Rumbek	20	Fokker 50	
Juba	Wau	8	Fokker 50	
Juba	Yei	4	Cessna 208_2	
Juba	Yida	13	Cessna 208_2: 10, Cessna 208_3: 3	
Leer	Juba	6	Mil Mi8_2R, Fokker 50	
Maban	Juba	7	Dash 8_2	
Malakal	Juba	23	Dash 8_1: 12, Dash 8_2: 11	
Mingkaman	Juba	9	Mil Mi8_1	
Pochalla	Bor	5	Dornier 228	
Pochalla	Juba	6	Dornier 228	
Rubkona	Juba	25	Dash 8_1	
Rumbek	Juba	8	Fokker 50	
Wau	Juba	5	Fokker 50	
Wau	Rumbek	4	Fokker 50	
Yei	Juba	3	Cessna 208_2	
Yida	Juba	15	Cessna 208_2: 5, Cessna 208_3: 10	

Table E.4: Request transport, day 3

Day 4

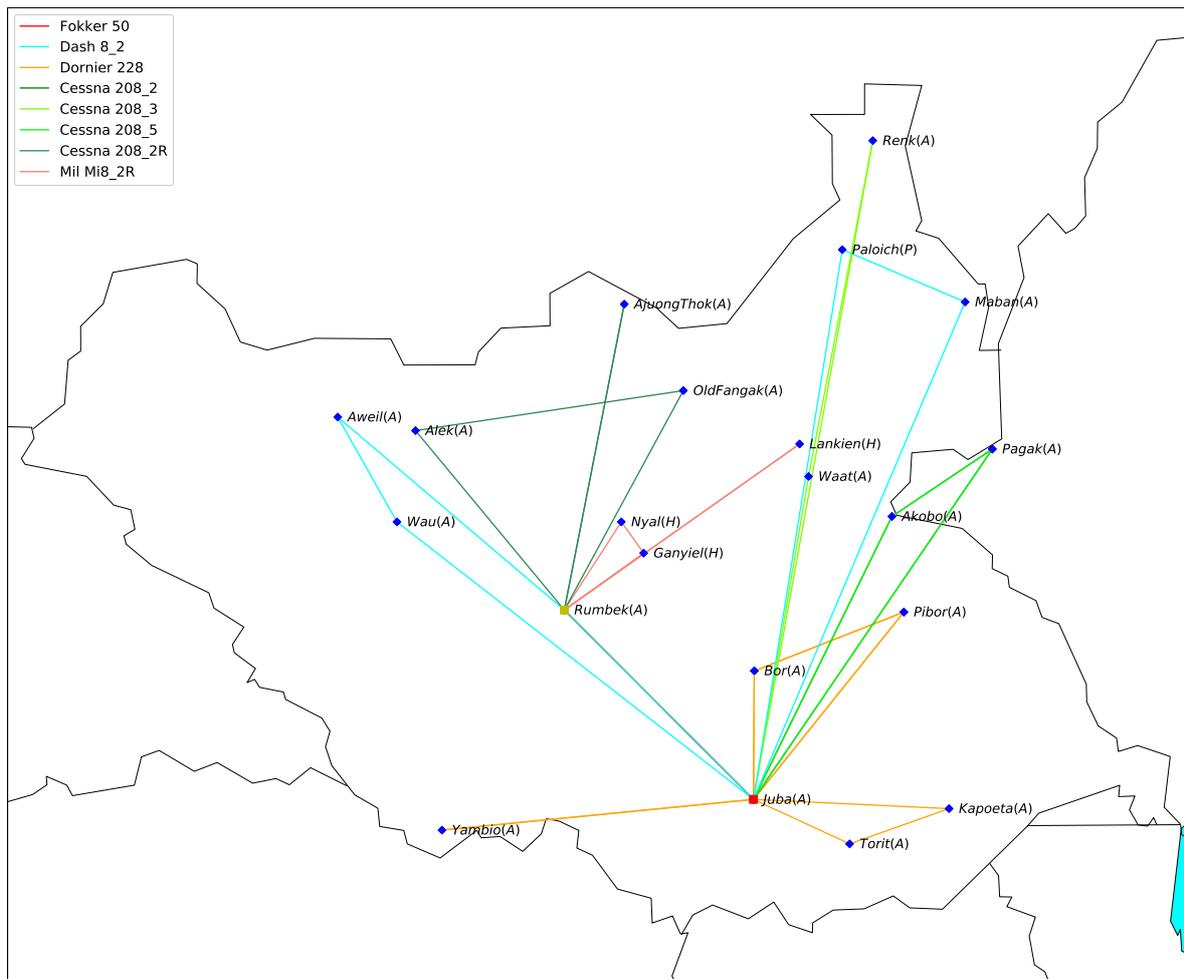


Figure E.3: Results of day 4, objective 87414, pax 305/305

	Origin	Destination	Departure time	Arrival time
Fokker 50	Juba	Rumbek	07:15	07:57
	Rumbek	Juba	14:51	15:34
Dash 8_1	-	-	-	-
Dash 8_2	Juba	Wau	07:15	08:37
	Wau	Aweil	08:57	09:19
	Aweil	Rumbek	09:39	10:34
	Rumbek	Juba	10:54	11:43
	Juba	Paloich	12:03	13:46
	Paloich	Maban	14:06	14:30
	Maban	Juba	14:50	16:30
Dornier 228	Juba	Bor	07:15	07:36
	Bor	Pibor	07:56	08:23
	Pibor	Juba	08:43	09:23
	Juba	Pibor	09:43	10:23
	Pibor	Bor	10:43	11:10
	Bor	Juba	11:30	11:52
	Juba	Yambio	12:12	13:04
	Yambio	Juba	13:24	14:16
	Juba	Torit	14:36	14:53
	Torit	Kapoeta	15:13	15:31
Kapoeta	Juba	15:51	16:23	
Cessna 208_1	-	-	-	-
Cessna 208_2	Juba	Akobo	07:15	08:20
	Akobo	Pagak	08:40	09:04
	Pagak	Juba	09:24	10:50
Cessna 208_3	Juba	Renk	07:15	09:32
	Renk	Waat	09:52	11:02
	Waat	Juba	11:22	12:30
Cessna 208_4	-	-	-	-
Cessna 208_5	Juba	Akobo	07:15	08:20
	Akobo	Pagak	08:40	09:04
	Pagak	Juba	09:24	10:50
Mil Mi8_1	-	-	-	-
Mil Mi8_2	-	-	-	-
Cessna 208_1R	-	-	-	-
Cessna 208_2R	Rumbek	Ajuong Thok	08:30	09:33
	Ajuong Thok	Rumbek	09:53	10:57
	Rumbek	Alek	11:17	12:05
	Alek	Old Fangak	12:25	13:20
	Old Fangak	Rumbek	13:40	14:31
Mil Mi8_1R	-	-	-	-
Mil Mi8_2R	Rumbek	Lankien	08:30	09:57
	Lankien	Rumbek	10:17	11:45
	Rumbek	Nyal	12:05	12:37
	Nyal	Ganyiel	12:57	13:09
	Ganyiel	Rumbek	13:29	13:59

Table E.5: Flight schedule, day 4

Origin	Destination	Request	Transported by	Spill
Ajuong Thok	Juba	9	Cessna 208_2R, Fokker 50	
Alek	Juba	3	Cessna 208_2R, Fokker 50	
Aweil	Juba	11	Dash 8_2	
Bor	Juba	3	Dornier 228	
Bor	Pibor	4	Dornier 228	
Ganyiel	Juba	4	Mil Mi8_2R, Fokker 50	
Akobo	Juba	6	Cessna 208_2	
Paloich	Juba	4	Dash 8_2	
Renk	Juba	1	Cessna 208_5	
Torit	Juba	4	Dornier 228	
Torit	Kapoeta	1	Dornier 228	
Juba	Ajuong Thok	10	Fokker 50, Cessna 208_2R	
Juba	Alek	7	Fokker 50, Cessna 208_2R	
Juba	Aweil	13	Dash 8_2	
Juba	Bor	15	Dornier 228: 15	
Juba	Ganyiel	2	Fokker 50, Mil Mi8_2R	
Juba	Akobo	20	Cessna 208_2: 10, Cessna 208_5: 10	
Juba	Paloich	11	Dash 8_2	
Juba	Renk	3	Cessna 208_5	
Juba	Kapoeta	5	Dornier 228	
Juba	Lankien	5	Fokker 50, Mil Mi8_2R	
Juba	Maban	10	Dash 8_2	
Juba	Pibor	14	Dornier 228	
Juba	Rumbek	5	Fokker 50	
Juba	Waat	6	Cessna 208_5	
Juba	Wau	24	Dash 8_2	
Juba	Yambio	12	Dornier 228	
Kapoeta	Juba	11	Dornier 228	
Lankien	Juba	10	Mil Mi8_2R, Fokker 50	
Maban	Juba	17	Dash 8_2	
Nyal	Ganyiel	1	Mil Mi8_2R	
Nyal	Juba	2	Mil Mi8_2R, Fokker 50	
Old Fangak	Juba	4	Cessna 208_2R, Fokker 50,	
Pagak	Juba	14	Cessna 208_2: 4, Cessna 208_5: 10	
Pibor	Bor	3	Dornier 228	
Pibor	Juba	2	Dornier 228	
Rumbek	Juba	8	Dash 8_2	
Waat	Juba	2	Cessna 208_5	
Wau	Aweil	1	Dash 8_2	
Wau	Juba	7	Dash 8_2	
Wau	Rumbek	1	Dash 8_2	
Yambio	Juba	10	Dornier 228	

