

UNIVERSITEIT GENT

FACULTEIT ECONOMIE EN BEDRIJFSKUNDE

ACADEMIEJAAR 2008 – 2009

Fleet Assignment Problem in de Luchtvaartsector: Korte termijn optimalisaties bij Brussels Airlines

Masterproef voorgedragen tot het bekomen van de graad van

Master in de Toegepaste Economische Wetenschappen: Handel ingenieur

Andrew Fecheyrs-Lippens

onder leiding van

Prof. Dr. Mario Vanhoucke en Dr. Broos Maenhout

Permission

Ondergetekende verklaart dat de inhoud van deze masterproef mag geraadpleegd en/of gereproduceerd worden, mits bronvermelding.

Andrew Fecheyr Lippens

Woord vooraf

Graag wil ik Brussels Airlines en in het bijzonder de heer Lyssens en mevrouw Massart bedanken voor de verkregen informatie en voor het geven van inzicht in hun werkwijze. Zonder hen zou de case study niet tot stand gekomen zijn.

Promotor Prof. Broos Maenhout verdient een dankwoord voor zijn vertrouwen en gedeelde kennis inzake operationeel onderzoek en het modelleren van problemen.

Een speciaal dankwoord voor Melanie die me altijd blijft steunen en motiveren, ook als ik daardoor minder tijd voor haar heb. In het bijzonder wil ik ze bedanken voor het nalezen van de scriptie.

Mijn moeder en Bruce wil ik bedanken voor de verkregen hulp. Bruce in het bijzonder omdat hij me blijft motiveren om de lat hoger te leggen. Zijn scriptie over personeelsplanning in de luchtvaartsector was een bron van inspiratie. Mijn vader verdient een extra dankwoord voor het delen van zijn ervaringen en enthousiasme in de luchtvaart en voor het navragen van informatie bij zijn collega's.

Medestudent Tom Wouters wil ik bedanken omdat hij tijd vrijmaakte om mijn vragen te beantwoorden en richtlijnen te geven wanneer ik die nodig had, niettegenstaande het feit dat hij ook een scriptie te schrijven had.

Ook Alec Detournay wil ik bedanken voor het nalezen van deze scriptie.

Mei 2009,

Andrew Fecheyr-Lippens

Inhoudstafel

TODO

I. INLEIDING

We leven in een wereld waar we afhankelijk geworden zijn van goedkoop transport. Vliegen naar verre bestemmingen tegen een betaalbare prijs vinden we vanzelfsprekend. Door de jaren heen hebben meer en meer mensen de mogelijkheid gekregen om te reizen en ze willen graag zo snel mogelijk hun bestemming bereiken. De economie is geglobaliseerd in een wereldeconomie waardoor zakenlui en bedrijfsleiders dagelijks gebruik moeten maken van de mobiliteit die de luchtvaart hen biedt. Beide trends hebben voor een explosie in de luchtvaartsector gezorgd en liggen aan de basis van het ontstaan van lagekostenmaatschappijen. De komst van die bedrijven en de stagnerende groei hebben echter de luchtvaartmaatschappijen onder druk gezet en gedwongen om efficiënter te gaan werken.

Door het streven naar efficiëntie is de luchtvaartsector een complex geheel geworden dat bestaat uit talloze deelgebieden. Al deze deelgebieden moeten zo goed mogelijk op elkaar worden afgestemd. Door de operationele complexiteit van de sector is de vraag naar beslissingsondersteunende instrumenten hoger dan ooit. Bovendien staat, door de economische crisis en de stijgende brandstofprijzen, de winstgevendheid van de luchtvaartmaatschappijen recentelijk nog meer onder druk, waardoor het noodzakelijk is dat ze streven naar efficiëntie en kostenreductie.

Luchtvaartmaatschappijen trachten hun werking zo goed mogelijk te plannen en te optimaliseren. Omdat er bij het plannen van hun operaties zo veel komt kijken en de complexiteit te hoog wordt als men alles tezamen tracht te optimaliseren, verkiest men om de domeinen één voor één te plannen. Door het sequentieel oplossen van elk planningsprobleem verliest men weliswaar een deel van de nauwkeurigheid, maar dit zorgt er voor dat de problemen handelbaar blijven.

Fleet assignment is slechts één deeldomein van de hele planning. Het betreft het zo optimaal mogelijk toewijzen van de beschikbare vliegtuigen aan de geplande vluchten en is het onderwerp van deze scriptie. We onderzoeken of er significante verbeteringen mogelijk zijn door de fleet assignment op korte termijn te heroptimaliseren. De werkwijze, resultaten en conclusies van het onderzoek bevinden zich in hoofdstuk 3. We zullen aantonen dat een kostenreductie van 2,38% haalbaar is door gebruik te maken van een eigen ontwikkeld model en dat hiermee tot ruim een half miljoen euro per maand bespaard kan worden bij Brussels Airlines.

De scriptie begint met een overzicht van de planningsdomeinen in de luchtvaartsector en licht de relevante deeldomeinen toe. In hoofdstuk 2 bespreken we de twee basismodellen voor fleet assignment uit de literatuur zodat de lezer een inzicht krijgt in het optimalisatieprobleem en de case study van hoofdstuk 3 kan situeren.

Bij deze scriptie hoort een cd-rom met digitaal materiaal waarnaar verwezen zal worden in de tekst. Indien de cd-rom ontbreekt en u graag het materiaal wilt raadplegen kunt u dit navragen door een email te verzenden naar andrew@fecheyr.be.

In het kader van deze scriptie wordt eerst een kort overzicht gegeven van de relevante planningsdomeinen in de luchtvaartsector. Dit overzicht is gebaseerd op Fecheyr-Lippens B. (2007)

I.I. Planning in de luchtvaartsector

De groei van het luchtverkeer, de sterke concurrentie in de luchtvaartsector en de vraag naar meer comfort vanwege de passagiers hebben de complexiteit en het belang van optimalisatie in alle planningsonderdelen sterk doen toenemen (Dorndorf et al, 2007). In de luchtvaartsector wordt gebruik gemaakt van een breed aantal resources of middelen zoals vluchten, terminals, vliegtuigen, personeel en bagage. De beslissingen omtrent het gebruik van deze middelen beïnvloeden elkaar. Dit maakt van planning in de luchtvaartsector een gesofisticeerd geheel dat makkelijker geoptimaliseerd kan worden als het onderdeel per onderdeel benaderd wordt.

De planning in de luchtvaartsector kan opgesplitst worden in een aantal hoofddomeinen: de vluchtplanning, de vliegtuigplanning, de personeelsplanning, het herplannen van storingen, het plannen van de landingen, het toewijzen van de standplaatsen of gate's en het bepalen van een adequate strategie met het oog op maximale ticketinkomsten. Fleet Assignment is slechts één van de vele planningsdomeinen die luchtvaartmaatschappijen trachten te optimaliseren. Daarnaast bestaan een aantal kleine planningsdomeinen zoals het optimaliseren van de wachttijden, de bagageplanning, het laden van de cargo, het schoonmaken van de vliegtuigen, enzovoort. De belangrijkste planningsdomeinen die luchtvaartmaatschappijen moeten optimaliseren worden in figuur I.I geïllustreerd.

Voor een gedetailleerde beschrijving van elk domein verwijzen we graag naar "Personeelsplanning in de luchtvaartsector: Case Brussels Airlines" (Fecheyr-Lippens B., 2007). Hieronder worden kort de eerste vier hoofddomeinen beschreven, omdat die van belang zijn voor het vervolg van deze scriptie.



Fig 1.1. Overzicht van de verschillende planningsdomeinen in de luchtvaartsector

Vluchtplanning

De planning in de luchtvaartsector begint bij het plannen van de vluchten. Deze worden gepland op basis van het geraamde marktaandeel van de luchtvaartmaatschappij en in een vluchtrooster of timetable neergeschreven, inclusief het vertrek- en aankomstuur van elke vlucht. Het doel van het plannen van de vluchten is zoveel mogelijk passagiers aantrekken door aantrekkelijke bestemmingen aan te bieden op geschikte tijdstippen. De uitdaging voor de luchtvaartmaatschappijen zit hem in het inschatten van hun toekomstig marktaandeel. Teveel vluchten plannen naar dezelfde stad zorgt ervoor dat de vliegtuigen niet gevuld zullen raken. Plant men er te weinig dan verliest de luchtvaartmaatschappij potentiële passagiers en inkomsten.

In de literatuur werd reeds veel geschreven over optimalisaties van vluchtroosters. Lee et al. (2007) ontwikkelden met behulp van simulatie een genetisch algoritme voor het bekomen van robuuste roosters. Jacobs et al. (2005) bestudeerden de vluchtplanning bij KLM.

Vliegtuigplanning

Vliegtuigplanning bestaat uit “Fleet Assignment” en “Aircraft maintenance routing”. Figuur 1.2 illustreert het fleet assignment probleem: rechts op de figuur wordt een lijst van vloottypes afgebeeld en links een lijst van vluchten.

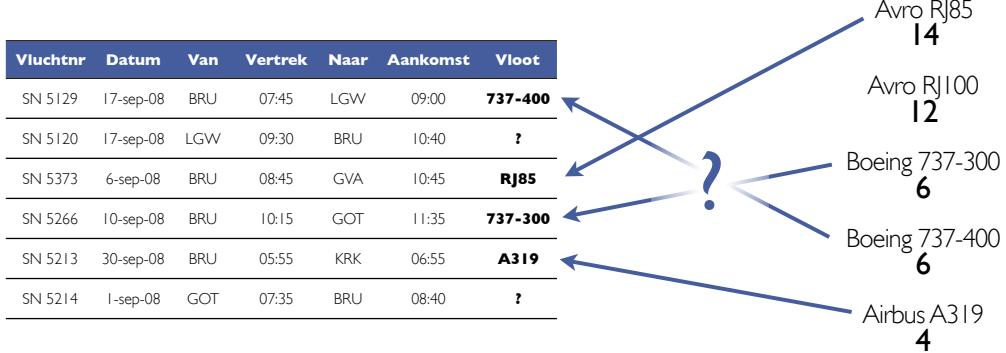


Fig 1.2. Fleet assignment is het toewijzen van vloottypes aan een vluchtrooster

Het Fleet Assignment Problem bestaat erin voor een gegeven vluchtrooster of *flight schedule* te bepalen welk vlootype toegewezen moet worden aan elke *flight leg*, met als doel de totaal verwachte winst te maximaliseren. De aircraft routing en beschikbaarheidsrestricties moeten uiteraard gehandhaafd blijven. De winst van een *flight leg* is een functie van de passagiersvraag, de capaciteit van het toegewezen vliegtuig en de operationele kosten van dit vliegtuig. De keuze van het vlootype voor een *flight leg* wordt voornamelijk bepaald door de benodigde capaciteit (de verwachte passagiersvraag). Fleet assignment vormt de kern van deze scriptie en wordt in hoofdstuk 2 uitvoerig behandeld.

Aircraft maintenance routing vormt het tweede deelprobleem en behandelt het onderhoud van de vliegtuigen en de planning hiervan. Het doel is het verzekeren van voldoende onderhoudssessies zodat het vliegtuig conform de normen vliegveilig is (Sherali et al., 2006).

Personeelsplanning

'Personeelsplanning in de luchtvaartsector' is het onderwerp van Fecheyr-Lippens B (2007). Zijn scriptie legde tevens de basis voor deze scriptie dus citeren we hier zijn omschrijving:

"Operationele onderzoekers noemen dit optimalisatieprobleem het 'Airline Crew Scheduling' probleem. De probleemstelling die zo optimaal mogelijk moet worden opgelost bestaat uit twee stappen. Eerst moeten strategisch gekozen vluchtsequenties, pairings genaamd, worden gegenereerd. Daarna volgt de toewijzingsprocedure. In het bijzonder moet het personeel toegewezen worden aan deze pairings en aan andere activiteiten die de maatschappij doet, en moeten individuele werkroosters worden opgesteld. De twee stappen worden respectievelijk crew pairing en crew rostering genoemd. Het belangrijkste doel dat crew scheduling nastreeft, vormt het genereren van een efficiënte personeelsplanning die de kosten minimaliseert, terwijl de set van beperkingen wordt gerespecteerd." (Fecheyr-Lippens B., 2007 p1).

Revenue management

Revenue management ook "Yield management" genoemd kan naar het Nederlands vertaald worden als inkomstenmanagement of "Strategie voor maximale ticketinkomsten". Deze managementvorm vindt haar oorsprong in de luchtvaartsector en betreft het begrijpen, anticiperen en beïnvloeden van het consumentengedrag met als doel het maximaliseren van de inkomsten voor een vast en vergankelijk product. De uitdaging is het verkopen van het juiste product aan de juiste passagier op het juiste tijdstip aan de juiste prijs. Ticketinkomsten zijn de grootste inkomstenbron van een luchtvaartmaatschappij en vormen daarom een belangrijk domein voor het management. Door de deregulering van 1979 heeft het inkomstenmanagement fors aan belang gewonnen doordat luchtvaartmaatschappijen variabele prijzen mogen aanrekenen voor een identieke product. Prijsdiscriminatie door de luchtvaartmaatschappijen werd legaal.

Passagiers kunnen typisch worden ingedeeld in twee segmenten: business reizigers, die flexibiliteit en annuleringsopties eisen, laattijdig hun vlucht boeken maar ook bereid zijn een hogere te betalen, en vrijetidsreizigers, die goedkope tickets verkiezen en doorgaans vroegtijdig boeken. Figuur 1.3 illustreert beide segmenten.

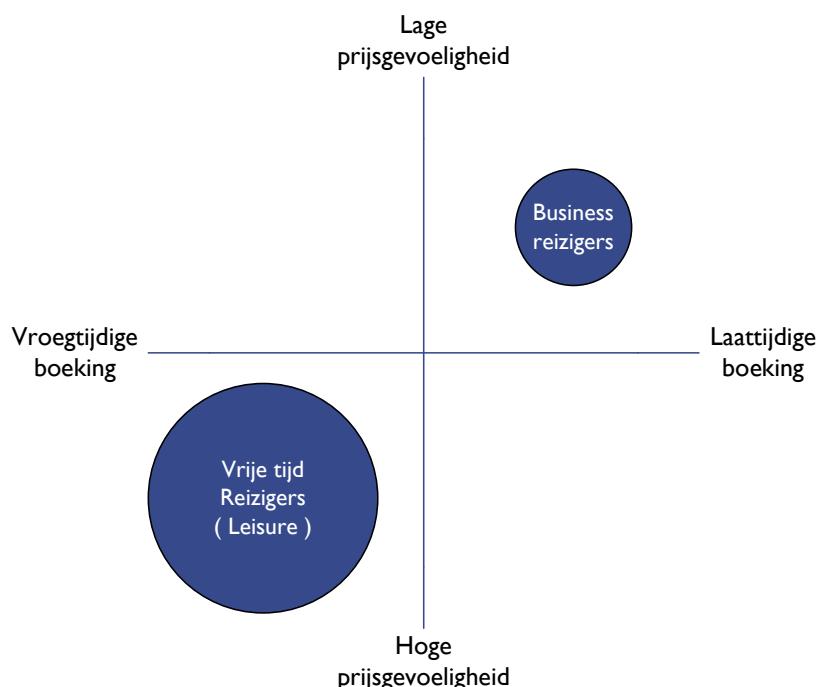


Fig 1.3. Segmentering van passagiers.¹

¹ overgenomen van Fechey-Lippens B (2007) p10

Luchtvaartmaatschappijen kunnen drie strategieën kiezen voor het aanpakken van deze segmenten. Ofwel focussen ze zich op één doelgroep en positioneren ze zich volledig naar deze doelgroep, zoals Ryanair doet voor vrijetijdsreizigers. Ofwel wordt gekozen voor een combinatiestrategie, waarbij prijsdifferentiatie centraal staat. Deze laatste strategie wordt onder andere door Brussels Airlines gevolgd met hun b.flex en b.light segmenten. De uitdaging zit hem in het bepalen hoeveel zetels men lang op voorhand wilt verkopen aan een lagere prijs, wetend dat een aantal laattijdige reizigers bereid zijn een hogere prijs te betalen. De prijszetting vormt dus een cruciaal aspect en daarom is het interessant te weten dat vele luchtvaartmaatschappijen de prijs in functie stellen van het aantal dagen voor vertrek, het huidige boekingspercentage, de voorspelling van de totale vraag, de dag van de week en het uur van de dag (Gosavi et al., 2007).