Table E.6: Request transport, day 4

Day 5

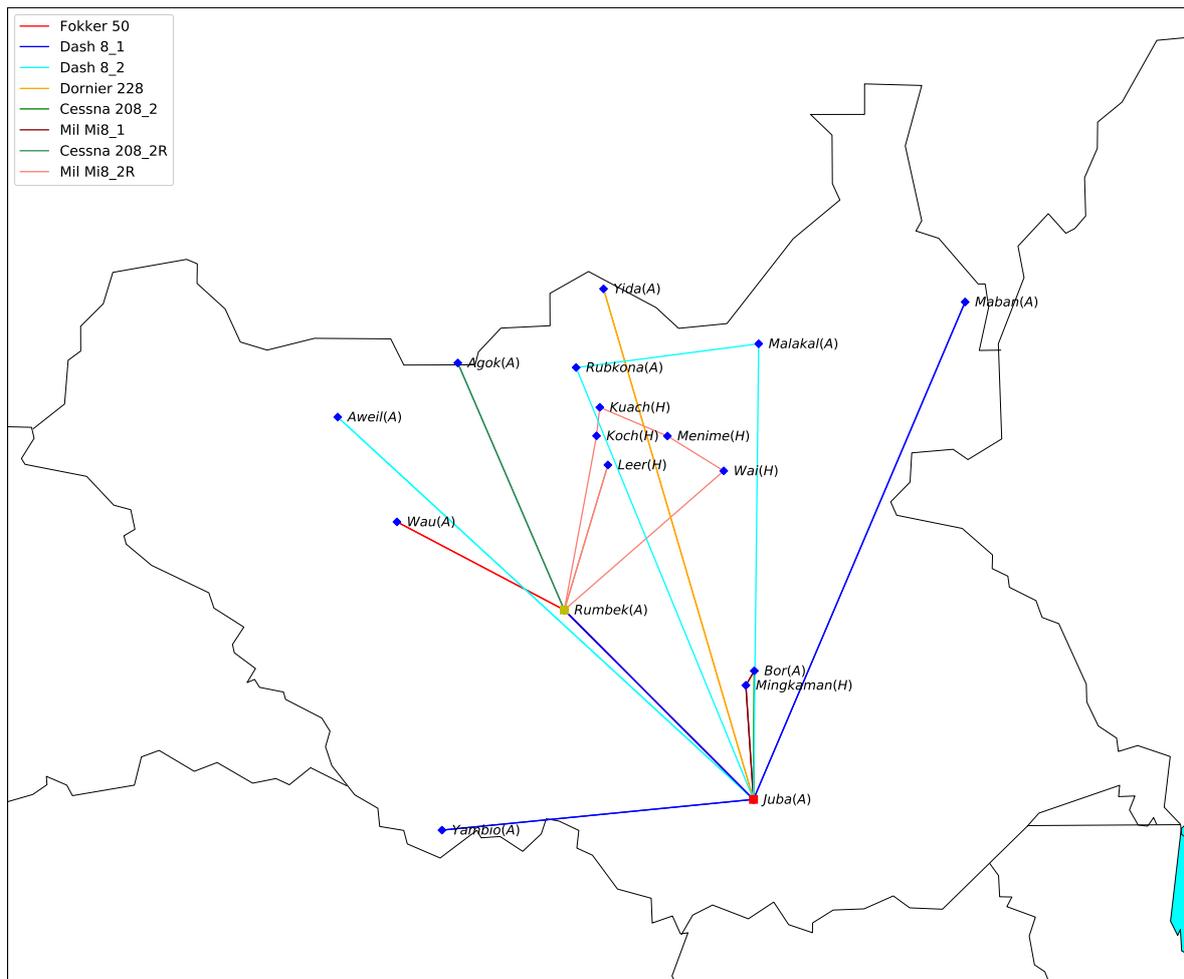


Figure E.4: Results of day 5, objective 92773, pax 417/420

	Origin	Destination	Departure time	Arrival time
Fokker 50	Juba	Rumbek	07:15	07:57
	Rumbek	Wau	08:17	08:48
	Wau	Rumbek	09:08	09:38
	Rumbek	Juba	15:13	15:56
Dash 8_1	Juba	Rumbek	07:15	08:04
	Rumbek	Juba	08:24	09:13
	Juba	Maban	09:33	11:13
	Maban	Juba	11:33	13:12
	Juba	Yambio	13:32	14:30
	Yambio	Juba	14:50	15:47
Dash 8_2	Juba	Aweil	07:15	08:58
	Aweil	Juba	09:18	11:02
	Juba	Malakal	11:22	12:46
	Malakal	Rubkona	13:06	13:40
	Rubkona	Juba	14:00	15:26
Dornier 228	Juba	Yida	07:15	08:44
	Yida	Juba	09:04	10:33
Cessna 208_1	-	-	-	-
Cessna 208_2	Juba	Bor	07:15	07:41
	Bor	Juba	08:01	08:28
Cessna 208_3	-	-	-	-
Cessna 208_4	-	-	-	-
Cessna 208_5	-	-	-	-
Mil Mi8_1	Juba	Mingkaman	07:15	07:50
	Mingkaman	Bor	08:10	08:15
	Bor	Mingkaman	08:35	08:40
	Mingkaman	Juba	09:00	09:35
Mil Mi8_2	-	-	-	-
Cessna 208_1R	-	-	-	-
Cessna 208_2R	Rumbek	Agok	08:30	09:25
	Agok	Rumbek	09:45	10:40
Mil Mi8_1R	-	-	-	-
Mil Mi8_2R	Rumbek	Wai	08:30	09:34
	Wai	Menime	09:54	10:14
	Menime	Kuach	10:34	10:56
	Kuach	Koch	11:16	11:26
	Koch	Rumbek	11:46	12:40
	Rumbek	Leer	13:00	13:47
	Leer	Rumbek	14:07	14:53

Table E.7: Flight schedule, day 5

Origin	Destination	Request	Transported by	Spill
Agok	Juba	11	Cessna 208_2R: 10, Fokker 50: 10	1
Aweil	Juba	12	Dash 8_2	
Bor	Juba	9	Cessna 208_2	
Juba	Agok	10	Dash 8_1, Cessna 208_2R	
Juba	Aweil	20	Dash 8_2	
Juba	Bor	11	Mil Mi8_1	
Juba	Koch	5	Dash 8_1, Mil Mi8_2R	
Juba	Kuach	3	Dash 8_1, Mil Mi8_2R	
Juba	Leer	6	Dash 8_1, Mil Mi8_2R	
Juba	Maban	22	Dash 8_1	
Juba	Malakal	21	Dash 8_2	
Juba	Menime	2	Dash 8_1, Mil Mi8_2R	
Juba	Mingkaman	6	Mil Mi8_1	
Juba	Rubkona	16	Dash 8_2	
Juba	Rumbek	22	Fokker 50	
Juba	Wai	5	Dash 8_1, Mil Mi8_2R	
Juba	Wau	27	Fokker 50	
Juba	Yambio	19	Dash 8_1	
Juba	Yida	9	Dornier 228	
Koch	Juba	8	Cessna 208_2R, Fokker 50	
Leer	Juba	7	Cessna 208_2R, Fokker 50	
Maban	Juba	34	Dash 8_1	
Malakal	Juba	16	Dash 8_2: 14	2
Mingkaman	Bor	4	Mil Mi8_1	
Mingkaman	Juba	15	Mil Mi8_1	
Rubkona	Juba	23	Dash 8_2	
Rumbek	Juba	29	Dash 8_1	
Rumbek	Wau	2	Fokker 50	
Wau	Juba	16	Fokker 50	
Yambio	Juba	16	Dash 8_1	
Yida	Juba	14	Dornier 228	

Table E.8: Request transport, day 5

F

Appendix F

Manual Schedule											Planner: 1				day: 1			
Route	Leg	Origin	Destination	Aircraft	Departure time	Arrival time	Pax on leg	Pax On	Pax off	Pax Connecting: (route, #LegID)	Connecting To: (route, #LegID)	Seat Capacity (-)	Ground Distance (NM)	Block [hours]	Load factor [%]	Flight Cost [Cost Units]		
1	1	Juba (A)	Rumbek (A)	Fokker 50	07:15	08:00	38	38	13	4:4		50	164	0.71	76%	3277		
1	2	Rumbek (A)	Awelil (A)	Fokker 50	08:20	08:41	17	0	17			50	182	0.79	34%	3631		
1	3	Awelil (A)	Rumbek (A)	Fokker 50	09:01	09:20	16	16	2	14		50	182	0.79	32%	3631		
1	4	Rumbek (A)	Juba (A)	Fokker 50	11:00	11:48	40	26	40			50	164	0.71	80%	3277		
2	1	Rumbek (A)	Koch (H)	MI1 Mi8_2R	08:30	09:24	8	4	4			17	109	0.91	47%	3379		
2	2	Koch (H)	Leer (H)	MI1 Mi8_2R	09:44	09:54	5	1	4			17	19	0.16	29%	595		
2	3	Leer (H)	Rumbek (A)	MI1 Mi8_2R	10:14	10:14	6	5	0		6	17	93	0.78	35%	2888		
3	1	Juba (A)	Rubkona (A)	Dornier 228	07:00	08:18	12	12	0			15	287	1.30	80%	3156		
3	2	Rubkona (A)	Agok (A)	Dornier 228	08:38	08:57	14	2	11			15	71	0.32	93%	785		
3	3	Agok (A)	Mankein (A)	Dornier 228	09:17	09:27	12	9	3			15	35	0.16	80%	384		
3	4	Mankein (A)	Juba (A)	Dornier 228	09:47	11:06	9	0	9			15	292	1.33	60%	3208		
3	5	Juba (A)	Yida (A)	Dornier 228	11:26	12:56	15	15	15			15	327	1.49	100%	3601		
3	6	Yida (A)	Juba (A)	Dornier 228	13:16	14:45	10	10	10			15	327	1.49	67%	3601		
4	1	Juba (A)	Pibor (A)	MI1 Mi8_1	07:00	08:13	16	16	16			17	147	1.23	94%	4712		
4	2	Pibor (A)	Juba (A)	MI1 Mi8_1	08:33	09:47	9	9	9			17	147	1.23	53%	4712		
5	1	Juba (A)	Bor (A)	Cessna 208_2	08:00	08:26	6	6	6			10	79	0.44	60%	728		
5	2	Bor (A)	Pibor (A)	Cessna 208_2	08:46	09:18	8	8	8			10	98	0.54	80%	900		
5	3	Pibor (A)	Juba (A)	Cessna 208_2	09:38	10:11	0	0	0			10	98	0.54	0%	900		
5	4	Bor (A)	Juba (A)	Cessna 208_2	10:31	10:57	6	6	6			10	79	0.44	60%	728		
6	1	Juba (A)	Mingkamani (H)	MI1 Mi8_2	08:00	08:35	15	15	15			17	70	0.59	88%	2253		
6	2	Mingkamani (H)	Juba (A)	MI1 Mi8_2	08:55	09:30	7	7	7			17	70	0.59	41%	2253		
7	1	Juba (A)	Rubkona (A)	Dash 8_2	12:15	13:41	28	28	28			37	287	1.43	76%	5165		
7	2	Rubkona (A)	Juba (A)	Dash 8_2	14:01	15:27	15	15	15			37	287	1.43	41%	5165		
8	1	Juba (A)	Yambio (A)	Cessna 208_3	08:00	09:03	10	10	10			10	191	1.06	100%	1812		
8	2	Yambio (A)	Juba (A)	Cessna 208_3	09:23	10:27	10	10	10			10	191	1.06	100%	1812		
8	3	Juba (A)	Yambio (A)	Cessna 208_3	10:47	11:50	2	2	2			10	191	1.06	20%	1812		
8	4	Yambio (A)	Juba (A)	Cessna 208_3	12:10	13:14	10	10	10			10	191	1.06	100%	1812		
9	1	Juba (A)	Malakal (A)	Dash 8_1	07:00	08:24	35	35	35			37	281	1.40	95%	4771		
9	2	Malakal (A)	Juba (A)	Dash 8_1	08:44	10:08	16	16	16			37	281	1.40	43%	4771		
10	1	Juba (A)	Wau (A)	Dash 8_1	11:00	12:22	27	27	27			37	276	1.38	73%	4690		
10	2	Wau (A)	Rumbek (A)	Dash 8_1	12:42	13:17	22	22	2			37	115	0.58	59%	1956		
10	3	Rumbek (A)	Juba (A)	Dash 8_1	13:37	14:26	20	0	20			37	164	0.82	54%	2785		

Figure F.1: Planner 1 results, day 1

Manual schedule														Planner: 1		Day: 2	
Route	Leg	Origin	Destination	Aircraft	Departure time	Arrival time	Pax on leg	Pax On	Pax off	Pax Connecting:	Connecting To: (route, #Leg/D)	Seat Capacity [-]	Ground Distance (NM)	Block [hours]	Load factor [%]	Flight Cost [Cost Units]	
1	1	Juba (A)	Rumbek (A)	Fokker 50	07:00	07:42	50	50	50	5:9:5:5:6:4	(12,3)(13,1)(12,3)(5,1)(13,2)(12,4)	50	164	0.71	100%	3277	
1	2	Rumbek (A)	Aweil (A)	Fokker 50	08:02	08:30	13	0	12			50	182	0.79	26%	3631	
1	3	Aweil (A)	Wau (A)	Fokker 50	09:10	09:29	17	16	3			50	74	0.32	34%	1477	
1	4	Wau (A)	Juba (A)	Fokker 50	09:49	11:01	23	9	23			50	276	1.20	46%	5518	
2	1	Juba (A)	Mogok (A)	Cessna 208_2	07:00	08:10	9	9	7			10	213	1.18	90%	1959	
2	2	Mogok (A)	Motot (A)	Cessna 208_2	08:30	08:46	3	1	2			10	46	0.25	30%	419	
2	3	Motot (A)	Juba (A)	Cessna 208_2	09:06	10:12	1	0	1			10	199	1.11	10%	1834	
3	1	Juba (A)	Kapoeta (A)	Dornier 228	07:00	07:32	15	15	11			15	119	0.54	100%	1311	
3	2	Kapoeta (A)	Torit (A)	Dornier 228	07:52	08:10	5	1	1			15	64	0.29	33%	707	
3	3	Torit (A)	Kapoeta (A)	Dornier 228	08:30	08:47	0	0	0			15	64	0.29	0%	707	
3	4	Kapoeta (A)	Juba (A)	Dornier 228	09:07	09:40	11	11	11			15	119	0.54	73%	1311	
4	1	Juba (A)	Maban (A)	Dash 8_1	07:00	08:39	34	34	27			37	332	1.66	92%	5646	
4	2	Maban (A)	Malakal (A)	Dash 8_1	08:59	09:37	23	16	7			37	127	0.63	62%	2158	
4	3	Malakal (A)	Juba (A)	Dash 8_1	09:57	11:21	18	2	18			37	281	1.40	49%	4771	
5	1	Rumbek (A)	Ajuong Thok (A)	Cessna 208_1R	08:30	09:33	5	5	5			10	192	1.07	50%	2110	
5	2	Ajuong Thok (A)	Rumbek (A)	Cessna 208_1R	09:53	10:57	9	9	0	9 (9,4)		10	192	1.07	90%	2110	
6	1	Juba (A)	Bor (A)	Cessna 208_2	10:30	10:56	7	7	1			10	79	0.44	70%	728	
6	2	Bor (A)	Mabior (A)	Cessna 208_2	11:16	11:36	6	0	6			10	59	0.33	60%	542	
6	3	Mabior (A)	Juba (A)	Cessna 208_2	11:56	12:41	4	4	4			10	137	0.76	40%	1264	
7	1	Juba (A)	Pagak (A)	Cessna 208_3	07:00	08:26	10	10	10			10	260	1.44	100%	2470	
7	2	Pagak (A)	Juba (A)	Cessna 208_3	08:46	10:13	4	4	4			10	260	1.44	40%	2470	
8	1	Juba (A)	Pagak (A)	Dornier 228	07:00	08:10	13	13	13			15	260	1.18	87%	2860	
8	2	Pagak (A)	Juba (A)	Dornier 228	08:30	09:41	0	0	0	(12,9)		15	260	1.18	0%	2860	
9	1	Juba (A)	Rumbek (A)	Dash 8_1	11:45	12:34	37	37	18			37	164	0.82	100%	2785	
9	2	Rumbek (A)	Wau (A)	Dash 8_1	12:54	13:28	10	0	10			37	115	0.58	27%	1956	
9	3	Wau (A)	Rumbek (A)	Dash 8_1	13:48	14:23	0	0	0			37	115	0.58	0%	1956	
9	4	Rumbek (A)	Juba (A)	Dash 8_1	16:30	17:19	24	24	24			37	164	0.82	65%	2785	
10	1	Juba (A)	Paloich (P)	Cessna 208_3	11:30	13:24	7	7	7			10	343	1.90	70%	3257	
10	2	Paloich (P)	Juba (A)	Cessna 208_3	13:44	15:38	5	5	5			10	343	1.90	50%	3257	
11	1	Rumbek (A)	Alek (A)	Cessna 208_2R	09:00	09:47	9	9	9			10	143	0.79	90%	1497	
11	2	Alek (A)	Rumbek (A)	Cessna 208_2R	10:07	10:55	4	4	0	4 (9,4)		10	143	0.79	40%	1497	
12	1	Rumbek (A)	Jiech (H)	Mill Mi8_1R	08:30	09:33	14	14	3			17	127	1.06	82%	4079	
12	2	Jiech (H)	Kurwat (H)	Mill Mi8_1R	09:53	10:20	13	2	5			17	53	0.44	76%	1686	
12	3	Kurwat (H)	Nyal (h)	Mill Mi8_1R	10:40	11:33	8	0	6			17	106	0.88	47%	3394	
12	4	Nyal (h)	Rumbek (A)	Mill Mi8_1R	11:53	12:25	7	5	9	7		91	64	0.54	41%	2059	
13	1	Rumbek (A)	ganyiel (h)	Mill Mi8_2R	12:30	12:59	15	15	9			17	60	0.50	88%	1846	
13	2	ganyiel (h)	Lankien (H)	Mill Mi8_2R	13:19	14:17	9	3	6			17	116	0.97	53%	3592	
13	3	Lankien (H)	Rumbek (a)	Mill Mi8_2R	14:37	16:05	8	5	0	8,5	(9,4)(9,4)	17	175	1.46	47%	5438	
14	1	Rumbek (a)	old fangak (a)	Cessna 208_1	07:00	07:51	4	4	4			10	153	0.85	40%	1531	
14	2	old fangak (a)	Rumbek (a)	Cessna 208_1	08:11	09:02	10	10	10			10	153	0.85	100%	1531	

Figure F2: Planner 1 results, day 2

Manual schedule										Planner: 1		Day: 3				
Route	Leg	Origin	Destination	Aircraft	Departure time	Arrival time	Pax on leg	Pax On	Pax off	Pax Connecting:	Connecting To: (route, #LegID)	Seat Capacity [-]	Ground Distance (NM)	Block [hours]	Load factor [%]	Flight Cost [Cost Units]
1	1	Juba (A)	Rumbek (A)	Fokker 50	07:00	07:42	46	46	20	11:15:18	(10,1,7,2)(11,1)(11,4)(11,2)(7,1)	50	164	0.71	92%	3277
1	2	Rumbek (A)	Juba (A)	Fokker 50	11:30	12:12	28	28	28			50	164	0.71	56%	3277
2	1	Juba (A)	Pochalla (A)	Cessna 208_3	07:00	08:07	9	9	9			10	204	1.13	90%	1934
2	2	Pochalla (A)	Juba (A)	Cessna 208_3	08:27	09:35	6	6	6			10	204	1.13	60%	1934
3	1	Juba (A)	Bor (A)	Cessna 208_2	07:00	07:26	5	5	1			10	79	0.44	50%	728
3	2	Bor (A)	Pochalla (A)	Cessna 208_2	07:46	08:39	9	5	5			10	160	0.89	90%	1472
3	3	Pochalla (A)	Bor (A)	Cessna 208_2	08:59	09:53	9	5	5			10	160	0.89	90%	1472
3	4	Bor (A)	Yei (A)	Cessna 208_2	10:13	10:57	4	0	4			10	135	0.75	40%	1239
3	5	Yei (A)	Juba (A)	Cessna 208_2	11:17	11:40	3	3	3			10	69	0.38	30%	631
4	1	Juba (A)	Mingkaman (H)	Mil Mi8_1	07:00	07:35	17	17	17			17	70	0.59	100%	2253
4	2	Mingkaman (H)	Juba (A)	Mil Mi8_1	07:55	08:30	0	0	0			17	70	0.59	0%	2253
4	3	Juba (A)	Mingkaman (H)	Mil Mi8_1	08:50	09:25	5	5	5			17	70	0.59	29%	2253
4	4	Mingkaman (H)	Juba (A)	Mil Mi8_1	09:45	10:20	9	9	9			17	70	0.59	53%	2253
5	1	Juba (A)	Malakal (A)	Dash 8_1	07:00	08:24	25	25	25			37	281	1.40	68%	4771
5	2	Malakal (A)	Juba (A)	Dash 8_1	08:44	10:08	23	23	23			37	281	1.40	62%	4771
6	1	Juba (A)	Maban (A)	Dash 8_1	11:00	12:39	23	23	23			37	332	1.66	62%	5646
6	2	Maban (A)	Juba (A)	Dash 8_1	12:59	14:39	7	7	7			37	332	1.66	19%	5646
7	1	Rumbek (A)	Wau (A)	Cessna 208_2R	08:15	08:53	9	9	8			10	115	0.64	90%	1208
7	2	Wau (A)	Agok (A)	Cessna 208_2R	09:13	09:48	10	9	1			10	105	0.58	100%	1098
7	3	Agok (A)	Rumbek (A)	Cessna 208_2R	10:08	11:03	9	0	4		5 (1,2)	10	165	0.92	90%	1735
8	1	Juba (A)	Rubkona (A)	Dash 8_2	07:00	08:26	19	19	19			37	287	1.43	51%	5165
8	2	Rubkona (A)	Juba (A)	Dash 8_2	08:46	10:12	25	25	25			37	287	1.43	68%	5165
9	1	Juba (A)	Yida (A)	Dornier 228	07:00	08:29	13	13	13			15	327	1.49	87%	3601
9	2	Yida (A)	Juba (A)	Dornier 228	08:49	10:18	15	15	15			15	327	1.49	100%	3601
10	1	Rumbek (A)	Agok (A)	Cessna 208_1R	08:15	09:10	10	10	10			10	165	0.92	100%	1818
10	2	Agok (A)	Rumbek (A)	Cessna 208_1R	09:30	10:25	8	8	0		8 (1,2)	10	165	0.92	80%	1818
11	1	Rumbek (A)	Leer (H)	Mil Mi8_2R	07:00	07:46	7	7	5			17	93	0.78	41%	2888
11	2	Leer (H)	Haat (h)	Mil Mi8_2R	08:06	08:23	8	6	0			17	33	0.28	47%	1035
11	3	Haat (h)	pagil (h)	Mil Mi8_2R	08:43	09:02	10	2	1			17	39	0.33	59%	1222
11	4	pagil (h)	ganyiel (h)	Mil Mi8_2R	09:22	10:08	9	0	1			17	92	0.76	53%	2839
11	5	ganyiel (h)	Rumbek (a)	Mil Mi8_2R	10:28	10:58	9	1	0		9	17	60	0.50	53%	1846