Een volledig overzicht van de literatuur wordt gegeven door McGill and van Ryzin (1999). Boyd and Bilegan (2003) presenteren een overzicht van de geschiedenis van revenue management. Andere relevante papers over dit domein zijn te vinden in Bertsimas and de Boer (2005), Chen et al. (2003), Karaesmen en van Ryzin (2004). Darius Walczak en Shelby Brumelle (2006) stellen in hun artikel "Semi-Markov information model for revenue management and dynamic pricing" een framework voor rond traditionele yield management en dynamische prijszetting.

Voor een bespreking van de overige planningsdomeinen in de luchtvaartsector verwijzen we graag naar Fechey-Lippens B. (2007). In het volgende hoofdstuk kijken we welke fleet assignment modellen de literatuur te bieden heeft en verkennen we het fleet assignment problem.

2. FLEET ASSIGNMENT

We beginnen de verkenning van het fleet assignment problem door eerst enkele termen te definiëren en bespreken daarna twee basismodellen uit de literatuur: het model van Abara (1989) en dat van Hane et al. (1995).

2.1. Terminologie

Flight leg: Individuele vlucht van plaats A naar plaats B zonder tussenstop, zoals een rechtstreekse vlucht van Brussel naar Madrid. De terugvlucht van Madrid naar Brussel wordt gezien als een nieuwe flight leg. Alle flight legs staan in de flight schedule of het vluchtrooster.

Vloot: De set vliegtuigen van hetzelfde type van een luchtvaartmaatschappij. Als men bijvoorbeeld spreekt over de Airbus A319 vloot van Brussels Airlines, dan bedoelt men de hele groep A319 vliegtuigen van die maatschappij.

Vloottype (*vliegtuigtype*): Het specifieke model van een vliegtuig, zoals een Boeing 737-300. Alle vliegtuigen van hetzelfde type hebben dezelfde cockpit configuratie, crew kwalificatie vereisten, onderhoudsvereisten en capaciteit.

Rotatietijd: De minimale tijd dat een vliegtuig nodig heeft tussen de landing en de volgende take-off. De rotatietijd hangt af van het vliegtuigtype en de luchthaven, en ligt rond de 30 a 40 minuten voor Europese vluchten. Vliegtuigen voor lange afstanden hebben een langere rotatietijd. Dit komt onder andere doordat er meer brandstof getankt wordt, er meer passagiers en bagage aan boord zijn, etc. Lagekostenmaatschappijen hechten veel belang aan het verlagen van de rotatietijden omdat één vliegtuig hierdoor ingezet kan worden om meerdere vluchten per dag te vliegen.

2.2. Overzicht van bestaande modellen

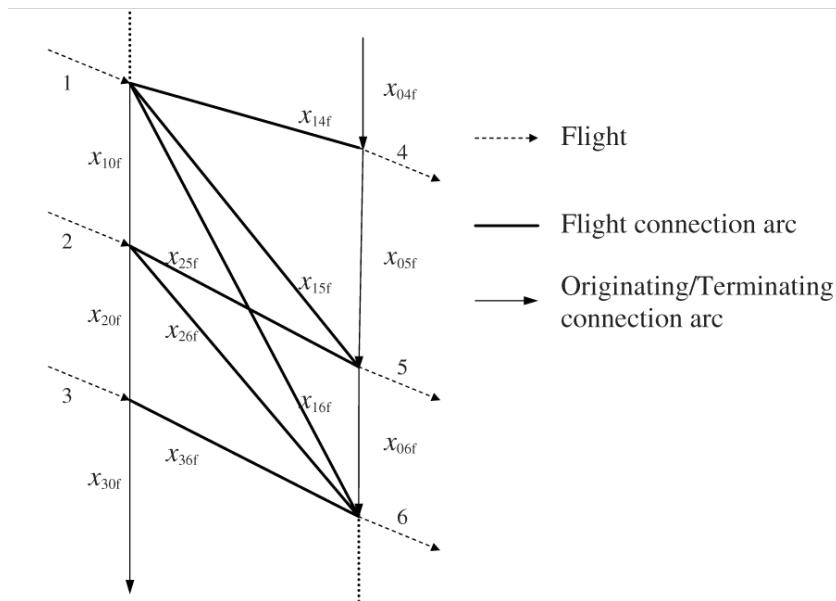
Het volgende overzicht is voornamelijk gebaseerd op Sherali et al (2006).

Het fleet assignment probleem wordt vaak geformuleerd als een mixed-integer probleem, gebaseerd op het netwerk van een luchtvaartmaatschappij. Men kan het netwerk voorstellen op twee manieren: door *arcs* connecties te laten voorstellen (*connectie netwerken*) of door *arcs* flight legs te laten voorstellen (*tijd-ruimte netwerken*). Beide manieren van representatie verzekeren dat het model voldoet aan de volgende hoofdrestricties: (1) *cover constraints*: zodat elke flight leg wordt toegewezen aan exact één vloot. (2) *balance constraints*: voor het voldoen aan de flow. (3) *vliegtuig beschikbaarheidsconstraints* waarbij het aantal beschikbare vliegtuigen het gebruik ervan beperkt. Door de verschillende interpretaties van *arcs* in beide netwerk representaties zijn de mathematische formuleringen van de restricties verschillend.

Fleet Assignment Model met een connectie netwerk (FAM)

In 1989 was Abara één van de eerste onderzoekers die het fleet assignment probleem aanpakte door gebruik te maken van een *connectie netwerk*. In dit netwerk stellen de knooppunten (*nodes*) punten in de tijd voor wanneer vluchten aankomen of vertrekken. Er wordt ook gebruik gemaakt van twee denkbeeldige knooppunten: een '*master source*' knooppunt en een '*master sink*'

knooppunt om begin-van-de-dag en einde-van-de-dag effecten op te vangen. Drie soorten *arcs* worden gebruikt om verschillende soorten connecties voor te stellen: *vlucht connecties* linken de aankomst- en vertrekknooppunten, *termination arcs* linken de aankomstknooppunten met de master sink om vliegtuigen voor te stellen die op een station blijven tot het einde van de dag, en *originating arcs* linken de master source met de vertrekknooppunten om vliegtuigen voor te stellen die aan het begin van de dag vertrekken van een station. Alle connecties moeten haalbaar zijn, dit wil zeggen dat de vertrek- en aankomsttijden moeten gerespecteerd worden rekening houdend met de rotatietijd van het vliegtuig. In Abara's model komen de binaire beslissingsvariabelen overeen met de verschillende vloottypes die deze connectie kunnen uitvoeren. Fig 2.1 illustreert het connectie netwerk met 12 connecties voor één station (6 vlucht connecties, 3 terminating arcs en 3 originating arcs)



Pijlen 1-6 stellen vluchten voor en behoren niet tot de beslissingsvariabelen. De binaire "connectie" variabelen x_{ijf} zijn de beslissingsvariabelen en tonen aan dat vlootype f wordt ingezet voor de connectie i naar j. x_{iof} stellen terminating arcs voor en x_{ofj} originating arcs.

Fig 2.1. Connectie netwerk van 1 luchthaven met 12 haalbare connecties²

Vervolgens zullen we wiskundige notaties gebruiken om Abara's model te formuleren. L is de set van flight legs, geïndexeerd met i en j , F is de set van vlootypes, geïndexeerd met f . De set van stations wordt aangeduid met het symbool S , met index s . Overigens stellen we de set aankomende legs en vertrekende legs voor station s voor door respectievelijk A_s en D_s ($s \in S$). We

² Abara (1989) p21

definiëren x_{ijf} als binaire beslissingsvariabele die de waarde 1 aanneemt als de connectie van leg $i \in L$ naar leg $j \in L$ gevlogen wordt door de vloot $f \in F$, zo niet dan neemt de beslissingsvariabele de waarde 0 aan. De indices $i = 0$ en $j = 0$ verwijzen respectievelijk naar de *originating* en *terminating* arcs. Het doel van dit model is het maximaliseren van de opbrengsten verminderd met de operationele kosten.

Het voordeel van vloot f te gebruiken voor een connectie wordt arbitrair bepaald door het voordeel van de vertrekkende vlucht j van de connectie, en wordt genoteerd als p_{jf} . p_{jf} is een combinatie van winst, vliegtuig *utilization*, etc. Abara (1989) gebruikte een nominale operationele kost, aangeduid door c , voor elke toewijzing van het type x_{0jf} die het gebruik van vloot f voor flight leg j betekent. Deze kostenfactor c kan echter ook afhankelijk gemaakt worden van de specifieke vlucht en vloot, en berekend worden door het combineren van de operationele kosten met de kosten van 'verloren' passagiers. In de luchtvaartsector worden verschillende berekeningswijzen gebruikt voor het berekenen van de kosten.

De wiskundige formulering van Abara's model is als volgt.

Model #1: Abara's fleet assignment model

$$\text{Maximize} \quad \sum_{i \in L \cup \{0\}} \sum_{j \in L} \sum_{f \in F} p_{jf} x_{ijf} - c \sum_{j \in L} \sum_{f \in F} x_{0jf} \quad (\text{Ia})$$

$$\text{subject to} \quad \sum_{i \in L \cup \{0\}} \sum_{f \in F} x_{ijf} = 1 \quad \forall j \in L, \quad (\text{Ib})$$

$$\sum_{i \in L \cup \{0\}} x_{ilf} - \sum_{j \in L \cup \{0\}} x_{ljf} = 0 \quad \forall l \in L, f \in F, \quad (\text{Ic})$$

$$\sum_{i \in D_s} x_{0if} - \sum_{j \in A_s} x_{i0f} = 0 \quad \forall s \in S, f \in F, \quad (\text{Id})$$

$$\sum_{i \in L} x_{0if} \leq A_f \quad \forall f \in F, \quad (\text{Ie})$$

$$x \text{ binair.} \quad (\text{If})$$

In dit model zorgt de cover constraint (Ib) er voor dat elke vlucht voorafgegaan wordt door een aankomst of een *originating* arc. De balance constraint (Ic) verzekert de *flow balance* van elke leg voor elke vloot, i.e. dat er voor elke vloot f evenveel connecties toekomen bij leg l als er

vertrekken. Er is een tweede balance constraint (1d) die zorgt dat de *schedule balance* wordt nageleefd: er moeten evenveel vliegtuigen van elk type eindigen op elk station als er beginnen. De laatste constraint (1e) verzekert dan weer dat er niet meer vliegtuigen van elke vloot worden ingezet dan er beschikbaar zijn.

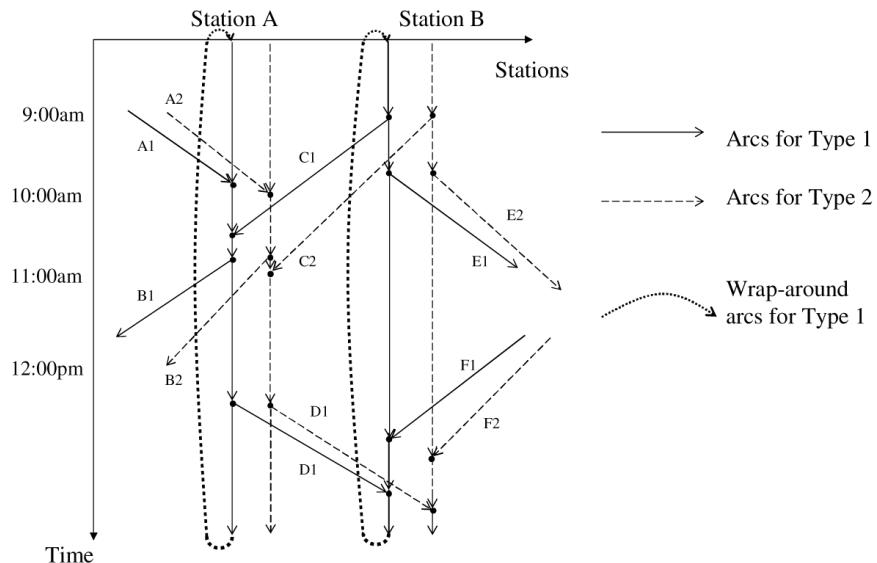
Het grote nadeel van Abara's model is dat alle mogelijke (haalbare) connecties van een netwerk moeten worden gespecificeerd. Het model groeit echter snel door het grote aantal mogelijke vluchtconnecties en wordt praktisch onhandelbaar voor grote netwerken. Abara (1989) probeerde dit op te lossen door de connectie variabelen per vlucht te beperken. Zodoende worden slechts een maximaal aantal connecties per vlucht overwogen, wat de kwaliteit van de oplossing niet ten goede komt.

Fleet Assignment Model met een tijd-ruimte netwerk (FAM2)

In tegenstelling tot connectie netwerken richten *tijd-ruimte* netwerken zich op het voorstellen van flight legs. Het model kiest zelf welke connecties gemaakt worden. Deze structuur maakt het model niet alleen intuïtiever, het verlaagt ook het aantal beslissingsvariabelen aanzienlijk omdat het aantal flight legs vele malen kleiner is dan het aantal mogelijke connecties. Berge en Hopperstad (1993) en Hane et al. (1995) waren één van de eerste onderzoekers die gebruik maakten van tijd-ruimte netwerken om het fleet assignment probleem voor te stellen en hebben er mede voor gezorgd dat deze structuur tegenwoordig de voorkeur geniet bij het formuleren van dit probleem.

Bij tijd-ruimte netwerken is het zo dat voor elke vloot een apart subnetwerk wordt opgebouwd. Op die manier kan men gebruik maken van verschillende vlucht- en rotatietijden per vloot. Als de vlucht- en rotatietijden niet significant verschillen tussen specifieke vloottypes, kan er een samengesteld netwerk opgebouwd worden voor deze types. In het netwerk van elk vlootype wordt elke vluchtaankomst en vertrek op een specifiek tijdstip gekoppeld aan een knooppunt (node). Om er voor te zorgen dat vluchtconnecties haalbaar zijn, worden de aankomstknooppunten (*arrival nodes*) geplaatst wanneer de vliegtuigen klaar zijn om opnieuw te vertrekken, d.w.z. dat het aankomsttijdstip wordt vermeerderd met de rotatietijd van het vliegtuig. Het vliegtuig is vanaf dat tijdstip klaar om opnieuw op te stijgen. Er zijn drie soorten *arcs*: *flight arcs* die flight legs voorstellen, *ground arcs* die vliegtuigen voorstellen die voor een gegeven tijdsinterval aan de grond blijven op een luchthaven en *wrap-around arcs* die de laatste knooppunten van de dag verbinden met de eerste van de dag. Bij *same daily schedule* zorgt dit voor de continuïteit van het Fleet Assignment. Bij deze vorm van netwerkrepresentatie worden luchthavens geduplicateerd: 1 kopij voor elk vlootype.