Figure F.3: Planner 1 results, day 3

Manual Schedule Planner: 1 day: 4																
Route	Leg	Origin	Destination	Aircraft	Departure time	Arrival time	Pax on leg	Pax On	Pax off	Pax Connecting	Connecting To: (route, #LegID)	Seat Capacity [-]	Ground Distance (NM)	Block [hours]	Load factor [%]	Flight factor [Cost Units]
1	1	Juba (A)	Yambio (A)	Dornier 228	07:00	07:48	12	12	12			15	191	0.87	80%	2098
1	2	Yambio (A)	Juba (A)	Dornier 228	08:08	08:48	10	10	10			15	191	0.87	67%	2098
2	1	Juba (A)	Paloch (P)	Dash 8_1	07:00	08:42						37	343	1.71	0%	5828
2	2	Paloch (P)	Mabani (A)	Dash 8_1	09:02	09:27	21	0	11			37	81	0.40	57%	1370
3	1	Juba (A)	Juba (A)	Dash 8_1	09:47	11:26	14	4	10			37	332	1.66	38%	5646
3	2	Mabani (A)	Ajuong Thok (A)	Cessna 208_2	07:00	08:44	10	10	10			10	315	1.75	100%	2897
3	3	Ajuong Thok (A)	Rumbek (A)	Cessna 208_2	09:04	10:08	9	9	0			10	192	1.07	90%	1765
4	1	Juba (A)	Juba (A)	Cessna 208_2	10:28	11:23	10	1	10			10	164	0.91	100%	1507
4	2	Rumbek (A)	Rumbek (A)	Fokker 50	07:00	07:42	49	49	0	7:5		50	164	0.71	98%	3277
4	3	Wau (A)	Wau (A)	Fokker 50	08:02	08:32	37	0	24			50	115	0.50	74%	2301
4	4	Wau (A)	Aweti (A)	Fokker 50	08:52	09:12	15	2	14			50	74	0.32	30%	1477
4	5	Aweti (A)	Rumbek (A)	Fokker 50	09:32	10:19	12	11	1			50	182	0.79	24%	3631
5	1	Rumbek (A)	Juba (A)	Fokker 50	12:30	13:12	50	39	50			50	164	0.71	100%	3277
5	2	Akobo (A)	Akobo (A)	Cessna 208_3	07:00	08:04	10	10	10			10	194	1.08	100%	1838
5	3	Akobo (A)	Akobo (A)	Cessna 208_3	08:24	09:29	6	6	6			10	194	1.08	60%	1838
5	4	Akobo (A)	Juba (A)	Cessna 208_3	09:49	10:53	10	10	10			10	194	1.08	100%	1838
6	1	Juba (A)	Juba (A)	Cessna 208_1	11:13	12:18	0	0	0			10	194	1.08	0%	1838
6	2	Renk (A)	Renk (A)	Cessna 208_1	07:00	09:17	3	3	3			10	412	2.29	30%	4121
7	1	Juba (A)	Juba (A)	Cessna 208_1	09:37	11:54	1	1	1			10	412	2.29	10%	4121
7	2	Pibor (A)	Pibor (A)	Dornier 228	13:45	14:25	14	14	14			15	147	0.67	93%	1620
7	3	Bor (A)	Bor (A)	Dornier 228	14:45	15:11	3	3	3			15	98	0.44	20%	1076
7	4	Pibor (A)	Pibor (A)	Dornier 228	15:31	15:58	7	7	4			15	98	0.44	47%	1076
8	1	Juba (A)	Juba (A)	Dornier 228	16:18	16:58	5	2	5			15	147	0.67	33%	1620
8	2	Torit (A)	Torit (A)	Cessna 208_4	07:00	07:21	5	5	0			10	65	0.36	50%	660
8	3	Kapoeta (A)	Kapoeta (A)	Cessna 208_4	07:41	08:02	5	5	1			10	64	0.36	50%	656
8	4	Juba (A)	Juba (A)	Cessna 208_4	08:22	09:02	10	6	10			10	119	0.66	100%	1216
8	5	Kapoeta (A)	Kapoeta (A)	Cessna 208_4	09:22	10:02	0	0	0			10	119	0.66	0%	1216
9	1	Juba (A)	Rumbek (A)	Dornier 228	09:30	10:14	13	13	5			10	119	0.66	50%	1216
9	2	Rumbek (A)	Pagak (A)	Dornier 228	09:30	10:14	13	13	5			15	164	0.74	87%	1802
9	3	Pagak (A)	Juba (A)	Dornier 228	10:34	11:50	0	0	0			15	278	1.26	0%	3054
10	1	Rumbek (A)	Alek (A)	Cessna 208_2R	08:00	08:47	7	7	7			15	260	1.18	93%	2860
10	2	Alek (A)	Rumbek (A)	Cessna 208_2R	09:07	09:55	3	3	3		3	10	143	0.79	70%	1497
11	1	Rumbek (A)	Nyal (H)	Mil Mi8_2R	08:15	08:47	2	2	0			17	64	0.54	12%	1995
11	2	Nyal (H)	Ganyiel (H)	Mil Mi8_2R	09:07	09:18	5	3	1			17	24	0.20	29%	729
11	3	Ganyiel (H)	Old Fangak (A)	Mil Mi8_2R	09:38	10:30	6	4	0			17	103	0.86	35%	3191
11	4	Old Fangak (A)	Rumbek (A)	Mil Mi8_2R	10:50	12:06	10	4	0			17	153	1.28	59%	4747
12	1	Rumbek (A)	Lankien (H)	Mil Mi8_1R	08:30	09:57	11	11	5			17	175	1.46	65%	5613
12	2	Lankien (H)	Weat (A)	Mil Mi8_1R	10:17	10:28	16	10	6			17	21	0.17	94%	663
12	3	Weat (A)	Rumbek (A)	Mil Mi8_1R	10:48	12:12	12	2	0			17	169	1.41	71%	5417
13	1	Juba (e)	bor (a)	Cessna 208_5	07:00	07:26	10	10	10			10	79	0.44	100%	792
13	2	bor (a)	Juba (a)	Cessna 208_5	07:46	08:12	3					10	79	0.44	30%	792
13	3	Juba (a)	bor (a)	Cessna 208_5	08:52	08:59	5	5	5			10	79	0.44	50%	792
13	4	bor (a)	Juba (a)	Cessna 208_5	09:19	09:45						10	79	0.44	0%	792

Figure F.4: Planner 1 results, day 4

Manual Schedule												Planner: 1		day: 5		
Route	Leg	Origin	Destination	Aircraft	Departure time	Arrival time	Pax on leg	Pax On	Pax off	Pax Connecting:	Connecting To: (route, #LegID)	Seat Capacity (-)	Ground Distance (NM)	Block [hours]	Load factor [%]	Flight Cost [Cost Units]
1	1	Juba (A)	Rumbek (A)	Fokker 50	07:00	07:42	50	50	2	27:23:6:5:5	(1,3):(3,1):(3,2):(3,4):(3,5):(11,1)	50	164	0.71	100%	3277
1	2	Rumbek (A)	Wau (A)	Fokker 50	08:02	08:32	50	50	29			50	115	0.50	100%	2301
1	3	Wau (A)	Juba (A)	Fokker 50	08:52	10:04	37	37	37			50	276	1.20	74%	5518
2	1	Rumbek (A)	Agok (A)	Cessna 208_2R	07:00	07:55	10	10	10			10	165	0.92	100%	1735
2	2	Agok (A)	Rumbek (A)	Cessna 208_2R	08:15	09:10	10	10	10			10	165	0.92	100%	1735
3	1	Rumbek (A)	Menime (H)	Mil Mi8_1R	08:30	09:32	16	16	2		10 (9,3)	17	124	1.03	94%	3969
3	2	Menime (H)	Kuach (H)	Mil Mi8_1R	09:52	10:14	14	0	3			17	45	0.37	82%	1425
3	3	Kuach (H)	Koch (H)	Mil Mi8_1R	10:34	10:43	11	0	5		6	17	18	0.15	65%	566
3	4	Koch (H)	Leer (H)	Mil Mi8_1R	11:03	11:12	14	8	6		8	17	19	0.16	82%	614
3	5	Leer (H)	Rumbek (A)	Mil Mi8_1R	11:32	12:19	15	7	0		15	17	93	0.78	88%	2981
4	1	Juba (A)	Mingkaman (H)	Mil Mi8_1	07:00	07:35	8	8	6			17	70	0.59	47%	2253
4	2	Mingkaman (H)	Bor (A)	Mil Mi8_1	07:55	08:00	6	4	6			17	10	0.09	35%	330
4	3	Bor (A)	Mingkaman (H)	Mil Mi8_1	08:20	08:25	0	0	0			17	10	0.09	0%	330
4	4	Mingkaman (H)	Juba (A)	Mil Mi8_1	08:45	09:20	15	15	15			17	70	0.59	88%	2253
5	1	Juba (A)	Maban (A)	Dash 8_1	07:00	08:39	22	22	22			37	332	1.66	59%	5646
5	2	Maban (A)	Juba (A)	Dash 8_1	08:59	10:39	34	34	34			37	332	1.66	92%	5646
6	1	Juba (A)	Bor (A)	Cessna 208_2	07:00	07:26	9	9	9			10	79	0.44	90%	728
6	2	Bor (A)	Juba (A)	Cessna 208_2	07:46	08:12	9	9	9			10	79	0.44	90%	728
7	1	Juba (A)	Malakai (A)	Dash 8_2	07:00	08:24	21	21	21			37	281	1.40	57%	5052
7	2	Malakai (A)	Juba (A)	Dash 8_2	08:44	10:08	16	16	16			37	281	1.40	43%	5052
8	1	Juba (A)	Yrda (A)	Dash 8_1	12:00	13:38	25	25	9			37	327	1.64	68%	5565
8	2	Yrda (A)	Rubkona (A)	Dash 8_1	13:58	14:13	30	14	16			37	51	0.26	81%	871
8	3	Rubkona (A)	Juba (A)	Dash 8_1	14:33	15:59	37	23	37			37	287	1.43	100%	4878
9	1	Juba (A)	Awelil (A)	Fokker 50	11:30	13:00	40	40	20			50	345	1.50	80%	6903
9	2	Awelil (A)	Rumbek (A)	Fokker 50	13:20	14:07	32	12	20			50	182	0.79	64%	3631
9	3	Rumbek (A)	Juba (A)	Fokker 50	14:27	15:10	45	33	45			50	164	0.71	90%	3277
10	1	Juba (A)	Yambio (A)	Cessna 208_3	07:00	08:03	10	10	10			10	191	1.06	100%	1812
10	2	Yambio (A)	Juba (A)	Cessna 208_3	08:23	09:27	10	10	10			10	191	1.06	100%	1812
10	3	Juba (A)	Yambio (A)	Cessna 208_3	09:47	10:50	9	9	9			10	191	1.06	90%	1812
10	4	Yambio (A)	Juba (A)	Cessna 208_3	11:10	12:14	6	6	6			10	191	1.06	60%	1812
11	1	Rumbek (A)	Wai (H)	Mil Mi8_2R	09:00	10:04	5	5	5			17	129	1.08	29%	4003
11	2	Wai (H)	Rumbek (A)	Mil Mi8_2R	10:24	11:29	0	0	0			17	129	1.08	0%	4003
12	1	Juba (a)	wau (a)	Cessna 208_1	07:00	08:31	15	15	15			10	276	1.53	150%	2759
12	2	wau (a)	Juba (a)	Cessna 208_1	08:51	10:23	0	0	0			10	276	1.53	0%	2759