In het subnetwerk van elk gegeven vloottype kan men een tijdslijn aanmaken voor elke luchthaven, die bestaat uit een serie knooppunten die na elkaar in de tijd plaatsgrijpen. Een grafische voorstelling hiervan vindt u terug op Fig 2.2.



De schuine pijlen zijn *vlucht arcs*, de verticale pijlen zijn *ground arcs* en de gebogen pijlen zijn *wrap-around arcs*. (Om de duidelijkheid van het diagram te behouden werden de wrap-around arcs voor vloottype 2 weggelaten) De arc paren (A_1, A_2) tot (F_1, F_2) stellen vluchten A - F voor, gevlogen door respectievelijk vloottypes 1 en 2. In dit diagram hebben de vliegtuigen van vloottype 2 een langere rotatietijd en zijn hierdoor de aankomsttijden later dan die van vloottype 1.

Fig 2.2. Netwerk tijdslijn van 2 luchthavens in een same daily schedule³

Figuur 2.2. toont twee luchthavens in een tijd-ruimte netwerk voor twee vloottypes. De tijd verloopt volgens de verticale as waarbij elk knooppunt een vluchtaankomst of vertrek voorstelt. De volle pijlen behoren tot het netwerk van vloottype 1 terwijl de gestreepte pijlen tot het netwerk van type 2 behoren. Zo stelt het begin van pijl C_1 het vertrek van vlucht C op de luchthaven B voor vloottype 1 voor, en het uiteinde van deze pijl stelt de aankomst op luchthaven A voor.

In de formulering van dit model wordt de flow op elke flight arc gerepresenteerd door een binaire beslissingsvariabele x_{fi} met de restrictie dat de overeenstemmende flight leg slechts door één vloottype wordt gevlogen. De flows voor de ground arcs en wrap-around arcs nemen echter gehele waarden aan. Deze waarden stellen het aantal vliegtuigen voor die gedurende de tijd van de

³ Sheralli et al. (2006) p4

arc aan de grond blijven op de gegeven luchthaven. Zoals in het vorige model worden we hier opnieuw geconfronteerd met de drie hoofdrestricties: *cover*, *balance* en *beschikbaarheid*. Ook wordt er een specifieke tijdslijn gedefinieerd voor het tellen van vliegtuigen. Voor elk vloottype worden de flows over alle ground en flight arcs die deze tijdslijn passeren gesommeerd om te verzekeren dat het totaal aantal vliegtuigen van elk type in gebruik, het beschikbare aantal niet overtreft. De *balance* restrictie garandeert vervolgens dat deze beschikbaarheid voor elk tijdstip voldaan wordt.

Hieronder volgt de wiskunde formulering van het model van Hane et al. (1995)

Gebruikte symbolen⁴

S	set van luchthavens in het netwerk, geïndexeerd door s, o of d
F	set van vloottypes, geïndexeerd door f
L	set van flight legs, geïndexeerd door l of $\{odt\}$, met $o,d \in S$ en t de vertrektijd van de vlucht van luchthaven o of wanneer het vliegtuig klaar is om te vertrekken van luchthaven d
N	set van knooppunten in het netwerk, geïndexeerd door $\{fst\}$, met $f \in F, s \in S$ en t het tijdstip
$O(f)$	set van arcs voor vloottype f die de vliegtuig count time-line passeren, $f \in F$
c_{fl}	de kost om vloot f de flight leg l te laten uitvoeren, $f \in F$ en $l \in L$
A_f	aantal beschikbare vliegtuigen voor vloottype $f, f \in F$
$x_{fl} =$	1 , als vloottype f de flight leg l uitvoert, $f \in F$ en $l \in L$ 0 , indien niet (de beslissingsvariabele x_{fl} kan ook genoteerd worden als x_{fst} voor $f \in F, \{odt\} \in L$)
$y_{fst'}$	doorstroming van vliegtuigen op de ground arc van knooppunt $\{fst\} \in N$ naar knooppunt $\{fst'\} \in N$ op luchthaven $s \in S$ in het netwerk van vloottype f
t^-, t^+	de tijd in de tijdslijn respectievelijk voor en na t .

De tijdslijn waarmee de vliegtuigen geteld worden, wordt gebruikt als startpunt voor een reeks gebeurtenissen in het netwerk. De eerste set knooppunten na deze tijdslijn worden aangeduid met de notatie $\{fst_1\}, f \in F, s \in S$ en de laatste met de notatie $\{fst_n\}, f \in F, s \in S$.

Model #2: Basic FAM met een tijd-ruimte netwerk

$$\text{Minimize} \quad \sum_{l \in L} \sum_{f \in F} c_{fl} x_{fl} \quad (2a)$$

4 Deze notatie werd letterlijk overgenomen van Sherali et al (2006) p7

$$\text{Subject to} \quad \sum_{f \in F} x_{fl} = 1 \quad \forall l \in L, \quad (2b)$$

$$\sum_{o \in S} x_{fst} + y_{fst^-t} - \sum_{d \in S} x_{fsdt} - y_{fst^+t} = 0 \quad \forall \{fst\} \in N, \quad (2c)$$

$$\sum_{l \in O(f)} x_{fl} + \sum_{s \in S} y_{fst_n t_1} \leq A_f \quad \forall f \in F, \quad (2d)$$

$$x \text{ binair}, y \geq 0 \quad (2e)$$

Men vindt hier dezelfde drie basis restricties terug, weliswaar in een andere gedaante dan bij het vorige model: de cover constraints (2b), de balance constraints (2c) en de vliegtuig beschikbaarheidsconstraints (2d). De cover constraint verzekert dat elke vlucht aan exact één vloottype toegewezen wordt en de balance constraint zorgt er voor dat de doorstroming vóór het knooppunt gelijk is aan die ná het knooppunt. Tenslotte telt de beschikbaarheidsconstraint het aantal vliegtuigen van elk vloottype en zorgt er voor dat er geen vliegtuigen worden ingezet die niet beschikbaar zijn.

De doelfunctie bevat bij Hane et al. (1995), in tegenstelling tot Abara (1989), enkel een kostenfactor voor elke (vlucht, vloot) combinatie. In deze kostenfactor wordt enerzijds een *spill* kost en anderzijds een operationele kost in rekening gebracht. De verwachte passagiersvraag minder de capaciteit van het vloottype wordt *spill* genoemd. Passagiers die niet mee kunnen op een vlucht als gevolg van de vlootkeuze zijn een aanzienlijke kost voor een luchtvaartmaatschappij. Deze passagiers kunnen bediend worden door een volgende vlucht van dezelfde luchtvaartmaatschappij, men spreekt van *recaptured* passagiers, of door een vlucht van een concurrerende maatschappij, waarbij de mogelijke omzet voorgoed verloren gaat. Met een gegeven *recapture rate*, prijsstructuur, beschikbare zetels en verwachte vraag kan men een kost voor *spilled* passagiers berekenen.

De operationele kosten van een assignment zijn mogelijks nog moeilijker te bepalen. De brandstofkosten en geschatte bemanningskosten zijn nog enigsins makkelijk te bepalen, de kost om een vloottype te introduceren in een luchthaven waar deze nog niet vliegt is daarentegen een stuk complexer om te schatten. Die schatting bevat namelijk veel vaste componenten, zoals training of relocatie van onderhoudspersoneel, inventaris van reserveonderdelen en wettelijke regelingen.

Dit model kan, als het gebruikt wordt met honderden luchthavens en duizenden vluchten, zeer omvangrijk en moeilijk te optimaliseren zijn. Gu et al. (1994) kwamen tot dezelfde conclusie

toen ze aantonden dat dit probleem, zelfs zonder de beschikbaarheidsrestricties, NP-hard is voor drie vlootypes. Mede hierdoor stelden Hane et al. (1995) voor om een aantal verwerkingsstappen uit te voeren voordat het model wordt toegepast, met als doel het verminderen van de grootte van het netwerk.

De eerste voorverwerkingsstap is gebaseerd op de vaststelling dat het exacte tijdstip van elk knooppunt onbelangrijk is, zolang het netwerk de juiste verbindingen voorstelt. Het doel van de tijdslijn is er voor zorgen dat de vliegtuigbalans klopt en dat enkel haalbare connecties worden uitgevoerd. Voor dit model maakt het niet uit of een vliegtuig om 10:00 of 10:05 aankomt, zo lang er geen vertrek plaatsvindt tussen deze twee tijden. Met dit inzicht in het achterhoofd is het makkelijk om een groot aantal overbodige knooppunten samen te nemen. De knooppunten van opeenvolgende aankomsten gevolgd door opeenvolgende vertrekken kunnen samengenomen worden tot één knooppunt, zonder de oplossing van het model te beïnvloeden. Fig 2.3. illustreert deze aggregatie van knooppunten. De linkse tijdslijn toont alle oorspronkelijke knooppunten en de middelste toont een tijdslijn waarbij de knooppunten (B,C) en (D,E,F,G) worden samengenomen. Deze techniek wordt knooppuntaggregatie genoemd. Een gelijkaardige techniek werd eerder ook door Berge en Hopperstad (1993) gebruikt.

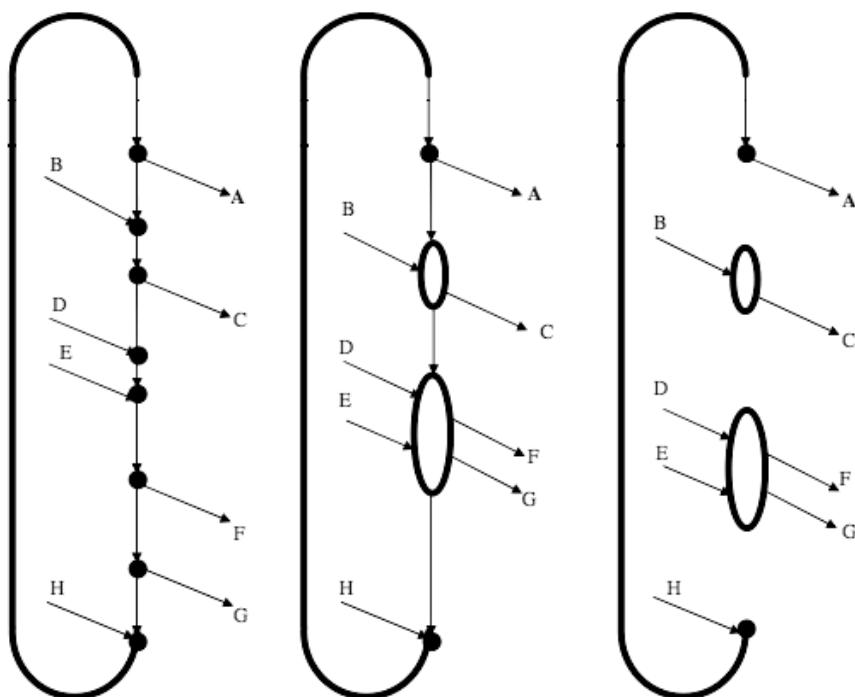


Fig 2.3. Knooppuntaggregatie en isoleren van eilanden.⁵

⁵ Sherali et al. (2006) p9

De tweede voorverwerkingsstap is het verwijderen van ground arcs waar geen vliegtuigen doorploegen. Het idee is dat in “hub and spoke” netwerken bij de “spoke” luchthavens vaak periodes zijn waar er geen vliegtuigen aan de grond wachten. Deze ground-arcs met nul doorstroming kunnen verwijderd worden uit het model. Er ontstaan “eilanden”. We verwijzen naar Hane et al. (1995) voor de motivatie en uitwerking van deze techniek. Men komt met deze techniek tot een verzameling tijdslijnen zoals die er in het rechtse deel van figuur 2.3 uitzien.

Samen met een derde voorverwerkingsstap, het verwijderen van gemiste verbindingen, slaagden Hane et al. (1995) er in om de grootte van een typerend probleem te verminderen van 48.982 rijen en 66.942 kolommen naar 7.703 rijen en 20.464 kolommen. Ze verbeterden ook het oplossingsalgoritme met een aantal technieken en brachten zo de totaal benodigde cpu tijd aanzienlijk lager tot een werkbaar niveau voor omvangrijke netwerken.

De toepassing van beide fleet assignment modellen heeft een enorme impact gehad op de luchtvaartsector. Abara's (1989) implementatie zorgden voor een 1,4% verbetering van de operationele kostenmarge voor American Airlines. Rushmeier en Kontogiorgis (1997) boekten een jaarlijks voordeel van ten minste \$15 miljoen voor US Airways. De technieken van Hane et al. (1995) worden breed toegepast in de sector.

3. CASE: BRUSSELS AIRLINES

Na een overzicht te hebben gegeven van planning in de luchtvaartsector in het algemeen en twee basis fleet assignment modellen uit de literatuur te hebben besproken is het nu tijd om de case study aan te vangen. We beginnen met een besprekking van de werkwijze bij Brussels Airlines. Daarna lichten we de gevuldte methode van dataverzameling toe. Tenslotte behandelen we de aanpassingen aan het model van Hane et al. (1995), leggen onze oplossingsmethode uit en interpreteren de resultaten.

Alle informatie van Brussels Airlines die volgt komt voort uit een interview met de heer David Lyssens te Zaventem op 21 november 2009 en een tweede interview met mevrouw Daniella Massart en de heer David Lyssens op 30 april 2009.

3.1. Fleet Assignment bij Brussels Airlines

Het fleet assignment probleem wordt doorgaans op lange termijn opgelost, vaak langer dan 3 maanden voor vertrekdatum. In de literatuur wordt gewerkt met een gekende passagiersvraag gebaseerd op geraamde marktaandelen van de routes die de luchtvaartmaatschappij aanbiedt. Het aantal passagiers per flight leg wordt als gegeven beschouwd. Om die passagiersvraag te kunnen bepalen beschikken grote luchtvaartmaatschappijen over forecastingssystemen of kopen ze deze informatie aan bij onderzoeksgebureaus.

Brussels Airlines vindt de prijs die de onderzoeksgebureaus aanrekenen te hoog, en koopt de marktinformation niet. Door de relatief jonge leeftijd van de luchtvaartmaatschappij en het feit dat de markt een aantal tumultueuze jaren achter de rug heeft, beschikt Brussels Airlines niet over een nauwkeurig forecastingssysteem. De vraag van de passagiers naar routes aangeboden door Brussels Airlines verandert nog te veel om betrouwbaar te kunnen voorspellen. Als gevolg hiervan moet Brussels Airlines vaak bijsturen wanneer ze merkt dat een route meer of minder gegeerd is dan verwacht. Er wordt dan manueel gezocht naar swappingmogelijkheden waarbij een vliegtuig met een hogere capaciteit en een lage bezettingsgraad gewisseld wordt met een vliegtuig met een kleinere capaciteit dat overbezeten is.

De huidige werkwijze van Brussels Airlines

Bij Brussels Airlines zijn er drie departementen die een invloed uitoefenen op de fleet assignment. Deze departementen zijn Revenue Management, Capacity Management en Crew Planning. Het Revenue Management departement houdt zich voornamelijk bezig met de verkoop van vliegtuigtickets. Het analyseert de trends in de markt, houdt de competitie in de gaten en zoekt naar evenementen die de vraag naar bepaalde bestemmingen kunnen beïnvloeden. Het is ook verantwoordelijk voor het maken van ruwe forecasts voor de passagiersvraag.

De taak van het Capacity Management departement is het toewijzen van de beschikbare vloottypes aan de geplande vluchten. Het departement is verantwoordelijk voor het swappen van vliegtuigen tussen vluchten en speelt de grootste rol bij het oplossen van het fleet assignment probleem.

Het derde departement, Crew Planning, probeert met de opgegeven planning van Revenue Management en Capacity Management het beschikbare personeel te verdelen over alle vluchten en toestellen. Dit departement heeft een beperkende rol in het fleet assignment probleem en zorgt voor een belangrijke rigiditeit in de planning naarmate men dichter bij de vertrekdatum komt.

Het oplossen van het fleet assignment probleem begint bij Brussels Airlines ongeveer zes maanden op voorhand. Men maakt de planning op van een *typische* week voor het volgende seizoen. Rond de maand maart maakt men bijvoorbeeld de planning van zo'n typische week voor het komende najaar. Men houdt hierbij slechts rekening met twee grote groepen vliegtuigen: de short-haul en de medium-haul vloottypes. Dit vereenvoudigt het probleem en behoudt een zekere algemeenheid.

Wanneer men dichter dan zes maanden voor vertrek komt, zullen Capacity Management en Revenue Management de *typische* week opdelen naar een day-to-day planning en kopiëren over de hele maand. Deze planning wordt ook op niveau van de vloot gedetailleerde: elke vlucht wordt nu aan een specifiek vloottype toegewezen.

Drie maanden voor vertrek wordt de eerste draft opgemaakt van de fleet assignment ter voorbereiding van de slotfilings bij IATA, de International Air Transport Association. Na het onderhandelen van de slots wordt de tweede draft opgemaakt. Er wordt nu gekeken naar het aantal boekingen die bij elke vlucht staan en eventuele bijsturingen worden uitgevoerd, want wanneer men dichter dan drie maand voor vertrek komt, begint het Crew Planning departement met de *crewing*. Dit zorgt er voor dat Capacity Management veel van zijn bewegingsvrijheid voor het wisselen van vloottypes verliest omdat het voor elke wissel de toestemming moet krijgen van Crew Planning.

Zes weken voor vertrek worden de crewroosters opgemaakt. Op dat moment worden de individuele bemanningsleden van Brussels Airlines toegewezen aan de vluchten. Omdat Brussels Airlines van de Belgische wetgeving en de sociale partners weinig flexibiliteit krijgt met betrekking tot het inzetten van het personeel, wordt het nu een stuk moeilijker om veranderingen in de fleet assignment door te voeren. Dit in tegenstelling tot wat er in andere landen soms gangbaar is. In Spanje bijvoorbeeld is het bij luchtvaartmaatschappij Iberia de gewoonte dat men tot 1 maand voor vertrek de crew toewijzing nog aanpast en kan men de fleet assignment dus finetunen met de verkregen boekingsinformatie.

Figuur 3.1 toont de tijdslijn van de fleet assignment bij Brussels Airlines. De drie kritische momenten die we zojuist besproken hebben staan er op afgebeeld.

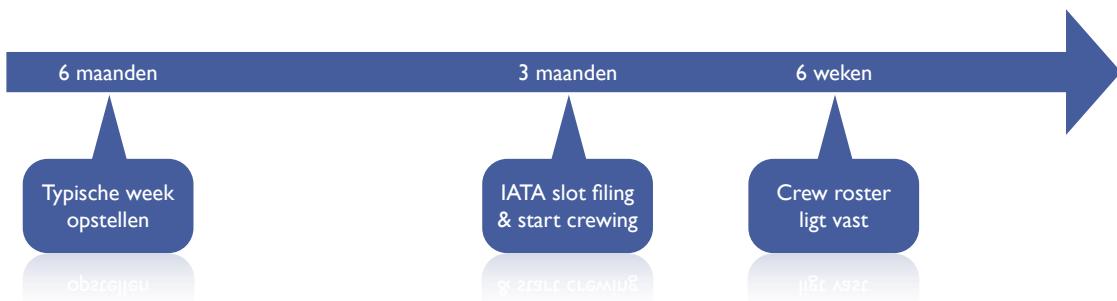


Fig 3.1. Planningshorizon voor de fleet assignment bij Brussels Airlines

Voor het toewijzen van een vloottype aan een vlucht kijkt Brussels Airlines onder andere naar de types die door de concurrenten worden ingezet en naar de samenstelling van de passagiers (business - vrijetijd) van de specifieke route. Sommige vloottypes bieden namelijk meer ruimte aan de passagiers en dit extra comfort wordt des te meer op prijs gesteld door business reizigers.

Opportuniteit voor optimalisatie

De manuele werkwijze van Brussels Airlines biedt een fantastische opportuniteit voor optimalisatie. De standaard fleet assignment modellen uitgelegd in hoofdstuk 2 vereisen een gekende passagiersvraag per fight leg, informatie waarover Brussels Airlines niet beschikt.

Wanneer men dichter bij de vertrekdag van een vlucht komt, beschikt Brussels Airlines echter over het aantal boekingen voor die vlucht. Men kan namelijk op elk moment zien hoeveel zetels er al verkocht zijn per vlucht. Bijgevolg is er een opportuniteit om met de gegevens beschikbaar op korte termijn, een heroptimalisatie uit te voeren in de fleet assignment. Het is deze opportuniteit waar we op inspelen in het vervolg van deze case study.

We hebben, in samenspraak met Brussels Airlines, de tijdshorizon voor onze optimalisatie vastgelegd op één maand voor vertrekdag. Aangezien de flight schedule een wekelijkse cyclus volgt, kiezen we vier weken als tijdshorizon.

3.2. Dataverzameling

Voor we ons model presenteren, willen we eerst de methode van dataverzameling en onze bevindingen hierover toelichten. De formulering van het model werd namelijk sterk beïnvloed door de data die beschikbaar waren.

Van Brussels Airlines kregen we, naast alle mondeling overgedragen informatie, twee datasets ter beschikking. De eerste bevat de verkoophistoriek van alle vluchten uit een bepaalde, drukke maand. Hierin staat voor elke vlucht het aantal verkochte zetels genoteerd per dag voor de zes maanden voorafgaand aan het vertrek van de vlucht. De confidentialiteit van deze gegevens is belangrijk voor Brussels Airlines en dus werd de echte maand en de bestemming van de vluchten gemaskeerd. De maand werd in de dataset 'september 2008' genoemd, we nemen die benaming over voor het vervolg van de case study.

De tweede dataset is de flight schedule die bij de verkoopgegevens hoort. Om de confidentialiteit te behouden werd het tijdstip van elke vlucht verschoven met een constant aantal minuten. Deze dataset wordt in detail besproken op p27.

De vloot van Brussels Airlines⁶

Brussels Airlines vliegt met verschillende types toestellen naar bestemmingen in Europa en Afrika. De gehele vloot bestaat uit 7 types vliegtuigen die hieronder kort beschreven worden per "familie". Een vliegtuigfamilie is een groep gelijkaardige vliegtuigtypes die dezelfde crew kwalificatie vereisen en vaak ook dezelfde ground handling procedures hebben. Het voordeel hiervan is dat bemanningsleden op alle types vliegtuigen binnen eenzelfde familie ingezet kunnen worden en de voorzieningen op luchthavens gedeeld worden.

Avro RJ85 – AVRO RJ100 – BAe 146

De Europese bestemmingen worden meestal gevlogen met een AVRO RJ85, een AVRO RJ100 of een BAe 146. Brussels Airlines heeft 14 Avro RJ85, 12 AVRO RJ100 en 6 BAe 146 vliegtuigen. Deze bieden respectievelijk 82, 97 en 84 zitplaatsen. Sinds begin 2009 zijn, ten gevolge van de teruglopende vraag, de 6 Britisch Aerospace BAe 146 vliegtuigen op non actief gezet.

De Avro RJ serie is een doorontwikkeling van de BAe 146 met verbeterde motoren en flightdeck instrumentatie. De RJ85 en RJ100 hebben bij maximale load een relatief kort bereik. De



Fig 3.2. Brussels Airlines Avro RJ85
Registratie 00-DJX

⁶ De basis informatie over de vloot van Brussels Airlines komt van <<http://company.brusselsairlines.com/nl/company/fleet.aspx>> (14/11/2008) en de fotos van <<http://www.airliners.net/>>

verste bestemming die door deze vliegtuigen gehaald wordt is Helsinki, op zo'n 1.700km van Brussel. Het voordeel van deze vliegtuigen is dat ze relatief klein en kosten-efficiënt zijn, waardoor vluchten met een lage vraag ook rendabel kunnen zijn.

Airbus A319

Sommige bestemmingen in Europa en het Midden Oosten worden gevlogen met een Airbus A319. Brussels Airlines heeft 4 Airbussen A319 met 132 zitplaatsen.

De Airbus A319 is het kleine broertje van de heel succesvolle A320 en werd ontwikkeld in 1993-1995. Het toestel heeft een groter bereik dan de Avro's en kan daardoor vanuit Brussel alle bestemmingen binnen Europa bedienen en zelfs enkele bestemmingen erbuiten zoals Moskou, Eskisehir en Tel Aviv.



Fig 3.3. Brussels Airlines Airbus A319-112

Registratie 00-SSG

Boeing 737-300 & 737-400

Vele Europese bestemmingen worden verzorgd door 11 Boeings 737-300 en 737-400. Een 737-300 kan 142 passagiers vervoeren en 5 leden cabinepersoneel. Een 737-400 daarentegen zorgt voor het transport van 164 passagiers en 6 leden cabinepersoneel.



Fig 3.4. Brussels Airlines Boeing 737-3M8

Registratie 00-LTM

Constructeur Boeing heeft met zijn 737 de populairste commerciële jet ooit op de markt gebracht. De 737-400 is binnen de 737 familie een variant met een lange cabine en hoge capaciteit. Het bereik en de kruissnelheid van beide varianten zijn vergelijkbaar met de Airbus A319.

Brussels Airlines kreeg deze vliegtuigen in handen na de fusie met Virgin Express in 2007. Sommige toestellen dateren echter al van 1989, toen ze voor het eerst in gebruik werden genomen door Braathens, een Noorse luchtvaartmaatschappij⁷. De ouderdom van de Boeing 737 vloot zorgt voor hoge onderhoudskosten en recentelijk is gebleken dat Brussels Airlines overweegt deze vloot te verkleinen.

⁷ Airfleets.net - Geschiedenis van vliegtuigen <<http://www.airfleets.net/ficheapp/plane-b737-24270.htm>> (09/02/2009)

Airbus A330-300

Bestemmingen in Afrika worden verzorgd door 4 Airbussen 330-300. Een Airbus 330-300 heeft 284 zetels, waarvan 30 in Business Class. Bovendien kan dit type vliegtuig per jaar ongeveer 30.000 ton aan cargo vervoeren.



Fig 3.5. Brussels Airlines Airbus A330-301

Registratie OO-SFN.

De Airbus A330 is Brussels Airlines' enige lange afstand vloot die met zijn maximale bereik van ongeveer 8.000 km⁸ 284 passagiers kan vervoeren naar 14 bestemmingen in Afrika. De A330 is het grootste vliegtuig van Airbus met 2 motoren.

In de tabel hieronder is een overzicht opgenomen van de vloot van Brussels Airlines.

Type (f)	Aantal (Af)	Passagiers (CAPf)	Haul
AVRO RJ85	14	82	Short
AVRO RJ100	12	97	Short
BAe 146	6	84	Short
Boeing 737-300	6	142	Medium
Boeing 737-400	6	164	Medium
Airbus A319	4	132	Medium
Airbus A330-300	4	284	Long

Overzicht van de 51 vliegtuigen van Brussels Airlines begin 2009 met passagiers capaciteit. De kolom "Haul" staat voor het bereik van de vloottype en wordt later besproken.

Tabel 3.1. De vloot van Brussels Airlines

In de tijdsperiode van de gebruikte data waren er slechts drie BAe 146 vliegtuigen actief bij Brussels Airlines. Omdat ze met een systeem van spares werken en er altijd één of twee Avro of BAe vliegtuigen in onderhoud zijn, werd ons voorgesteld om er voor het onderzoek van uit te gaan dat er geen BAe 146 vliegtuigen beschikbaar zijn.

8 Airliners.net Aircraft Data <<http://www.airliners.net/aircraft-data/stats.main?id=23>> (16/11/2008)

Flight schedule

Brussels Airlines bezorgde ons een lijst met alle vluchten van één gemaskeerde maand. Per vlucht werd het vluchtnummer, de vertrek- en aankomstluchthaven, de vertrek- en aankomsttijd en het vloottype waarmee de vlucht werd gevlogen gegeven. Ter illustratie staan in de onderstaande tabel twee flight legs uit het flight schedule van Brussels Airlines. Om de confidentialiteit van de gegevens te bewaren maskeerde Brussels Airlines de bestemmingen en werden de vertrek- en aankomsttijden verschoven met een constant aantal minuten.

Vluchtnummer	Van	Vertrektijd	Naar	Aankomsttijd	vloottype
SN 5146	Dest16	07:30	BRU	08:40	Avro RJ85
SN 5269	BRU	14:30	Dest27	15:50	Boeing 737-300

Tabel 3.2. Voorbeeld met 2 vluchten uit het flight schedule

Wat opvalt bij het onderzoeken van het vluchtrooster is dat elke flight leg uitgevoerd door Brussels Airlines altijd ofwel vanuit Zaventem vertrekt ofwel in Zaventem aankomt. Dit betekent dat de short- en medium-haul vluchten van Brussels Airlines een perfect "hub and spoke" principe volgen met slechts één hub: Zaventem. Met andere woorden: er zijn geen flight legs tussen de bestemmingen onderling. Het "hub and spoke" principe wordt geïllustreerd in figuur 3.6 waarin Zaventem centraal als enige hub en een aantal bestemmingen (de "spokes") als zwarte punten afgebeeld staan.

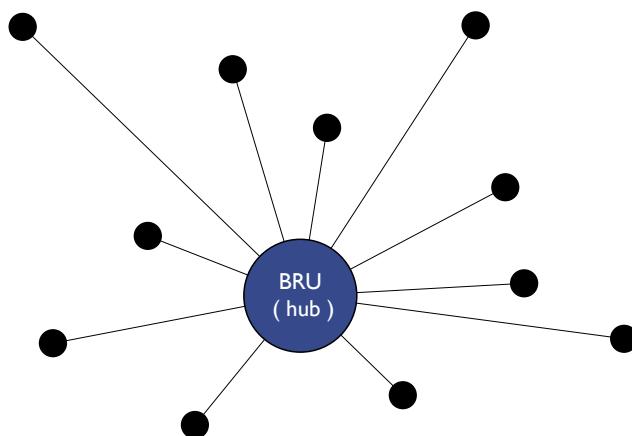


Fig 3.6. "Hub and spoke" principe, met één enkele hub.

Naast het feit dat Brussels Airlines een perfect "hub and spoke" principe met één hub hanteert hebben we nog iets vastgesteld: Wanneer een toestel naar een bestemming vertrekt, zal ditzelfde toestel altijd de eerstvolgende terugvlucht van die bestemming vliegen. Dit heeft als

gevolg dat er in de "spoke" luchthavens nooit meer dan één vliegtuig van Brussels Airlines aan de grond staat en laat ons toe de individuele flight legs van en naar een bestemming samen te nemen in één heen- en terugvlucht. Beide principes openen een aantal deuren voor vereenvoudiging van het model van Hane et al (1995) die we later bespreken.

Range restrictions

Niet alle vliegtuigtypes zijn voorzien om grote afstanden af te leggen. De maximale afstand die een vliegtuig kan afleggen als het volledig geladen is noemt men het bereik of de *range* van dat vliegtuig. Omdat Brussels Airlines vloottypes met verschillende bereiken heeft, vormt dit een nieuwe reeks restricties voor het model. Om deze restricties te vereenvoudigen kan men de vloottypes van Brussels Airlines samennemen in groepen. Zo behoren de Avro jets tot de short-haul, de Boeing 737 en de Airbus A319 tot de medium-haul en de Airbus A330 tot de long-haul. Op figuur 3.7 worden alle bestemmingen van Brussels Airlines weergegeven uit een andere flight schedule (van februari 2009). De cirkels tonen het bereik van elke groep.