Figure F5: Planner 1 results, day 5

Manual schedule										Planner: 1										Day: 1									
Route	Leg	Origin	Destination	Aircraft	Departure time	Arrival time	Pax on leg	Pax On	Pax off	Pax Connecting:	Connecting To: (route, #Leg/D)	Seat Capacity [-]	Ground Distance (NM)	Block [hours]	Load factor [%]	Flight Cost [Cost Units]													
1	1	Juba (a)	Rumbek (a)	Fokker 50	07:00	07:42	49	49	13	19	Agok, leer, koch, mankein	50	164	0.71	98%	3277													
1	2	Rumbek (a)	Aweil (a)	Fokker 50	08:02	08:30	37	20	17			50	182	0.79	74%	3631													
1	3	Aweil (a)	Juba (a)	Fokker 50	09:10	10:40	36	16	34	2		72	345	1.50	72%	6903													
2	1	Rumbek (a)	Agok (a)	Cessna 208_1R	07:00	07:55	10	10	10			10	165	0.92	100%	1818													
2	2	Agok (a)	Rumbek (a)	Cessna 208_1R	08:15	09:10	9	9	9	9		72	10	165	0.92	90%	1818												
3	1	Rumbek (a)	Leer (h)	MilliM8_1R	07:00	07:46	8	8	4			17	93	0.78	47%	2981													
3	2	Leer (h)	koch (h)	MilliM8_1R	08:06	08:16	9	5	4			17	19	0.16	53%	614													
3	3	koch (h)	rumbek (a)	MilliM8_1R	08:36	09:30	6	1	0	6		72	17	109	0.91	35%	3488												
4	1	Juba (a)	yda (a)	Dornier 228	07:00	08:29	15	15	15			15	327	1.49	100%	3601													
4	2	yda (a)	Juba (a)	Dornier 228	08:49	10:18	10	10	10			15	327	1.49	67%	3601													
4	3	Juba (a)	plbor (a)	Dornier 228	10:38	11:18	15	15	15			15	147	0.67	100%	1620													
4	4	plbor (a)	Juba (a)	Dornier 228	11:38	12:18	9	9	9			15	147	0.67	60%	1620													
5	1	Juba (a)	wau (a)	Dash 8_2	07:00	08:22	29	29	29			37	276	1.38	78%	4966													
5	2	wau (a)	rumbek (a)	Dash 8_2	08:42	09:17	22	22	2			37	115	0.58	59%	2071													
5	3	rumbek (a)	Juba (a)	Dash 8_2	09:37	10:26	35	15	35			37	164	0.82	95%	2949													
6	1	Juba (a)	rubkona (a)	Dash 8_1	07:00	08:26	28	28	28			37	287	1.43	76%	4878													
6	2	rubkona (a)	Juba (a)	Dash 8_1	08:46	10:12	15	15	15			37	287	1.43	41%	4878													
6	3	Juba (a)	malakal (a)	Dash 8_1	10:32	11:56	35	35	35			37	281	1.40	95%	4771													
6	4	malakal (a)	Juba (a)	Dash 8_1	12:16	13:40	16	16	16			37	281	1.40	43%	4771													
7	1	rumbek (a)	rubkona (a)	Cessna 208_2R	07:00	07:49	1	1	0			10	150	0.83	10%	1570													
7	2	rubkona (a)	mankain (a)	Cessna 208_2R	08:09	08:24	3	2	3			10	45	0.25	30%	467													
7	3	mankain (a)	rumbek (a)	Cessna 208_2R	08:44	09:30	0	0	0			10	138	0.77	0%	1451													
8	1	Juba (a)	Mingkaman (h)	MilliM8_1	07:00	07:35	15	15	15			17	70	0.59	88%	2253													
8	2	Mingkaman (h)	BOR (A)	MilliM8_1	07:55	08:00	7	7	7			17	10	0.09	41%	330													
8	3	BOR (A)	JUBA (A)	MilliM8_1	08:20	08:59	13	6	13			17	79	0.66	76%	2533													
9	1	Juba (a)	bor (a)	Cessna 208_1	07:00	07:26	7	7	6			10	79	0.44	70%	792													
9	2	bor (a)	plbor (a)	Cessna 208_1	07:46	08:18	9	8	9			10	98	0.54	90%	978													
9	3	plbor (a)	Juba (a)	Cessna 208_1	08:38	09:28	0	0	0			10	147	0.82	0%	1472													
10	1	JUBA (A)	YAMBIO (A)	Dash 8_2	07:00	07:57	12	12	12			37	191	0.95	32%	3432													
10	2	YAMBIO (A)	JUBA (A)	Dash 8_2	08:17	09:14	20	20	20			37	191	0.95	54%	3432													

Figure F.6: Planner 2 results, day 1

Manual schedule											Planner: 1		Day: 2				
Route	Leg	Origin	Destination	Aircraft	Departure time	Arrival time	Pax on leg	Pax On	Pax off	Pax Connecting:	Connecting To: (route, #LegID)	Seat Capacity [-]	Ground Distance (NM)	Block [hours]	Load factor [%]	Flight Cost [Cost Units]	
1	1	Juba (a)	Ajuong Thok (A)	Cessna 208_3	07:45	09:29	5	5	5			10	315	1.75	50%	2992	
1	2	Ajuong Thok (A)	Juba (a)	Cessna 208_3	09:49	11:34	10	10	10			10	315	1.75	100%	2992	
2	1	Juba (a)	Rumbek (a)	Fokker 50	07:15	07:57	50	50	11	39	atek ganyiel jiech, mogoknyal, kurwai	50	164	0.71	100%	3277	
2	2	Rumbek (a)	Juba (a)	Fokker 50	13:00	13:42	23	23	23			50	164	0.71	46%	3277	
3	1	Rumbek (a)	Ganyiel (h)	Mill Mi8_1R	08:45	09:14	15	15	5			17	60	0.50	88%	1905	
3	2	Ganyiel (h)	jiech (h)	Mill Mi8_1R	09:34	10:09	10	0	3			17	70	0.58	59%	2235	
3	3	jiech (h)	Mogok (a)	Mill Mi8_1R	10:29	10:35	9	2	7			17	12	0.10	53%	383	
3	4	Mogok (a)	Rumbek (a)	Mill Mi8_1R	10:55	12:04	2	0	0	2		91	17	138	1.15	12%	4407
4	1	Rumbek (a)	Nyal (h)	Mill Mi8_2R	08:30	09:02	15	15	6			17	64	0.54	88%	1995	
4	2	Nyal (h)	Kurwai (h)	Mill Mi8_2R	09:22	10:15	14	5	5			17	106	0.88	82%	3288	
4	3	Kurwai (h)	Old Fangak (a)	Mill Mi8_2R	10:35	10:44	9	0	4			17	20	0.16	53%	606	
4	4	Old Fangak (a)	Rumbek (a)	Mill Mi8_2R	11:04	12:21	17	12	0	17		91	17	153	1.28	100%	4747
5	1	Rumbek (a)	Aiek (a)	Cessna 208_1R	09:00	09:47	9	9	9			10	143	0.79	90%	1568	
5	2	Aiek (a)	Rumbek (a)	Cessna 208_1R	10:07	10:55	4	4	4	4		91	10	143	0.79	40%	1568
6	1	Juba (a)	Aweil (a)	Dash 8_2	08:00	09:43	32	32	12			37	345	1.73	86%	6212	
6	2	Aweil (a)	wau (a)	Dash 8_2	10:03	10:25	36	16	13			37	74	0.37	97%	1330	
6	3	wau (a)	Rumbek (a)	Dash 8_2	10:45	11:20	32	9	9			37	115	0.58	86%	2071	
6	4	Rumbek (a)	Juba (a)	Dash 8_2	11:40	12:29	29	6	29			37	164	0.82	78%	2949	
7	1	Juba (a)	Pagak (a)	Cessna 208_1	07:45	09:11	10	10	10			10	260	1.44	100%	2600	
7	2	Pagak (a)	Juba (a)	Cessna 208_1	09:31	10:58	4	4	4			10	260	1.44	40%	2600	
7	3	Juba (a)	Pagak (a)	Cessna 208_1	11:18	12:44	10	10	10			10	260	1.44	100%	2600	
7	4	Pagak (a)	Juba (a)	Cessna 208_1	13:04	14:31	0	0	0			10	260	1.44	0%	2600	
8	1	Juba (a)	Maban (a)	Dash 8_1	07:30	09:09	34	27				37	332	1.66	0%	5646	
8	2	Maban (a)	Malakal (a)	Dash 8_1	09:29	10:07	16	7	18			37	127	0.63	0%	2158	
8	3	Malakal (a)	Juba (a)	Dash 8_1	10:27	11:51			2			37	281	1.40	0%	4771	
9	1	Juba (a)	bor (a)	Mill Mi8_1	08:15	08:54	17	17				17	79	0.66	100%	2533	
9	2	bor (a)	Motot (a)	Mill Mi8_1	09:14	10:15	17	2	2			17	121	1.01	100%	3882	
9	3	Motot (a)	Lankien (h)	Mill Mi8_1	10:35	10:45	16	1	9			17	21	0.18	94%	687	
9	4	Lankien (h)	Mabior (a)	Mill Mi8_1	11:05	11:51	10	3	6			17	91	0.76	59%	2906	
9	5	Mabior (a)	bor (a)	Mill Mi8_1	12:11	12:40	8	4				17	59	0.49	47%	1885	
9	6	bor (a)	Juba (a)	Mill Mi8_1	13:00	13:40	8	0	8			17	79	0.66	47%	2533	
10	1	Juba (a)	Paloch (P)	Cessna 208_2	08:30	10:24	7	7	7			10	343	1.90	70%	3154	
10	2	Paloch (P)	Juba (a)	Cessna 208_2	10:44	12:38	5	5	5			10	343	1.90	50%	3154	
11	1	Juba (a)	Kapoeta (a)	Dornier 228	09:00	09:32	15	15	11			15	119	0.54	100%	1311	
11	2	Kapoeta (a)	Torit (a)	Dornier 228	09:52	10:10	15	11	5			15	64	0.29	100%	707	
11	3	Torit (a)	Juba (a)	Dornier 228	10:30	10:47	11	1	12			15	65	0.29	73%	712	