Fig 3.7. De bestemmingen van Brussels Airlines, met short-,medium- en long-haul cirkels.⁹

Op de figuur is te zien dat een short-haul vloottype, zoals de Avro RJ100, niet tot in Tel Aviv geraakt en dat enkel de long-haul A330 de Afrikaanse bestemmingen kan bedienen. Omdat de A330 vloot nooit ingezet wordt op medium- of short-haul bestemmingen, is het opportuun om deze vloot en de bestemmingen in Afrika weg te laten voor het vervolg van deze case study. Dit

⁹ Voor deze figuur werd een ongemaakte flight schedule gebruikt met de echte IATA codes van de luchthavens. De geografische coördinaten werden van <<http://gc.xls2.com>> gehaald en weergegeven op een kaart met behulp van ammap <<http://www.ammap.com>> - zie code in bijlage 5.9

verlaagt het aantal flight legs aanzienlijk en vermindert het aantal vloottypes, waardoor de complexiteit van het model daalt.

In de gemaakte flight schedule van Brussels Airlines is het haul type per flight leg niet inbegrepen. Om die informatie te achterhalen zijn we als volgt te werk gegaan. Brussels Airlines heeft ons verteld dat de eerste vlucht die met de short-haul vloottypes (de Avro's) gevlogen wordt Brussel - Helsinki is. Uit een andere ongemakkeerde flight schedule weten we dat deze vlucht 2u55 duurt. Men kan er dus van uitgaan dat elke vlucht die langer duurt dan 2u55 een medium-haul vlucht is, en dat deze niet gevlogen kan worden door de Avro's. We controleerden deze veronderstelling door te kijken of de lange routes in de verkoopgeschiedenis ooit door een Avro gevlogen werden. Dit bleek niet het geval te zijn dus we kunnen er van uitgaan dat onze assumptie klopt.

Kosten informatie

Om het Fleet Assignment model van Hane et al. (1995) te kunnen toepassen is het ook van belang de kostenfactoren van elk vloottype te kennen. Brussels Airlines was echter enkel bereid om de relatieve verschillen in kosten tussen de types mee te delen. Deze in de tabel 3.3 opgenomen en worden aangeduid met symbool RVK_f en RFK_f . De relatieve vaste en variabele kosten van de Boeing 737-400 werden arbitrair op 100 gezet.

Type (f)	Vaste kost (RVK_f)	Var. kost (RVK_f)
AVRO RJ85	84	83
AVRO RJ100	85	86
BAe 146	84	83
Boeing 737-300	96	94
Boeing 737-400	100	100
Airbus A319	108	100

Tabel 3.3. De vloot van Brussels Airlines met kosten informatie

De absolute kosten voor de vloottypes wou Brussels Airlines niet prijsgeven. Aangezien we toch concrete euro's nodig hebben voor ons onderzoek, besluiten we de kosten van een Boeing 737-400 te schatten. Later kunnen we de sensitiviteit van deze assumptie onderzoeken.

Variabele kosten (VK737-400)		Vaste kosten (FK737-400)	
2 piloten	400 eur / uur	per flight leg	
4 cabine personeel	400 eur / uur	takeoff & landing fees	200 eur
brandstof (2.5 ton/h x 590 eur/ton)	1475 eur / uur	vliegtuig kosten (onderh, leasing,...)	1400 eur
Totale variabele kosten	2375 eur / uur	Totale vase kosten	1600 eur

Tabel 3.4. Geschatte kosten van een Boeing 737-400¹⁰

Met de geschatte kosten uit tabel 3.4 voor een 737-400 en de relatieve verhoudingen die Brussels Airlines ons gaf, kunnen we de kosten van elk vloottype schatten aan de hand van de volgende formules:

$$FK_f = FK_{737-400} * \left(\frac{RFK_f}{100} \right) \quad \text{en} \quad VK_f = VK_{737-400} * \left(\frac{RVK_f}{100} \right)$$

Gebruikte symbolen

- FK_f De vaste kosten van vloottype f per flight leg
- RFK_f De relatieve verhouding tussen de vaste kosten van vloottype f en de 737-400
- VK_f De variabele kosten van vloottype f per vlieguur
- RVK_f De relatieve verhouding tussen de variabele kosten van vloottype f en de 737-400

Rotatietijd

In het model van Hane et al. worden de aankomsttijden vermeerderd met de rotatietijd van het vliegtuigtype om zo de knooppunten van het netwerk op te bouwen. Dit verzekert dat onhaalbare connecties niet gemaakt worden.

Luchthaven	Rotatietijd (ROTx)
BRU	40 min
andere	30-50 min

Tabel 3.5. De gebruikte rotatietijden per luchthaven

¹⁰ De schatting is gebaseerd op cijfers uit een informeel gesprek met een manager bij Brussels Airlines.

De rotatietijd in Zaventem is min of meer gelijk voor alle vloottypes en bedraagt 40 minuten. Andere luchthavens hebben vaak kortere of langere rotatietijden, afhankelijk van het vloottype. Voor ons onderzoek zijn de rotatietijden op externe luchthavens echter niet zo belangrijk. Brussels Airlines wees ons er namelijk op dat voor alle vloottypes, met een extra inspanning van de bemanning, een rotatietijd van 30 minuten mogelijk is. In het model dat we later in dit hoofdstuk zullen presenteren gebruiken we deze grens van 30 minuten voor de rotatie op externe luchthavens en 40 minuten voor Zaventem.

Spill kost per passagier

We gebruiken voor ons model het spill-kost principe uit Hane et al (1995) dat op p13 uitgelegd werd. Spill is de verwachte passagiersvraag van een flight leg min de capaciteit van het vloottype, m.a.w. het aantal “verloren” passagiers. We kiezen er dus voor om deze verloren passagiers als kost te modelleren. Dit in tegenstelling tot winstmaximaliserende modellen waarbij men de omzet van elke betalende passagier in rekening brengt. Het gebruik van het spill-kost principe heeft als voordeel dat we het prijsmechanisme van de revenue management niet moeten analyseren. Voor de duidelijkheid noemen we “spill prijs” de kostprijs per passagier en “spill kost” de totale “spilled” kost van een vlucht.

De spill prijs moet een weerspiegeling zijn van de opportunitetskost die de luchtvaartmaatschappij lijdt en kan berekend worden als men de *recapture rates* kent van elke route. Dit onderzoek valt buiten het bestek van deze scriptie en we moeten dus genoegen nemen met een assumptie over de spill prijs. Brussels Airlines wist ons te vertellen dat de kostprijs van de laatste tickets rond de 200 euro en 300 euro liggen, respectievelijk voor short- en medium-haul routes. Aangezien een deel van de verloren passagiers een plaats zal vinden op een andere vlucht van Brussels Airlines, en dus geen opportunitetskost voorstellen voor de luchtvaartmaatschappij, mogen die niet meegeteld worden in de spill prijs. We besluiten om 50% van de laatste ticket prijs als spill prijs per passagier te nemen.

Haul	Spill prijs (SPv)
Short	100 €
Long	150 €

Tabel 3.6. Spill kost per haul

Verwachte passagiersvraag

Het grote voordeel van een heroptimalisatie op korte termijn is dat er informatie beschikbaar is over het aantal verkochte zetels. Hoe dichter men bij de vertrekdatum van een vlucht komt, des te beter men kan inschatten wat de uiteindelijke vraag zal zijn.

Brussels Airlines stelde de verkoophistoriek van alle 6.391 vluchten uit de flight schedule voor ons beschikbaar. Voor elke vlucht werd het verkocht aantal zetels gegeven op elke dag gedurende zes maanden. Dit gaf ons per vlucht 180 dagen aan nuttige informatie. In totaal bevat de dataset ruim 1,1 miljoen datapunten. We schreven een script in de Ruby programmeertaal om de data in te lezen en op te slaan in een relationele database. Deze opzet vergemakkelijkt het analyseren en maakt het mogelijk om met specifieke SQL querries en de “R” programmeertaal accurate grafieken te genereren, ondanks de vele datapunten.¹¹

Om een beter inzicht te krijgen in de verkoopcijfers van Brussels Airlines staat, bij wijze van voorbeeld, het verkoopverloop voor alle vluchten uitgevoerd met een Avro RJ100 afgebeeld in figuur 3.8. Op de verticale as staat het aantal verkochte zetels en langs de horizontale as is het aantal dagen voor het vertrek van de vlucht af te lezen. De data werd gehaald uit 1602 vluchten, 180 dagen voor het vertrek en bestaat dus uit 288.360 datapunten.

Op figuur 3.8 is te zien dat voor de meeste vluchten de verkoop pas 50 dagen voor vertrek goed op gang komt. Daarvoor zitten de meeste vluchten onder de 20 verkochte zetels. Dit wordt enigszins duidelijker in figuur 3.9, waarop het gemiddelde aantal verkochte zetels per dag wordt weergegeven. Op die grafiek staat het algemene verloop van de verkoop van een vlucht bij Brussels Airlines.

¹¹ Zie bijlage 5.3 voor een gedetailleerdere beschrijving van de gebruikte methode

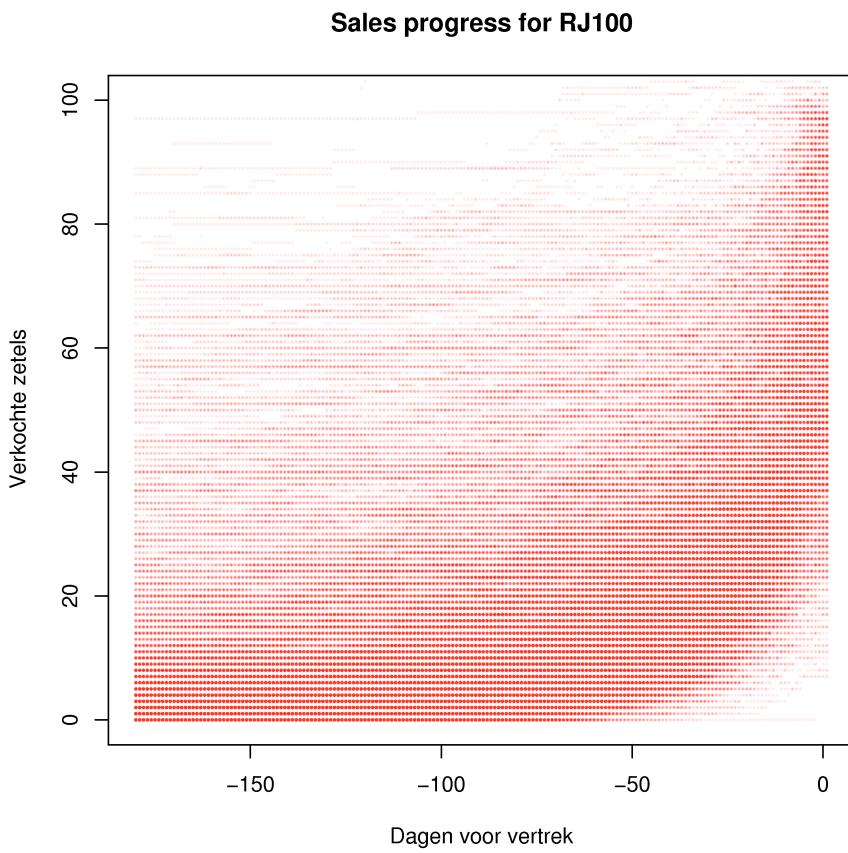


Fig 3.8. Verkoopverloop voor 1602 RJ100 vluchten, 180 dagen voor vertrek.

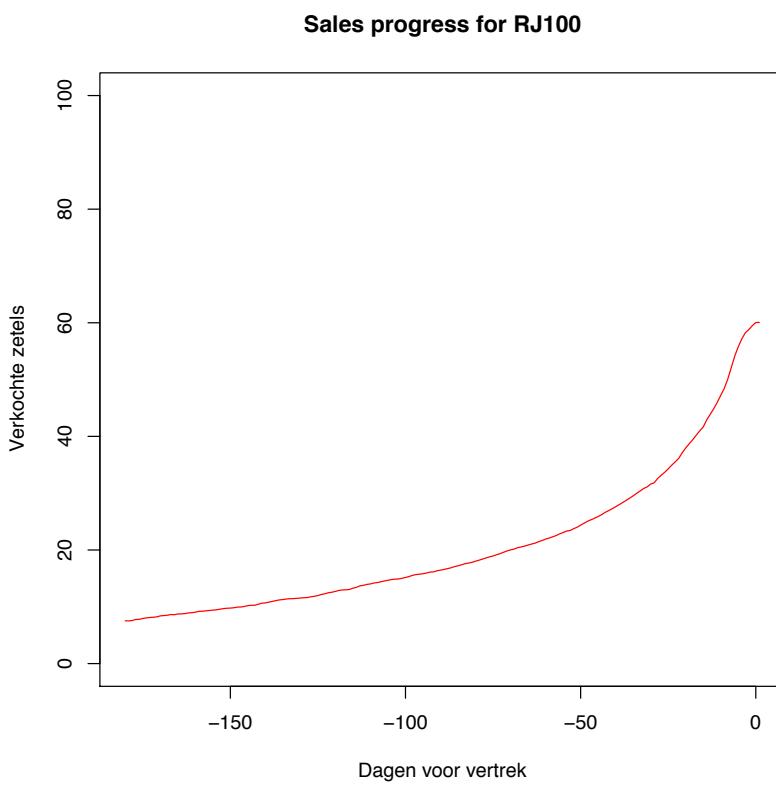


Fig 3.9. Gemiddelde verkoopverloop voor RJ100 vluchten.

Om de passagiersvraag voor ons model te kunnen voorspellen, onderzoeken we in eerste instantie of er een regelmaat voorkomt in de verhouding tussen de verkochte zetels vier weken (28 dagen) voor het vertrek en de verkochte zetels op de dag van het vertrek. We noteren de passagiersvraag x dagen voor vertrek als D_x . D_0 staat dus voor het verkochte aantal zetels op de dag van vertrek, en D_{28} voor het verkochte aantal zetels 28 dagen voor vertrek.

$$X = \frac{D_0}{D_{28}}$$

Om de verhouding representatiever te maken worden 2 groepen vluchten uit de statistieken gefilterd:

1. Vluchten waarbij het aantal verkochte zetels ooit boven de 90% van de toegewezen capaciteit kwam. Bij een hoge bezettingsgraad verhoogt Revenue Management de prijzen van de tickets en beperkt zo de verdere verkoop.
2. Vluchten die te weinig verkochte zetels hebben 28 dagen voor vertrek. De limiet werd gesteld op 10% van de toegewezen capaciteit, zodat onderverkochte vluchten niet meetellen in de analyse.

Uiteindelijk behoren van de 6.391 flight legs er 1.224 tot de eerste groep en 463 tot de tweede groep. Voor de overige 4.704 flight legs werd de verhouding X berekend. Figuur 3.10 toont de histogram van de verhoudingen.

Uit de resultaten kunnen we afleiden dat het grootste deel van de flight legs zich tussen factor 1 en 3 bevindt, maar dat er nog een aantal tussen factor 3 en 4 zitten. Dit is ook te zien in de histogram. Het mag duidelijk zijn dat deze afwijking te groot is om onze voorspellingen op te baseren.