Figure F.7: Planner 2 results, day 2

Manual schedule										Planner: 1		Day: 3					
Route	Leg	Origin	Destination	Aircraft	Departure time	Arrival time	Pax on leg	Pax On	Pax off	Pax Connecting:	Connecting To: (route, #LegID)	Seat Capacity [-]	Ground Distance (NM)	Block [hours]	Load factor [%]	Flight Cost [Cost Units]	
1	1	Juba (a)	rumbek (a)	Fokker 50	07:15	07:57	45	45		25	leer,pagil,ganyiel,agok,wau	50	164	0.71	90%	3277	
1	2	rumbek (a)	Juba (a)	Fokker 50	12:00	12:42	29	29	29			50	164	0.71	58%	3277	
2	1	Rumbek (a)	Leer (h)	Mil Mi8_1R	08:15	09:01	7	7	5			17	93	0.78	41%	2981	
2	2	Leer (h)	Haat (h)	Mil Mi8_1R	09:21	09:38	8	6	0			17	33	0.28	47%	1068	
2	3	Haat (h)	pagil (h)	Mil Mi8_1R	09:58	10:17	10	2	1			17	39	0.33	59%	1262	
2	4	pagil (h)	ganyiel (h)	Mil Mi8_1R	10:37	11:23	9	0	1			17	92	0.76	53%	2930	
2	5	ganyiel (h)	Rumbek (a)	Mil Mi8_1R	11:43	12:13	9	1	0	9		91	60	0.50	53%	1905	
3	1	Rumbek (a)	agok (a)	Cessna 208_1R	08:30	09:25	10	10	10			10	165	0.92	100%	1818	
3	2	agok (a)	Rumbek (a)	Cessna 208_1R	09:45	10:40	7	7		7		91	10	165	0.92	70%	1818
4	1	Rumbek (a)	wau (a)	Cessna 208_2R	08:45	09:23	8	8	8			10	115	0.64	80%	1208	
4	2	wau (a)	Rumbek (a)	Cessna 208_2R	09:43	10:21	9	9	4	5		91	10	115	0.64	90%	1208
5	1	Juba (a)	yida (a)	Dornier 228	07:30	08:59	13	13	13			15	327	1.49	87%	3601	
5	2	yida (a)	Juba (a)	Dornier 228	09:19	10:48	15	15	15			15	327	1.49	100%	3601	
5	3	Juba (a)	bor (a)	Dornier 228	11:08	11:30	10	10	1			15	79	0.36	67%	871	
5	4	bor (a)	Pochalla (a)	Dornier 228	11:50	12:33	14	5	14			15	160	0.73	93%	1760	
5	5	Pochalla (a)	bor (a)	Dornier 228	12:53	13:37	11	11	5			15	160	0.73	73%	1760	
5	6	bor (a)	Juba (a)	Dornier 228	13:57	14:19	6	0	6			15	79	0.36	40%	871	
6	1	Juba (a)	Yei (a)	Cessna 208_3	07:00	07:22	4	4	4			10	69	0.38	40%	652	
6	2	Yei (a)	Juba (a)	Cessna 208_3	07:42	08:05	3	3	3			10	69	0.38	30%	652	
7	1	Juba (a)	rubkona (a)	Dash 8_2	08:00	09:26	19	19	19			37	287	1.43	51%	5165	
7	2	rubkona (a)	juba (a)	Dash 8_2	09:46	11:12	25	25	25			37	287	1.43	68%	5165	
8	1	Juba (a)	maban (a)	Dash 8_1	07:45	09:24			23	37		37	332	1.66	0%	5646	
8	2	maban (a)	malakal (a)	Dash 8_1	09:44	10:22			7	14		37	127	0.63	0%	2158	
8	3	malakal (a)	juba (a)	Dash 8_1	10:42	12:06			23	30		37	281	1.40	0%	4771	
9	1	Juba (a)	malakal (a)	Cessna 208_1	08:15	09:48	0	0	0			10	281	1.56	0%	2807	
9	2	malakal (a)	Juba (a)	Cessna 208_1	10:08	11:42	9	9	9			10	281	1.56	90%	2807	
10	1	Juba (a)	mingkaman (h)	Mil Mi8_2	08:30	09:05	17	17	17			17	70	0.59	100%	2253	
10	2	mingkaman (h)	juba (a)	Mil Mi8_2	09:25	10:00	9	9	9			17	70	0.59	53%	2253	
10	3	juba (a)	mingkaman (h)	Mil Mi8_2	10:20	10:55	5	5	5			17	70	0.59	29%	2253	
10	4	mingkaman (h)	juba (a)	Mil Mi8_2	11:15	11:50	0	0	0			17	70	0.59	0%	2253	

Figure F8: Planner 2 results, day 3

Manual schedule										Planner: 1		Day: 4					
Route	Leg	Origin	Destination	Aircraft	Departure time	Arrival time	Pax on leg	Pax On	Pax off	Pax Connecting:	Connecting To: (route, #Leg/D)	Seat Capacity [-]	Ground Distance (NM)	Block [hours]	Load factor [%]	Flight Cost [Cost Units]	
1	1	juba (a)	yambio (a)	Cessna 208_2	07:00	08:03	10	10	10			10	191	1.06	100%	1754	
1	2	yambio (a)	juba (a)	Cessna 208_2	08:23	09:27	10	10	10			10	191	1.06	100%	1754	
2	1	juba (a)	waat (a)	MilliM18_1	07:00	08:40	11	11	6			17	202	1.68	65%	6454	
2	2	waat (a)	lankien (h)	MilliM18_1	09:00	09:11	7	2	5			17	21	0.17	41%	663	
2	3	lankien (h)	juba (a)	MilliM18_1	09:31	11:21	12	10	12			17	221	1.84	71%	7062	
3	1	juba (a)	maban (a)	Dash 8_2	07:00	08:39	21	21	10			37	332	1.66	57%	5978	
3	2	maban (a)	paloch (p)	Dash 8_2	08:59	09:23	28	17	11			37	81	0.40	76%	1450	
3	3	paloch (p)	juba (a)	Dash 8_2	09:43	11:26	21	4	21			37	343	1.71	57%	6171	
4	1	juba (a)	bor (a)	Cessna 208_1	07:00	07:26	10	10	7			10	79	0.44	100%	792	
4	2	bor (a)	renk (a)	Cessna 208_1	07:46	09:37	3	0	3			10	334	1.86	30%	3343	
4	3	renk (a)	juba (a)	Cessna 208_1	09:57	12:15	1	1	1			10	412	2.29	10%	4121	
4	4	juba (a)	ajuong thok (a)	Cessna 208_1	12:35	14:20	10	10	10		51	10	315	1.75	100%	3149	
4	5	ajuong thok (a)	juba (a)	Cessna 208_1	14:40	16:25	9	9	9			10	315	1.75	90%	3149	
5	1	juba (a)	torit (a)	Dornier 228	07:00	07:17	5	5	5		0	15	65	0.29	33%	712	
5	2	torit (a)	kapoeta (a)	Dornier 228	07:37	07:55	10	5	6			15	64	0.29	67%	707	
5	3	kapoeta (a)	juba (a)	Dornier 228	08:15	08:47	15	11	15			15	119	0.54	100%	1311	
5	4	juba (a)	bor (a)	Dornier 228	09:07	09:29	15	15	8			15	79	0.36	100%	871	
5	5	bor (a)	plibor (a)	Dornier 228	09:49	10:15	11	4	11			15	98	0.44	73%	1076	
5	6	plibor (a)	juba (a)	Dornier 228	10:35	11:16	2	2	2			15	147	0.67	13%	1620	
5	7	juba (a)	pagak (a)	Dornier 228	11:36	12:47	0	0	0			15	260	1.18	0%	2860	
5	8	pagak (a)	juba (a)	Dornier 228	13:07	14:17	14	14	14			15	260	1.18	93%	2860	
6	1	juba (a)	akobo (a)	Cessna 208_5	07:00	08:04	10	10	10			10	194	1.08	100%	1935	
6	2	akobo (a)	juba (a)	Cessna 208_5	08:24	09:29	6	6	6			10	194	1.08	60%	1935	
6	3	juba (a)	akobo (a)	Cessna 208_5	09:49	10:53	10	10	10			10	194	1.08	100%	1935	
6	4	akobo (a)	juba (a)	Cessna 208_5	11:13	12:18	0	0	0			10	194	1.08	0%	1935	
7	1	rumbek (a)	alek (a)	Cessna 208_1R	07:00	07:47	7	7	7			10	143	0.79	70%	1568	
7	2	alek (a)	rumbek (a)	Cessna 208_1R	08:07	08:55	3	3	3		3	91	10	143	0.79	30%	1568
8	1	rumbek (a)	nyal (h)	MilliM18_1R	07:00	07:32	2	2	0			17	64	0.54	12%	2059	
8	2	nyal (h)	ganyiel (h)	MilliM18_1R	07:52	08:03	5	3	3			17	24	0.20	29%	753	
8	3	ganyiel (h)	rumbek (a)	MilliM18_1R	08:23	08:53	6	4	4		6	91	17	60	0.50	35%	1905
9	1	juba (a)	rumbek (a)	Fokker 50	07:00	07:42	50	50	5		9	50	164	0.71	100%	3277	
9	2	rumbek (a)	wau (a)	Fokker 50	08:02	08:32	36	0	23			50	115	0.50	72%	2301	
9	3	wau (a)	aweil (a)	Fokker 50	08:52	09:12	22	9	14			50	74	0.32	44%	1477	
9	4	aweil (a)	rumbek (a)	Fokker 50	09:32	10:19	19	11	1			50	182	0.79	38%	3631	
9	5	rumbek (a)	juba (a)	Fokker 50	10:39	11:22	35	17	35			50	164	0.71	70%	3277	
10	1	juba (a)	plibor (a)	Cessna 208_3	07:00	07:49	7	7	7			10	147	0.82	70%	1399	
10	2	plibor (a)	bor (a)	Cessna 208_3	08:09	08:41	3	3	3			10	98	0.54	30%	929	
10	3	bor (a)	juba (a)	Cessna 208_3	09:01	09:28	3	3	3			10	79	0.44	30%	752	

Figure F.9: Planner 2 results, day 4

Manual Schedule											Planner: 2				day: 5			
Route	Leg	Origin	Destination	Aircraft	Departure time	Arrival time	Pax on leg	Pax On	Pax off	Pax Connecting	Connecting To: (route, #LegID)	Seat Capacity (-)	Ground Distance (NM)	Block [hours]	Load factor [%]	Flight Cost [Cost Units]		
1	1	Juba (A)	Rumbek (A)	Fokker 50	07:15	08:00	38	38	13	4/4		50	164	0.71	76%	3277		
1	2	Rumbek (A)	Awelil (A)	Fokker 50	08:20	08:41	17	0	17			50	182	0.79	34%	3631		
1	3	Awelil (A)	Rumbek (A)	Fokker 50	09:01	09:20	16	16	2	14		50	182	0.79	32%	3631		
1	4	Rumbek (A)	Juba (A)	Fokker 50	11:00	11:48	40	26	40			50	164	0.71	80%	3277		
2	1	Juba (A)	Rumbek (A)	Dash 8_1	07:15	08:04	37	37			21 (6.1):(6.1):(6.1):(7.1):(7.1)	37	164	0.82	100%	2785		
2	2	Rumbek (A)	Rubkona (A)	Dash 8_1	08:24	09:09	23	7	16			37	150	0.75	62%	2542		
2	3	Rubkona (A)	Juba (A)	Dash 8_1	09:29	10:55	30	23	37			37	287	1.43	81%	4878		
2	4	Juba (A)	Mabani (A)	Dash 8_1	11:15	12:54	15	22	22			37	332	1.66	41%	5646		
2	5	Mabani (A)	Juba (A)	Dash 8_1	13:14	14:54	27	34	34			37	332	1.66	73%	5646		
3	1	Juba (A)	Agok (A)	Cessna 208_1	07:00	08:47	10	10	10			10	323	1.80	100%	3231		
3	2	Agok (A)	Juba (A)	Cessna 208_1	09:07	10:55	10	10	10			10	323	1.80	100%	3231		
4	1	Juba (A)	Yida (A)	Dornier 228	07:45	09:14	9	9	9			15	327	1.49	60%	3601		
4	2	Yida (A)	Juba (A)	Dornier 228	09:34	11:03	14	14	14			15	327	1.49	93%	3601		
5	1	Juba (A)	Mingkaman (H)	Mil Mi8_1	08:00	08:35	17	17	6			17	70	0.59	100%	2253		
5	2	Mingkaman (H)	Bor (A)	Mil Mi8_1	08:55	09:00	15	4	15			17	10	0.09	88%	330		
5	3	Bor (A)	Mingkaman (H)	Mil Mi8_1	09:20	09:25	2	2	0			17	10	0.09	12%	330		
5	4	Mingkaman (H)	Juba (A)	Mil Mi8_1	09:45	10:20	17	15	17			17	70	0.59	100%	2253		
6	1	Rumbek (A)	Leer (H)	Mil Mi8_1R	08:30	09:16	14	14	6			17	93	0.78	82%	2981		
6	2	Leer (H)	Koch (H)	Mil Mi8_1R	09:36	09:46	15	7	5			17	19	0.16	88%	614		
6	3	Koch (H)	Kuach (H)	Mil Mi8_1R	10:06	10:15	18	8	3			17	18	0.15	106%	566		
6	4	Kuach (H)	Rumbek (A)	Mil Mi8_1R	10:35	11:38	15	0			15 (1.4):(1.4)	17	127	1.06	88%	4053		
7	1	Rumbek (A)	Menime (H)	Mil Mi8_2R	08:45	09:47	7	7	2			17	124	1.03	41%	3845		
7	2	Menime (H)	Wai (H)	Mil Mi8_2R	10:07	10:27	5	0	5			17	40	0.34	29%	1247		
7	3	Wai (H)	Rumbek (A)	Mil Mi8_2R	10:47	11:51	0	0	0			17	129	1.08	0%	4003		
8	1	Juba (A)	Yambio (A)	Dash 8_2	07:30	08:27	37	37	19			37	191	0.95	100%	3432		
8	2	Yambio (A)	Rumbek (A)	Dash 8_2	08:47	09:33	34	16	18			37	155	0.77	92%	2784		
8	3	Rumbek (A)	Juba (A)	Dash 8_2	09:53	10:42	34	18	34			37	164	0.82	92%	2949		
8	4	Juba (A)	Malakal (e)	Dash 8_2	11:02	12:26	21	21	21			37	281	1.40	57%	5052		
8	5	Malakal (e)	Juba (A)	Dash 8_2	12:46	14:11	16	16	16			37	281	1.40	43%	5052		
9	1	Juba (a)	bor (a)	Cessna 208_5	07:00	07:26	0	0	0			10	79	0.44	0%	792		
9	2	bor (a)	Juba (a)	Cessna 208_5	07:46	08:12	7	7	7			10	79	0.44	70%	792		

Figure F:10: Planner 2 results, day 5

Bibliography

- Abara, J. (1989), 'Applying Integer Linear Programming to the Fleet Assignment Problem', *Interfaces* **19**(4), 20–28.
- Altinkemer, K. and Gavish, B. (1991), 'Parallel Savings Based Heuristic for the Delivery Problem', *Operations Research* **39**(3), 456–469.
- Argüello, M. F., Bard, J. F. and Yu, G. (1997), 'A GRASP for aircraft routing in response to groundings and delays', *Journal of Combinatorial Optimization* **1**(3), 211–228.
- Baker, B. M. and Ayechev, M. A. (2003), 'A genetic algorithm for the vehicle routing problem', *Computers & Operations Research* **30**(5), 787–800.
- Baldacci, R., Hadjiconstantinou, E. and Mingozzi, A. (2004), 'An Exact Algorithm for the Capacitated Vehicle Routing Problem Based on a Two-Commodity Network Flow Formulation', *Operations Research* **52**(5), 723–738.
- Baldacci, R., Mingozzi, A. and Roberti, R. (2012), 'Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints', *European Journal of Operational Research* **218**(1), 1–6.
- Barbarosoglu, G. and Ozgur, D. (1999), 'A tabu search algorithm for the vehicle routing problem', *Computers & Operations Research* **26**(3), 255–270.
- Bard, J. F., Yu, G. and Argüello, M. F. (2001), 'Optimizing aircraft routings in response to groundings and delays'.
- Barnhart, C. (2008), *Airline Scheduling: Accomplishments, Opportunities and Challenges*, in 'Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems', Springer Berlin Heidelberg, Berlin, Heidelberg.
- Barnhart, C., Boland, N. L., Clarke, L. W., Johnson, E. L., Nemhauser, G. L. and Shenoi, R. G. (1998), 'Flight String Models for Aircraft Fleeting and Routing', *Transportation Science* **32**(3), 208–220.
- Barnhart, C., Kniker, T. S. and Lohatepanont, M. (2002), 'Itinerary-Based Airline Fleet Assignment', *Transportation Science* **36**(2), 199–217.
- Bazargan, M. (2010), *Airline Operations and Scheduling*, 2 edn, Ashgate Pub. Ltd.
- Belobaba, P., Odoni, A. and Barnhart, C. (2009), *The Global Airline Industry*, John Wiley & Sons, Ltd, Chichester, West Sussex, UK.
- Berger, J. and Barkaoui, M. (2003), 'A new hybrid genetic algorithm for the capacitated vehicle routing problem', *Journal of the Operational Research Society* **54**(12), 1254–1262.
- Bodin, L. D. (1975), 'A taxonomic structure for vehicle routing and scheduling problems', *Computers & Urban Society* **1**(1), 11–29.
- Bullnheimer, B., Hartl, R. F. and Strauss, C. (1999), 'An improved ant system algorithm for the vehicle routing problem', *Annals of Operations Research* **89**(POM-10/97), 319–328.
- Chircop, K. and Zammit-Mangion, D. (2013), 'On Epsilon-Constraint Based Methods for the Generation of Pareto Frontiers', *Journal of Mechanics Engineering and Automation* **3**(September), 279–289.
- Christofides, N. (1976), 'The Vehicle Routing Problem', *Recherche opérationnelle* **10**(2), 55–70.
- Clarke, G. and Wright, J. (1964), 'Scheduling of Vehicles from a Central Depot to a Number of Delivery Points', *Operations Research* **12**(4), 568–581.

- Clarke, L., Johnson, E., Nemhauser, G. and Zhu, Z. (1996), 'The aircraft rotation problem', *Transportation Research Part A: Policy and Practice* **30**(1), 51.
- Clarke, L. W., Hane, C. a., Johnson, E. L. and Nemhauser, G. L. (1996), 'Maintenance and Crew Considerations in Fleet Assignment', *Transportation Science* **30**(3), 249–260.
- Clough, M. C., Jacobs, T. L. and Gel, E. S. (n.d.), 'New Formulations for Price and Ticket Availability Decisions in Choice-based Network Revenue Management'.
- Cohn, A. M. and Barnhart, C. (2003), 'Improving Crew Scheduling by Incorporating Key Maintenance Routing Decisions', *Operations Research* **51**(3), 387–396.
- Cordeau, J.-F. (2006), 'A Branch-and-Cut Algorithm for the Dial-a-Ride Problem', *Operations Research* **54**(3), 573–586.
- Cordeau, J. F., Desaulniers, G., Desrosiers, J., Solomon, M. M. and Soumis, F. (2002), VRP with Time Windows, in P. Toth and D. Vigo, eds, 'The vehicle routing problem', Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, chapter 7.
- Cordeau, J. F. and Laporte, G. (2007), 'The dial-a-ride problem: Models and algorithms', *Annals of Operations Research* **153**(1), 29–46.
- Cordeau, J.-F., Stojković, G., Soumis, F. and Desrosiers, J. (2001), 'Benders Decomposition for Simultaneous Aircraft Routing and Crew Scheduling', *Transportation Science* **35**(4), 375–388.
- Dantzig, G. B. and Ramser, J. H. (1959), 'The Truck Dispatching Problem', *Management Science* **6**(1), 80–91.
- De Jaegere, N., Defraeye, M. and Van Nieuwenhuyse, I. (2014), 'The vehicle routing problem: state of the art classification and review', *Research Report* p. 30.
- Desaulniers, G., Desrosiers, J., Dumas, Y., Solomon, M. M. and Soumis, F. (1997), 'Daily Aircraft Routing and Scheduling', *Management Science* **43**(6), 841–855.
- Desrochers, M. and Laporte, G. (1991), 'Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints', *Operations Research Letters* **10**(1), 27–36.
- Desrochers, M., Lenstra, J. and Savelsbergh, M. (1990), 'A classification scheme for vehicle routing and scheduling problems', *European Journal of Operational Research* **46**(3), 322–332.
- Desrochers, M. and Verhoog, T. (1989), 'A matching based savings algorithm for the vehicle routing problem', *Les Cahiers du GERAD G-89-04*.
- Desrosiers, J., Dumas, Y., Solomon, M. M. and Soumis, F. (1995), 'Time Constrained Routing and Scheduling', *Handbooks in Operations Research and Management Science* **8**, 35–139.
- Dorn, A. W. and Cross, R. W. (2014), 'Flying Humanitarians: The UN Humanitarian Air Service', *Air Power in UN Operations: Wings for Peace*.
- Eksioglu, B., Vural, A. V. and Reisman, A. (2009), 'The vehicle routing problem: A taxonomic review', *Computers and Industrial Engineering* **57**(4), 1472–1483.
- Etschmaier, M. M. and Mathaisel, D. F. X. (1985), 'Airline Scheduling: An Overview', *Transportation Science* **19**(2), 127–138.
- Fisher, M. (1995), 'Vehicle routing', *Handbooks in Operations Research and Management Science* **8**(C), 1–33.
- Fisher, M. L. and Jaikumar, R. (1981), 'A generalized assignment heuristic for vehicle routing', *Networks* **11**(2), 109–124.
- Foster, A. B. A. and Ryan, D. M. (1976), 'An Integer Programming Approach to the Vehicle Scheduling Problem', *Operational Research Quarterly* **27**(2), 367–384.
- Gagnon, G. (1967), 'A Model for Flowing Passengers over Airline Networks'.