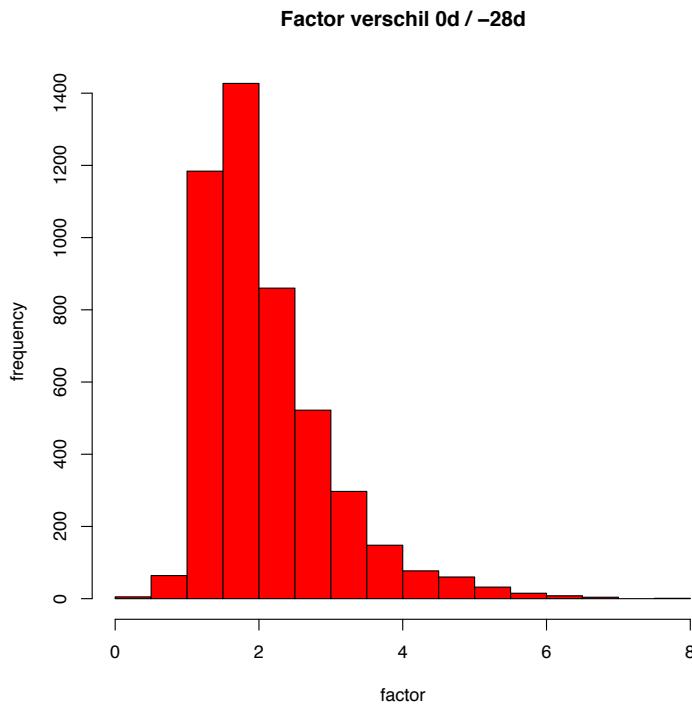


Fig 3.10. Histogram van de verhouding D_0/D_{28} voor 4704 flight legs.

We gaan na of een regressieanalyse geen betere forecasting methode is. We nemen dezelfde dataset als hierboven, met de oververkochte en onderverkochte vluchten weggefilterd en plotten D_{28} op de horizontale as tegenover D_0 . In totaal komen zo 4704 datapunten in de volgende grafiek (figuur 3.11). De lineaire regressieanalyse werd in de “R” statistische programmeertaal berekend met behulp van de code die in bijlage C terug te vinden is.

De formule voor de regressievergelijking is als volgt:

$$D_0 = 0,9526D_{28} + 29,8462$$

Voor deze regressie is $R^2 = 0,67$. We kunnen met deze formule de eindvraag schatten aan de hand van de vraag 28 dagen voor het vertrek, weliswaar met een beperkte nauwkeurigheid.

Een nauwkeurigere forecasting is waarschijnlijk mogelijk als men per bestemming de geschiedenis van de verkopen gedurende een aantal maanden bestudeert. Ook kunnen andere factoren de variatie helpen verklaren, maar dat onderzoeken zou ons buiten het bestek van deze scriptie leiden.

Verkoop analyse

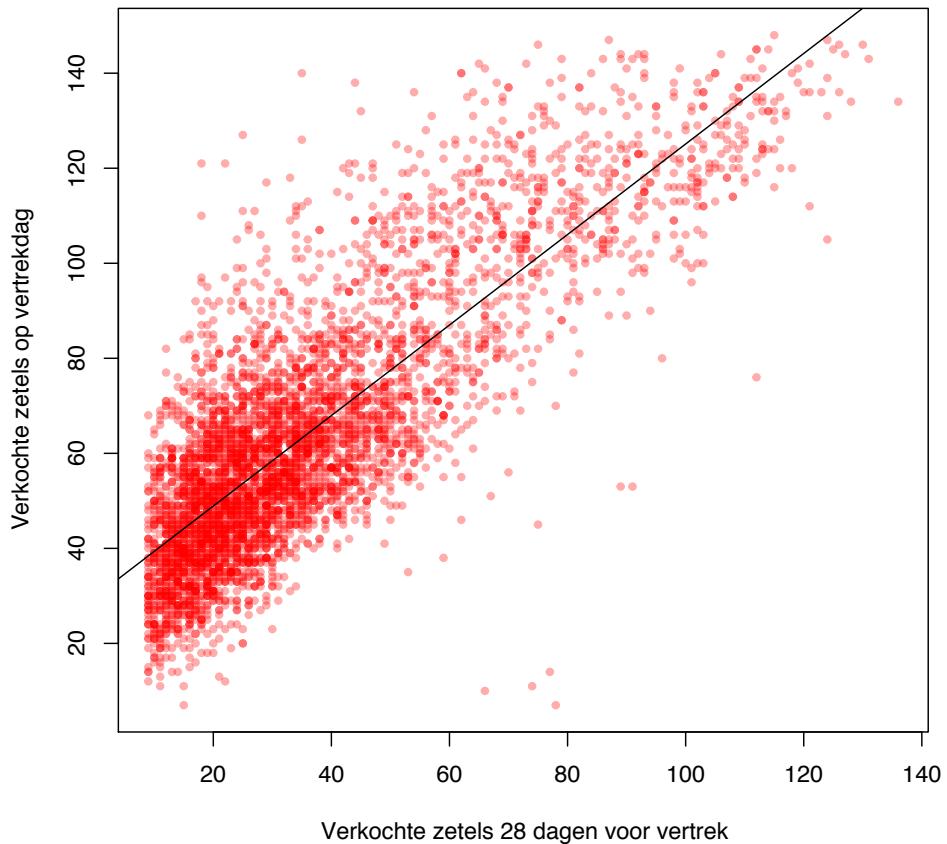


Fig 3.11. Verkochte zetels 28 dagen voor vertrek tov. verkochte zetels op vertrekdag

Strafkost per wissel

Mevrouw Massart van Capacity Management bij Brussels Airlines vertelde ons dat de wissels (swaps) die ze uitvoeren vooral wissels binnen dezelfde vliegtuigfamilie zijn. Ze wisselen voornamelijk Avro RJ100's met RJ85's en Boeing 737-400's met 737-300's. Wisselen binnen dezelfde vliegtuigfamilie heeft als grote voordeel dat dezelfde bemanning kan gebruikt worden als die gescheduled was om de oorspronkelijke vlucht te vliegen.

Als men vliegtuigen wisselt die niet tot dezelfde familie behoren dan moet het Crew Planning departement zoeken of er een bemanning gevonden kan worden voor het nieuwe vloottype. Dit neemt tijd en energie in beslag en vergt ook flexibiliteit van de piloten en het cabinepersoneel. Er hangt echter geen enkele monetaire kost vast aan het wisselen van vloottypes. Mevrouw Massart benadrukte dit sterk.

In de scenario-analyse zullen we kijken naar het effect van een strafkost per wissel (scenario 6), en wat het effect is als we wissels buiten vliegtuigfamilie onmogelijk maken (scenario 7).

Wissel	Strafkost (WP_x)
Binnen familie	50 €
Buiten familie	300 €

Tabel 3.7. Strafkost per wissel voor scenario 6

We hebben de relevante data verzameld en bestudeerd en we hebben inzichten gekregen in de belangrijkste aspecten voor het opstellen van een fleet assignment. In het volgende onderdeel zullen we ons model opbouwen vertrekende van wat we geleerd hebben.

3.3. Aanpassen van het FAM2 voor deze case

De specifieke situatie van de short- en medium-haul van Brussels Airlines, zoals besproken in de vorige delen, opent een aantal mogelijkheden tot vereenvoudiging van het FAM2 model van Hane et al (1995) besproken in hoofdstuk 2. De netwerkrepresentatie kan aangepast worden en we tonen aan dat de extra beslissingsvariabelen y_{fstt} niet langer nodig zijn.

Een eerste vereenvoudiging betreft het aantal netwerken dat opgebouwd moet worden. Omdat er geen significante verschillen in vlucht- en rotatietijden tussen de vlootypes zijn, kunnen we een samengesteld netwerk gebruiken. Werken met één netwerk is vanzelfsprekend makkelijker dan werken met subnetwerken voor elk vlootype.

De tweede vereenvoudiging komt voort uit het perfecte “hub and spoke” principe en het feit dat de individuele flight legs kunnen samengenomen worden als één enkele heen- en terugvlucht. Wanneer een vliegtuig vertrekt van de hub naar een spoke, dan weten we dat dit vliegtuig niet elders gebruikt kan worden totdat het terug land op de hub zoveel uren later. Alle heen- en terugvluchten kunnen we voorstellen met een *away-arc*. Het maakt niet uit waar die vluchten naartoe gaan, van belang is dat het vliegtuig dat wordt ingezet op de *away-arc* onbeschikbaar is om ingezet te worden op andere vluchten. Dit maakt het mogelijk om het hele netwerk te visualiseren in één enkele tijdslijn, namelijk die van de centrale hub. Een voorbeeld hiervan vindt u terug op figuur 3.12. De witte balken stellen *away-arcs* voor, die de heenvlucht en de terugvlucht bevatten. De bestemming van die vluchten maakt voor het model niets uit, maar zijn hier voor de duidelijkheid wel afgebeeld.

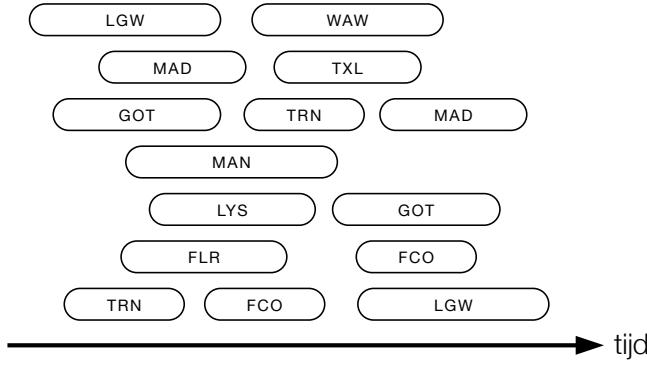


Fig 3.12. Conceptuele tijdslijn van de centrale hub.

Om het ons makkelijk te maken hebben we voor deze case study tijdsintervallen gekozen waar in het begin en op het einde alle vliegtuigen in Zaventem staan. Zo moeten we geen rekening houden met de begin- en eindsituaties.

Ons aangepaste FAM2 model voor een “hub and spoke” netwerk met één hub en samengestelde vluchten wordt hieronder wiskundig geformuleerd. We hebben grotendeels dezelfde symbolen gebruikt als in hoofdstuk 2 maar voeren ook een aantal nieuwe symbolen in.

Nieuwe symbolen

- V set van vluchten, opgebouwd uit de individuele flight legs $l \in L$
- T set van tijdstippen waarop vluchten kunnen vertrekken
- $O(t)$ set van vluchten die het tijdstip t passeren. $t \in T$
- F_S set van short-haul vloottypes (de Avro RJ85, RJ100 en BAe 146 vloten)
- V_M set van medium-haul vluchten

Model #3: FAM voor een h&s netwerk met één hub en samengestelde vluchten

$$\text{Minimize} \quad \sum_{v \in V} \sum_{f \in F} c_{fv} x_{fv} \quad (3a)$$

$$\text{Subject to} \quad \sum_{f \in F} x_{fv} = 1 \quad \forall v \in L, \quad (3b)$$

$$\sum_{v \in O(t)} x_{fv} \leq A_f \quad \forall t \in T, f \in F \quad (3c)$$

$$\sum_{f \in F_S} x_{fv} = 0 \quad \forall v \in V_M \quad (3d)$$

$$x \text{ binair} \quad (3e)$$

De doelfunctie en de eerste restrictie, de *cover-constraint*, zijn rechtstreeks overgenomen uit het fleet assignment model van Hane et al. (1995). De tweede restrictie (3c) is gebaseerd op vergelijking (2d) en telt per vlootype het aantal vliegtuigen die toegewezen zijn aan een vlucht op het moment t , met $t \in T$ en T de set van alle tijdstippen waarop vluchten kunnen vertrekken. Deze restrictie verzekert dat er nooit meer vliegtuigen gebruikt worden dan er beschikbaar zijn en vervangt zowel de *balance constraints* (2c) als de *beschikbaarheidsconstraints* (2d) uit het FAM2 model. De laatste restrictie (3d) verhindert dat de medium-haul vluchten ($v \in V_M$) gevlogen worden door short-haul fleets ($f \in F_S$).

Om de kostenmatrix voor de doelfunctie te construeren moeten we de toewijzingskost van elke (vlucht, vlootype) combinatie berekenen. Dit doen we aan de hand van de volgende formule.

$$C_{fv} = SK_{fv} + FK_f + VK_{fv} + WP_{fv}$$

De formule bestaat uit vier kostencomponenten: de spill kost, de vaste en variabele kost en een penalty voor het switchen van vlootype. De vaste kost is enkel afhankelijk van het vlootype, en de berekening ervan werd toegelicht op p26. De spill kost wordt zo berekend:

$$SK_{fv} = SP_v * \left(\max \left[D_0^h - CAP_f, 0 \right] + \max \left[D_0^r - CAP_f, 0 \right] \right)$$

Met SP_v de spill prijs berekend op p26, D_0^h en D_0^r de verwachte passagiersvraag van respectievelijk de heen- en terugvlucht en CAP_f de passagierscapaciteit van vlootype f . De variabele kostencomponent wordt berekend door de kostprijs per uur van vlootype f (VK_f) te vermenigvuldigen met de totale vliegtijd van de heen- en terugvlucht.

$$VK_{fv} = VK_f * \left(arrT^h - depT^h + arrT^r - depT^r \right)$$

We gaan in de scenario analyse ook na wat de invloed is van een strafkost voor het wisselen van vlootype. De strafkost is 0 als dezelfde vlootype wordt gebruikt, en groter dan 0 als er binnen of buiten de vliegtuigfamilie wordt gewisseld.

$$WP_{fv} = \begin{cases} 0 & \text{voor zelfde vloottype} \\ WP_{family} & \text{wissel binnen familie} \\ WP_{nonfamily} & \text{wissel buiten familie} \end{cases}$$

Wissels worden enkel in twee van de zeven scenarios bestraft.

3.4. Visualisatie van de flight schedule en assignment

We zijn van mening dat een visuele voorstelling van een oplossing goud waard is tijdens het opsporen van fouten en onregelmatigheden. Een goede visualisatie kan het oplossen van een probleem vergemakkelijken en het verwerven van inzichten vergroten.

Vertrekkend vanuit die gedachte hebben we een visualisatieprogramma geschreven voor het fleet assignment probleem. Het programma toont de hele tijdslijn van de centrale hub, Zaventem in dit geval, en geeft in één oogopslag de volledige flight schedule en assignment weer (figuur 3.13). De gekleurde balken stellen de *away-arcs* of samengestelde vluchten voor. De blauwe balken zijn short-haul vluchten en de rode balken medium-haul vluchten. De helderheid van de kleur is een maat voor de bezettingsgraad. Donkere balken hebben een hogere bezettingsgraad dan lichte balken. Als er een cijfer in de balk staat, betekent dit dat het vloottype dat toegewezen is aan de vlucht niet kan voldoen aan de vraag. Het cijfer staat voor het aantal “spilled” passagiers.

Alle vluchten zijn gegroepeerd per vloot. Het vloottype is links af te lezen alsook het minimum aantal vliegtuigen dat per vloot nodig is om de gekozen assignment te kunnen uitvoeren. Zo krijgt men onmiddellijke feedback over de haalbaarheid van de assignment. Klikken op een vlucht toont de relevante informatie over die vlucht in het kader rechts onderaan. Zo ziet men onder andere welke vloot oorspronkelijk deze vlucht uitvoerde, welke vloot ze nu uitvoert, het vluchtnummer uit de flight schedule en het aantal passagiers D_0 en D_{28} voor de heen- en terugvlucht.

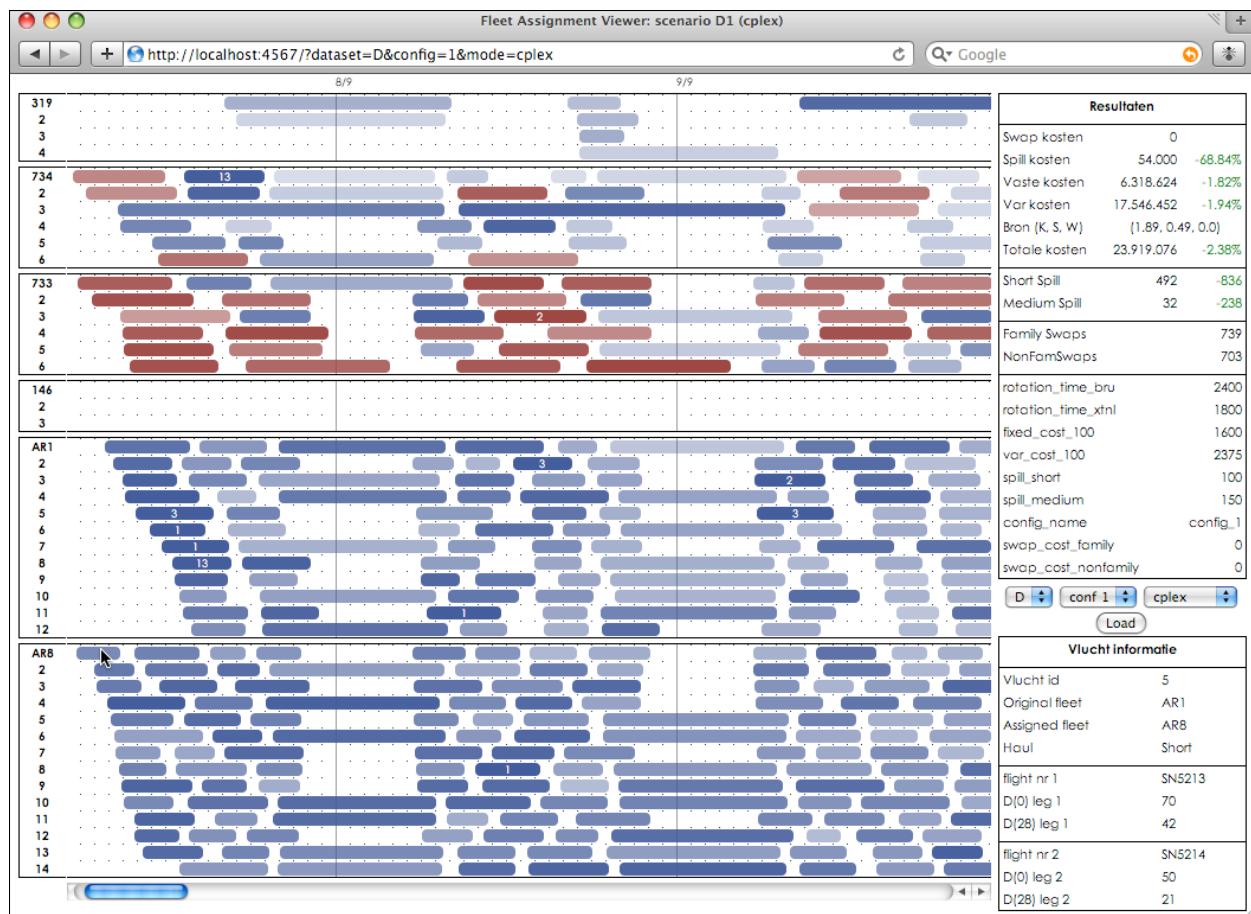


Fig 3.13. Screenshot van het flight schedule en assignment visualisatieprogramma.¹²

Het kader rechtsboven geeft een globale samenvatting van de huidige assignment weer en toont de totale kosten, de verschillende kostencomponenten, het aantal "spilled" passagiers, het aantal uitgevoerde wissels en de gebruikte parameters. Naast elk resultaat wordt het relatieve verschil met de oorspronkelijke assignment weergegeven zodat er direct feedback is over de prestatie van de nieuwe assignment. De regel 'bron (K, S, W)' verklaart vanwaar de kostenreductie komt en geeft de reductie in procentpunten weer voor de operationele kosten (K), de opportunitetskosten (S) en de strafkost voor wissels (W).

Alternatieve scenarios kunnen geladen worden via de dropdown keuzevakken en op de "load" knop te klikken.

Belangrijk om op te merken is dat in deze visualisatie -en in de hele scriptie- er geen rekening gehouden wordt met de routings van de individuele vliegtuigen. Als in de visualisatie twee vluchten in dezelfde horizontale rij voorkomen betekent dit niet noodzakelijk dat ze door hetzelfde vliegtuig gevlogen worden, enkel dat ze door dezelfde vloot worden gevlogen.

¹² zie bijlag 5.4 voor een grotere printout

Het installeren van het programma is niet triviaal dus hebben we een werkende versie online toegankelijk gemaakt op <http://thesis.andrewsblog.org>. Inloggen kan met gebruikersnaam “thesis” en wachtwoord “andrew”. De broncode is in digitale vorm meegeleverd op de cd-rom.

3.5. Oplossingsmethoden

Gewapend met een volledig visualisatiesysteem konden we de oplossing van het probleem aanpakken. Eerst werd een eenvoudig swapping algoritme geschreven in de Ruby programmeertaal, maar vrij snel bleek dat het lastig zou zijn om dit algoritme intelligenter te maken. We schakelden over op Cplex om het binaire probleem op te lossen.

Swapping algoritme

Het swapping algoritme is zeer simpel. Het overloopt de hele set vluchten op zoek naar swaps met positieve kostenreductie. Als er voordelige swaps gevonden zijn voor een vlucht wordt de swap met de hoogste kostenreductie uitgevoerd. Dit blijft zich herhalen tot er geen voordelige swaps meer worden gevonden. De high level implementatie voor één enkele iteratie is te zien in code 3.1. De volledige broncode voor het algoritme is terug te vinden in het digitale bestand “swap_algorithm.rb” en “models/assignment.rb” op de cd-rom.

```
def full_swap_pass
  @flights.each do |flight_1|
    possible_swaps = []
    @flights.each do |flight_2|
      possible_swaps.add(flight_2) if swap_valid?(flight_1,flight_2) and
        swap_profits(flight_1,flight_2) > 0
    end
    best_flight_for_swap = possible_swaps.sort_by(&:profit).first
    do_swap!(flight_1, best_flight_for_swap)
  end
end
```

Code 3.1. Ruby code voor één swapping iteratie¹³

Het algoritme is niet snel en vindt enkel eenvoudige swaps. Elke iteratie duurt 60-70 seconden op een Macbook 1.8Ghz. Voor het standaard scenario (1) waren na 4 iteraties geen voordelige swaps meer te vinden. Toch kon een totale kostenreductie van 0,95% gerealiseerd worden. De resultaten van de verschillende scenario's worden besproken in het volgende onderdeel.

Cplex model

¹³ De volledige broncode van het swapping algoritme staat in bijlage 5.6.

Het ontwikkelde cplex model is een vrij directe vertaling van het wiskundige model. De kostenmatrix wordt opgebouwd in een execute blok met behulp van Ilog Script.

Cplex lost het model voor een assignment van drie weken met ~2300 vluchten op in 1 minuut en 50 seconden op een single-core 2,3 Ghz PC.

3.6. Resultaten

We hebben tijdens de dataverzameling een aantal assumpties gemaakt. Om de sensitiviteit van de parameters die we geschat hebben na te gaan, stellen we 7 verschillende scenario's voor. We laten vervolgens zowel het swappingsalgoritme als het cplex model de assignment optimaliseren voor alle scenario's. We vertrekken van verschillende datasets om geen 'toevallige' goede weken te optimaliseren.

Datasets

Beide oplossingsmethoden worden op data van drie tijdsintervallen losgelaten. Alle tijdsintervallen beginnen op een zondag en eindigen op een zaterdag. We doen dit omdat er tijdens de nacht van zaterdag op zondag minder *nightstops* zijn.

We optimaliseren drie afzonderlijke weken uit de gemaskeerde maand en nemen vervolgens ook de drie weken samen voor de laatste dataset. Dit zijn de begin- en einddata van de datasets:

- A. De week van 7^e t.e.m 13^e
- B. De week van 14^e t.e.m 20^e
- C. De week van 21^e t.e.m 27^e
- D. Drie weken van 7^e t.e.m. 27^e

Tijdens het draaien van de optimalisaties merkten we al snel dat er weinig verschil tussen de weken onderling is en besloten we om enkel met de grotere dataset D verder te werken. Daarom zijn er van datasets A, B en C enkel voor de eerste twee scenario's resultaten. Voor het swapping algoritme werd enkel dataset B gebruikt, omdat het veel te traag was voor de grote dataset (één enkele pass duurt 19 minuten)

Scenarios

We trachten de sensitiviteit van de geschatte parameters te bestuderen aan de hand van 7 verschillende scenario's. Elk scenario is een combinatie van waarden voor elke parameter en wordt

ook een *configuratie* genoemd. De waarden van de parameters voor elk scenario staan in de onderstaande tabel (tabel 3.8).

Scenario	1	2	3	4	5	6	7
SP_{short}	100	175	75	100	100	100	100
SP_{medium}	150	225	100	150	150	150	150
$FK_{737-400}$	1,600	1,600	1,600	1,920	1,280	1,600	1,600
$VK_{737-400}$	2,375	2,375	2,375	2,850	1,900	2,375	2,375
WK_{family}	0	0	0	0	0	50	0
$WK_{nofamily}$	0	0	0	0	0	300	900,000

Tabel 3.8. De zeven parameter configuraties, uitgedrukt in euro's

Scenario 1 is het standaard scenario dat gebruik maakt van de geschatte parameterwaarden uit de vorige pagina's. In scenario's 2 en 3 bekijken we de invloed van verschillende kostprijzen voor 'verloren' passagiers. De impact van de operationele kosten parameters (de vaste en variabele vliegtuigkosten) worden in scenario's 4 en 5 bestudeerd.

In scenario 6 kijken we wat het effect is als er een strafkost wordt aangerekend voor het wisselen van vloottypes. Vloottypes wisselen buiten de families wordt zwaarder bestraft dan het wisselen binnen families. We doen dit omdat er in de realiteit ook meer werk is om de crew te herplannen. Het laatste scenario (7) hebben we toegevoegd om te zien wat er nog geoptimaliseerd kan worden als we wissels buiten vliegtuigfamilies volledig uitschakelen.

Behaalde kostenreducties

We evalueren de verbeteringen in de assignments aan de hand van de totale kostenreductie, genoteerd als een percentage t.o.v. de originele totale kosten.

$$R = \frac{TK_{origineel} - TK_{geoptimaliseerd}}{TK_{origineel}}$$

Tabel 3.9 toont de behaalde kostenreducties per scenario en per dataset. In de eerste vier rijen wordt het cplex model gebruikt. In de laatste rij tonen we de kostenreductie die behaald werden door het swapping algoritme op dataset B toe te passen.

Scenario	1	2	3	4	5	6	7
Cplex (A)	2.39%	2.60%	-	-	-	-	-
Cplex (B)	2.36%	2.66%	-	-	-	-	-
Cplex (C)	2.40%	2.67%	-	-	-	-	-
Cplex (D)	2.38%	2.64%	2.26%	2.30%	2.51%	1.60%	0.57%
Swapping (B)	0.95%	1.15%	0.79%	0.83%	1.03%	0.34%	0.25%

Tabel 3.9.Totale kostenreductie per scenario en per gebruikte dataset ¹⁴

Wat meteen opvalt is dat de behaalde kostenreducties spectaculair zijn. Het toepassen van ons model zorgt in het standaard scenario voor een verbetering van 2,38% (tabel 3.9). Dit is een kostenverlaging van 583.300 euro in 21 dagen, of 833.300 euro per maand. In het 2e scenario behalen we zelfs een verlaging van 926.487 euro per maand (tabel 3.10).

Als we het cplex model vergelijken met het swapping algoritme zien we duidelijk dat het cplex model veel betere resultaten neerzet. Dit is ook logisch omdat het cplex model altijd de meest optimale assignment uitkomt en het swapping algoritme enkel eenvoudige swaps uitvoert.

We zien ook dat er geen grote verschillen zijn tussen de drie wekelijkse datasets (A, B en C). Omdat de grotere 3-wekelijkse dataset (D) representatief is voor de drie aparte weken, voeren we de rest van onze analyse enkel uit op die dataset.

Scenario	1	2	3	4	5	6	7
Kostenreductie (21d)	583,317	648,541	552,301	674,592	492,166	390,958	138,601
Kostenreductie (maand)	833,310	926,487	789,001	963,703	703,094	558,511	198,001

Tabel 3.10. Kostenreductie per scenario, uitgedrukt in euro's

Figuur 3.14 brengt de impact van verschillende scenario's beter in beeld. De scenario's staan naast elkaar: De hoogte van de balk komt overeen met de totale kostenreductie. De verschillende kleuren tonen de oorsprong van de reductie. De donkere delen stellen de reducties in operationele kosten voor en de lichte delen de reducties in opportunitetskosten. De operationele kosten zijn de kosten van de vluchten en bestaan uit de vaste en variabele vliegtuigkosten. De spill kost is de enige opportunitetskost. De rode balk in scenario 6 is een representatie van de strafkost

14 De ruwe cijfers zijn te vinden in bijlage 5.5

en toont het deel van de kostenreductie dat verloren gaat doordat voor de uitgevoerde wissels een strafkost moet worden ‘betaald’.

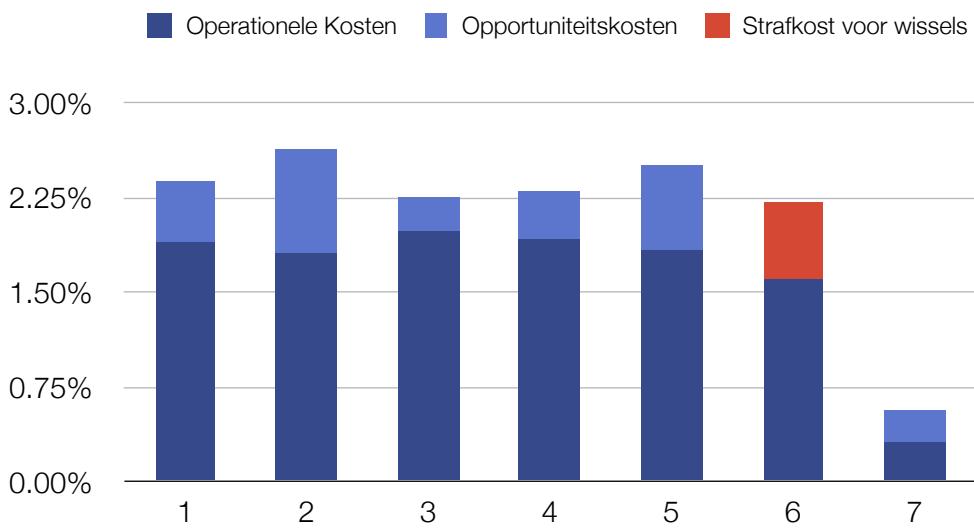


Fig 3.14. Kostenreductie per senario voor het Cplex model

Het resultaat van de optimalisatie daalt drastisch als we een strafkost invoeren voor uitgevoerde wissels: van 2,38% naar 1,60% in scenario 6 en naar 0,57% in scenario 7, waar we wissels buiten vliegtuigfamilies niet toelaten. De overige scenario’s hebben een geringere impact.

Uit figuur 3.14 leiden we af dat de reductie in operationele kosten het grootst is. Ons model haalt de beste verbeteringen door vloten zodanig toe te wijzen dat er minder kosten gemaakt worden. Er wordt vooral voor vluchten met een lage bezettingsgraad van grote naar kleine vloottypes gewisseld. Dit is duidelijk zichtbaar in de visualisatiesoftware waarvan we 2 screenshots hebben toegevoegd in bijlage 5.4. De eerste toont een deel van de originele assignment en de tweede toont een deel van de geoptimaliseerde assignment. Op de tweede screenshot ziet men duidelijk dat er meer vluchten toegewezen worden aan de kleinere (goedkopere) vloottypes zoals de Avro RJ100.