- Garey, M. R. and Johnson, D. S. (1979), *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., New York, NY, USA.
- Gen, M. and Cheng, R. (1997), *Genetic Algorithms and Engineering Design*, John Wiley & Sons, Ltd, New York, NY, USA.
- Gendreau, M., Hertz, a. and Laporte, G. (1994), 'A Tabu Search Heuristic for the Vehicle Routing Problem', *Management Science* **40**(10), 1276–1290.
- Gillett, B. and Miller, L. (1974), 'A Heuristic Algorithm for the Vehicle Dispatch Problem', *Operations Research* **22**(2), 340–349.
- Golden, B., Raghavan, S. and Wasil, E. (2008), *The Vehicle Routing Problem: Latest Advances and New Challenges*, Vol. 43, Springer Science+Business Media.
- Grosche, T., Heinzl, A. and Rothlauf, F. (2001), 'A Conceptual Approach for Simultaneous Flight Schedule Construction with Genetic Algorithms Conference', *Theoretical Computer Science*.
- Gurobi Optimizer* (2019).
URL: <https://www.gurobi.com>
- Haimes, Y., Lasdon, L. and Wismer, D. (1971), 'On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization', *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-1**(3), 296–297.
- Hane, C. A., Barnhart, C., Johnson, E. L., Marsten, R. E., Nemhauser, G. L. and Sigismondi, G. (1995), 'The fleet assignment problem: Solving a large-scale integer program', *Mathematical Programming* **70**(1-3), 211–232.
- IBM Decision Optimization CPLEX Modeling for Python* (2019).
URL: <https://pypi.org/project/docplex/>
- IBM ILOG CPLEX Optimization Studio* (2019).
URL: <https://www.ibm.com/products/ilog-cplex-optimization-studio>
- Jia, H., Li, Y., Dong, B. and Ya, H. (2013), 'An Improved Tabu Search Approach to Vehicle Routing Problem', *Procedia - Social and Behavioral Sciences* **96**(Cictp), 1208–1217.
- Jung, S. and Haghani, A. (2000), 'Genetic Algorithm for a Pickup and Delivery Problem with Time Windows', *Transportation Research Record: Journal of the Transportation Research Board* **1733**(1), 1–7.
- Klabjan, D., Johnson, E. L., Nemhauser, G. L., Gelman, E. and Ramaswamy, S. (2001), 'Solving large airline crew scheduling problems: Random pairing generation and strong branching', *Computational Optimization and Applications* **20**(1), 73–91.
- Klabjan, D., Johnson, E. L., Nemhauser, G. L., Gelman, E. and Ramaswamy, S. (2002), 'Airline Crew Scheduling with Time Windows and Plane-Count Constraints', *Transportation Science* **36**(3), 337–348.
- Knowles, J. (2010), 'Multiobjective Optimization', (May), 193–262.
- Labombarda, P. and Nicoletti, B. (1971), Aircraft Rotations by Computer, in 'AGIFORS'.
- Laporte, G. (1992), 'The traveling salesman problem: An overview of exact and approximate algorithms', *European Journal of Operational Research* **59**(2), 231–247.
- Laporte, G. (2009), 'Fifty Years of Vehicle Routing', *Transportation Science* **43**(4), 408–416.
- Laporte, G., Gendreau, M. and Potvin, J.-y. (2000), 'Classical and modern heuristics for the vehicle routing problem', *Intl. Trans. in Op. Res.* **7**, 285–300.
- Laporte, G. and Nobert, Y. (1983), 'A Branch and Bound Algorithm for the Capacitated Vehicle Routing Problem', *OR Spektrum* **5**(m), 77–85.
- Laporte, G. and Osman, I. H. (1995), 'Routing problems: A bibliography', **61**(JANUARY 1995), 227–262.

- Letchford, A. N. and Salazar-González, J. J. (2015), 'Stronger multi-commodity flow formulations of the Capacitated Vehicle Routing Problem', *European Journal of Operational Research* **244**(3), 730–738.
- Lin, C., Choy, K. L., Ho, G. T., Chung, S. H. and Lam, H. Y. (2014), 'Survey of Green Vehicle Routing Problem: Past and future trends', *Expert Systems with Applications* **41**(4 PART 1), 1118–1138.
- Lin, S. and Kernighan, B. (1973), 'An Effective Heuristic Algorithm for the Traveling-Salesman Problem', *Operations Research* **21**(2), 498–516.
- Lohatepanont, M. and Barnhart, C. (2004), 'Airline Schedule Planning: Integrated Models and Algorithms for Schedule Design and Fleet Assignment', *Transportation Science* **38**(1), 19–32.
- Marinakakis, Y. (2012), 'Multiple phase neighborhood search-GRASP for the capacitated vehicle routing problem', *Expert Systems with Applications* **39**(8), 6807–6815.
- Miller, C. E., Tucker, A. W. and Zemlin, R. A. (1960), 'Integer Programming Formulation of Traveling Salesman Problems', *Journal of the ACM* **7**(4), 326–329.
- Osman, I. H. (1991), *Metastrategy : simulated annealing and tabu search for combinatorial optimization problems.*, PhD thesis, Imperial College of Science and Technology and Medicine.
- Papadakos, N. (2009), 'Integrated airline scheduling', *Computers & Operations Research* **36**(1), 176–195.
- Parragh, S. N., Doerner, K. F. and Hartl, R. F. (2008), 'A survey on pickup and delivery problems. Part II: Transportation between pickup and delivery locations', *Journal für Betriebswirtschaft* **58**(2), 81–117.
- Pereira, F. B. and Tavares, J. (2009), *Bio-inspired Algorithms for the Vehicle Routing Problem*, Vol. 161 of *Studies in Computational Intelligence*, 161 edn, Springer Berlin Heidelberg, Berlin, Heidelberg.
- Prins, C. (2009), 'A GRASP X Evolutionary Local Search Hybrid for the Vehicle Routing Problem', *Bio-inspired algorithms for the Vehicle Routing Problem* **161**, 35–53.
- Python (2019).
URL: <https://www.python.org>
- Raff, S. (1983), 'Routing and scheduling of vehicles and crews. The state of the art', *Computers and Operations Research* **10**(2).
- Rego, C. (1998), 'A Subpath Ejection Method for the Vehicle Routing Problem', *Management Science* **44**(10), 1447–1459.
- Rego, C. (2001), 'Node-ejection chains for the vehicle routing problem: Sequential and parallel algorithms', *Parallel Computing* **27**(3), 201–222.
- Reimann, M., Doerner, K. and Hartl, R. F. (2004), 'D-ants: Savings based ants divide and conquer the vehicle routing problem', *Computers and Operations Research* **31**(4), 563–591.
- Rexing, B., Barnhart, C., Kniker, T., Jarrah, A. and Krishnamurthy, N. (2000), 'Airline Fleet Assignment with Time Windows', *Transportation Science* **34**(1), 1–20.
- Richter, H. (1968), 'Optimal Aircraft Rotations Based on Optimal Flight Timing'.
- Rushmeier, R. A. and Kontogiorgis, S. A. (1997), 'Advances in the Optimization of Airline Fleet Assignment', *Transportation Science* **31**(2), 159–169.
- Sherali, H. D., Bish, E. K. and Zhu, X. (2006), 'Airline fleet assignment concepts, models, and algorithms', *European Journal of Operational Research* **172**(1), 1–30.
- Siarry, P. (2016), *Metaheuristics*, Springer International Publishing AG.
- Simpson, R. W. (1966), 'Computerized schedule construction for an airline transportation system', *Department of Aeronautics and Astronautics,*
- Simpson, R. W. (1969), 'Scheduling and Routing Models for Airline Systems'.

- Solomon, M. M. and Desrosiers, J. (1988), 'Survey Paper—Time Window Constrained Routing and Scheduling Problems', *Transportation Science* **22**(1), 1–13.
- Subramanian, R., Scheff, R. P., Quillinan, J. D., Wiper, D. S. and Marsten, R. E. (1994), 'Coldstart: Fleet Assignment at Delta Air Lines', *Interfaces* **24**(1), 104–120.
- Taillard, É. (1993), 'Parallel iterative search methods for vehicle routing problems', *Networks* **23**(8), 661–673.
- Talluri, K. T. (1998), 'The Four-Day Aircraft Maintenance Routing Problem', *Transportation Science* **32**(1), 43–53.
- The Gurobi Python Interface for Python Users* (2019).
URL: https://www.gurobi.com/documentation/8.1/quickstart_mac/the_gurobi_python_interfac.html
- Toth, P. and Vigo, D. (2002), *The vehicle routing problem*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.
- Toth, P. and Vigo, D. (2003), 'The Granular Tabu Search and Its Application to the Vehicle-Routing Problem', *INFORMS Journal on Computing* **15**(4), 333–346.
- Wang, C. H. and Lu, J. Z. (2009), 'A hybrid genetic algorithm that optimizes capacitated vehicle routing problems', *Expert Systems with Applications* **36**(2 PART 2), 2921–2936.
- Wark, P. and Holt, J. (1994), 'A Repeated Matching Heuristic for the Vehicle Routeing Problem', *The Journal of the Operational Research Society* **45**(10), 1156–1167.
- WFP (2017a), 'WFP Aviation Annual Review 2017', (March).
- WFP (2017b), 'WFP Strategic Plan'.
URL: https://docs.wfp.org/api/documents/WFP-0000019573/download/?_ga=2.236942023.79184737.1518904482-139700886.1518904482
- WFP (2018a), 'WFP Aviation Operational Snapshot', (August).
- WFP (2018b), 'World Food Programme'.
URL: <http://www1.wfp.org/unhas> [Accessed on: 2018-09-27]
- Xu, J. and Kelly, J. P. (1996), 'A Network Flow-Based Tabu Search Heuristic for the Vehicle Routing Problem', *Transportation Science* **30**(4), 379–393.