De resultaten van scenario 2 & 3 en 4 & 5 zijn interessant. Wanneer de spill prijs hoger is (scenario 2) zal het model trachten zo weinig mogelijk passagiers te ‘verliezen’. Dit gaat echter ten koste van de operationele efficiëntie. Het omgekeerde kan men waarnemen in scenario 3: daar zijn de spill prijzen lager en is het dus interessanter voor het model om de operationele efficiëntie te verbeteren. Op het linkse deel van figuur 3.15 is dit fenomeen duidelijk zichtbaar: het aantal “spilled passagiers” ligt lager in scenario 2 en hoger in scenario 3.

Scenario 5 vertoont een analoog fenomeen: de parameters van de operationele kosten liggen lager. Daardoor is het “goedkoper” om spilled passagiers te vermijden. Scenario 4, waar de parameters voor de operationele kosten hoger liggen, vertoont weinig verschillen met het standaard scenario.

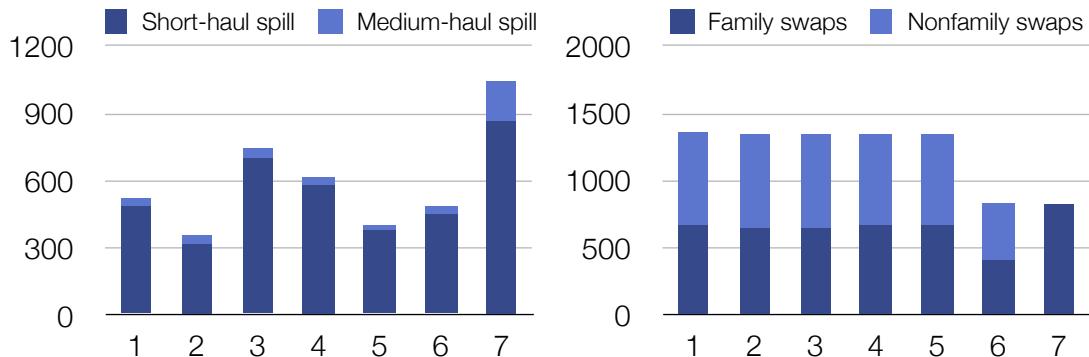


Fig 3.15. Spilled passagiers (links) en uitgevoerde wissels (rechts) per scenario

Het aantal uitgevoerde wissels blijft in de eerste 5 scenario's nagenoeg constant. Pas wanneer we wissels penalizeren in scenario 6 en 7 zien we een daling. Het verbieden van wissels buiten een vliegtuigfamilie (scenario 7) wordt door het model gecompenseerd door meer te wisselen binnen dezelfde vliegtuigfamilies.

Algemeen genomen blijft het niveau van de behaalde kostenreducties gehandhaafd, ook als we de geschatte parameters aanpassen. Enkel wanneer we wissels buiten vliegtuigfamilies verbieden dalen de resultaten aanzienlijk. We kunnen dus stellen dat het gebruik van ons model concrete kostenreducties verwezenlijkt en dat het niet al te gevoelig is voor de assumpties die we maakten.

3.7. Testen op robuustheid via simulatie

De resultaten van de optimalisatie klinken veelbelovend, maar omdat we de gebruikte methode zijn ze te optimistisch. Omdat we de optimalisatie baseren op een voorspelling van de vraag en later diezelfde voorspelde vraag ook gebruiken bij de berekening van het resultaat, bekijken we in feite het meest optimistische scenario. We optimaliseren namelijk met een zekere verwachte vraag en gaan er van uit dat die verwachting volledig uitkomt. De werkelijke vraag op vertrekdag zal echter nooit gelijk zijn aan wat we hadden voorspeld.

Monte-Carlo simulatie moet gelijk zijn aan de kans dat die waarde in de werkelijkheid voorkomt. De gegeven probabiliteitsdistributie moet dus representatief zijn voor de te simuleren parameter.

De parameter die we wensen te simuleren is de werkelijke passagiersvraag op vertrekdag. Zo kunnen we onderzoeken wat er gebeurt als onze voorspelling van de passagiersvraag niet uitkomt.

We halen de probabiliteitsdistributie van de passagiersvraag uit de verkoopgegevens van Brussels Airlines. In feite gaan we kijken wat de werkelijke vraag was, gegeven de vraag vier weken voor het vertrek. We doen dit voor alle flight legs uit de verkoopgegevens en stellen hiervan een probabiliteitsdistributie op die we gebruiken voor de simulatie. Voor de duidelijkheid noemen we de vraag vier weken voor vertrek de “input” en de vraag op vertrekdag de “output”. Omdat we niet genoeg verkoopgegevens hebben om voor elke inputwaarde een probabiliteitsdistributie op te stellen, hebben we gekozen om de inputwaarden samen te nemen per 5. Uit de verkoopdata blijkt dat verschillende inputwaarden gaan van 0 tot 168. We groeperen deze inputs in 34 groepen van 5: [0-4] , [5-9] , [10-14] , etc en bepalen de probabiliteitsdistributie voor elke groep aan de hand van de sql query in code 3.2. Als voorbeeld berekenen we de probabiliteitsdistributie voor inputs van 40 tot 44:

```
select seats_sold(l.id,0) as "0d",count(*)
from sales_legs l, sales_ticks t
where l.id = t.sales_leg_id and t.date = l.date - 28
and t.seats_sold between 40 and 44
```

Code 3.2. SQL query voor het ophalen van de probabiliteitsdistributie voor input [40,44]

Deze query geeft een lijst weer met alle voorkomende outputwaarden en hun frequentie. Een fractie van die lijst wordt weergegeven in tabel 3.II. Op figuur 3.16 is de histogram te zien van deze probabiliteitsdistributie.

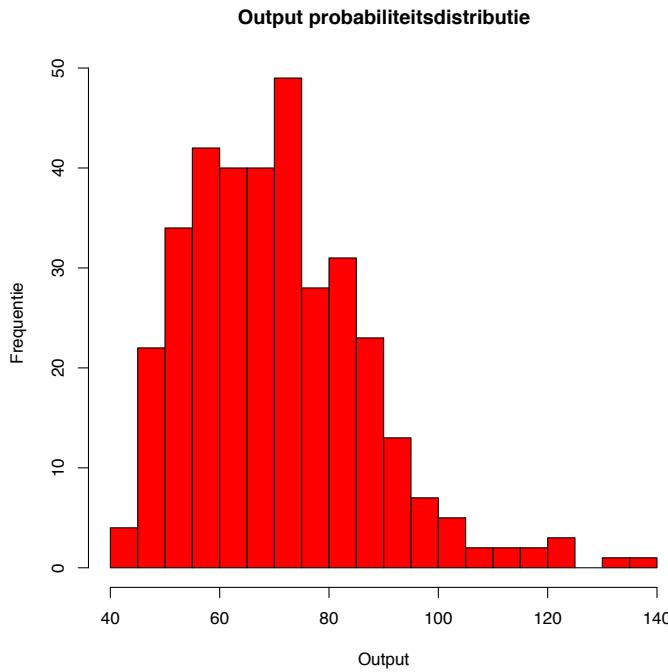


Fig 3.16. Histogram van de probabiliteitsdistributie voor inputs [40,44]

output	frequentie	probabiliteit
63	9	2.58%
64	8	2.29%
65	11	3.15%
66	10	2.87%
67	8	2.29%
68	5	1.43%
69	9	2.58%
70	8	2.29%
71	9	2.58%
72	13	3.72%
73	5	1.43%
74	14	4.01%
75	8	2.29%

Tabel 3.11. Fractie van de probabiliteitsdistributie voor inputs [40,44]

Ter illustratie: Stel dat we merken dat een flight leg 4 weken voor vertrek 42 verkochte zetels heeft (input), dan is de kans dat op de dag van vertrek deze flightleg 65 verkochte zetels heeft (output) gelijk aan 3,15 %, en de kans dat er 74 verkochte zetels zijn 4,01 %. De volledige kansverdeling wordt samengevat in figuur 3.16. waarbij de hoogte van elke balk, de kans weergeeft dat die output zal voorkomen.

Aan de hand van de probabiliteitsdistributies van de 34 input-groepen zullen we de werkelijke passagiersvraag simuleren, met als doel na te gaan hoe robuust onze geoptimaliseerde assignment is. We gebruiken de cplex geoptimaliseerde assignment van dataset D onder scenario 1.

Een visualisatie van deze simulatietechniek kan verhelderend zijn (figuur 3.17). Op de figuur ziet men dat voor elke flight leg het verkochte aantal zetels 28 dagen voor vertrek (D_{28}) gebruikt wordt als input voor de Monte-Carlo, die aan de hand van de probabiliteitsdistributie een verkocht aantal zetels op vertrekdag (D_0) kiest. Men ziet dat deze waarden verschillen van het verwachte aantal.

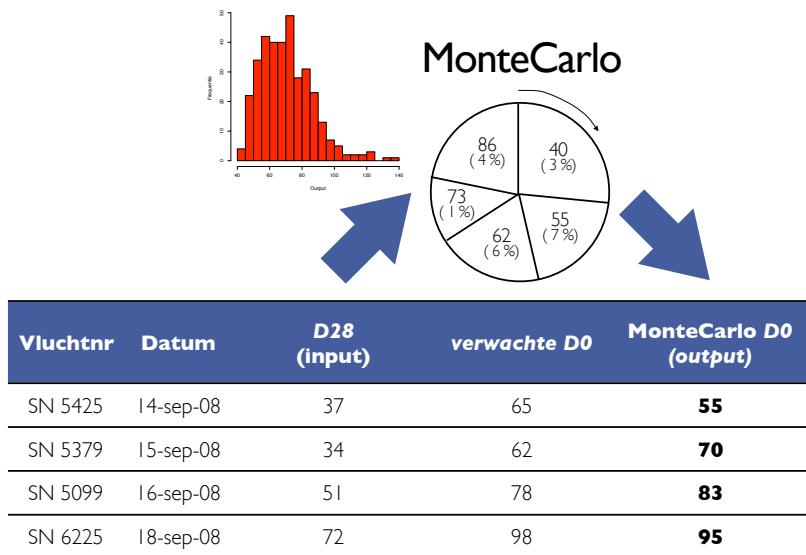


Fig 3.17. De Monte-Carlo simulatie kiest voor elke flight leg een eindvraag D_0

Ons optimalisatiemodel werd in het vorige deel toegepast uitgaande van de voorspelde vraag D_0 . Die voorspelde vraag werd geschat aan de hand van de regressievergelijking op p30. Met elke Monte-Carlo simulatie creëren we nu een andere *realistische* uitgangssituatie, met een andere passagiersvraag D_0 . Het doel van de simulatie is nagaan in hoeverre de geoptimaliseerde assignment beter scoort dan de originele assignment in elk van de gesimuleerde uitgangssituaties.

De ruby implementatie van de simulatie vindt u terug in bijlage 5.8. Dit script berekent de kostenreductie voor de nieuwe uitgangssituatie en geeft de verbetering ten opzichte van de originele assignment weer als een percentage. Dit wordt 10.000 keer herhaald en de uitkomst wordt weggeschreven in een tekstbestand. De typische statistieken (gemiddelde, standaard deviatie, minimum en maximum) worden weergegeven na elke run. De runtime van de simulatie voor de assignment van dataset D en scenario 1 duurt zo'n 26 minuten op een MacBook 1.8Ghz met 2GB

ram. De kostenreductie percentages zijn op Fig. 3.18 in een histogram opgenomen. De statistieken staan in de tabel 3.12.

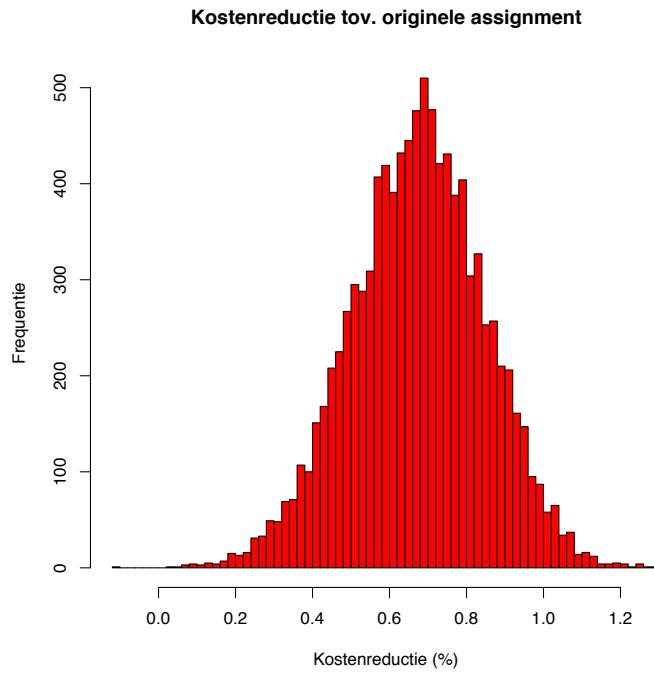


Fig 3.18. Histogram simulatie voor cplex assignment D1

Minimum	-0.118%
Maximum	1.287%
Average	0.673%
StdDev	0.173%

Tabel 3.12. Statistieken van de simulatie voor cplex assignment D1

Opvallend is dat van de kostenreductie van 2,38 % nog slechts weinig overblijft. Gemiddeld scoort onze geoptimaliseerde assignment namelijk slechts 0,67 % beter dan de originele assignment. We moeten echter bij deze zwakke resultaten één grote opmerking maken: De originele assignment van Brussels Airlines wordt ook beïnvloed door de nieuwe passagiersvraag.

Om de vergelijking van de resultaten eerlijker te maken wordt de simulatie herhaald, maar nu wordt de kost van de originele assignment ook herberekend aan de hand van de gesimuleerde uitgangssituatie. Op die manier worden de resultaten van beide assignments (de originele en de

geoptimaliseerde) berekend vertrekkende van dezelfde uitgangssituatie. Deze opzet is geconceptualiseerd in fig 3.19.

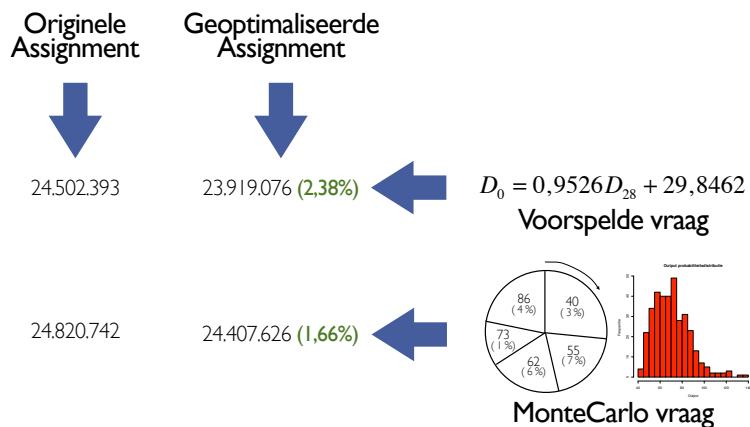


Fig 3.19. Monte-Carlo 2 conceptueel

Op figuur 3.20 ziet u een histogram met de resultaten van de 2e simulatieronde. Tabel 3.13 toont de statistieken. De resultaten van deze simulatieronde zijn een stuk beter dan die van de vorige simulatieronde.

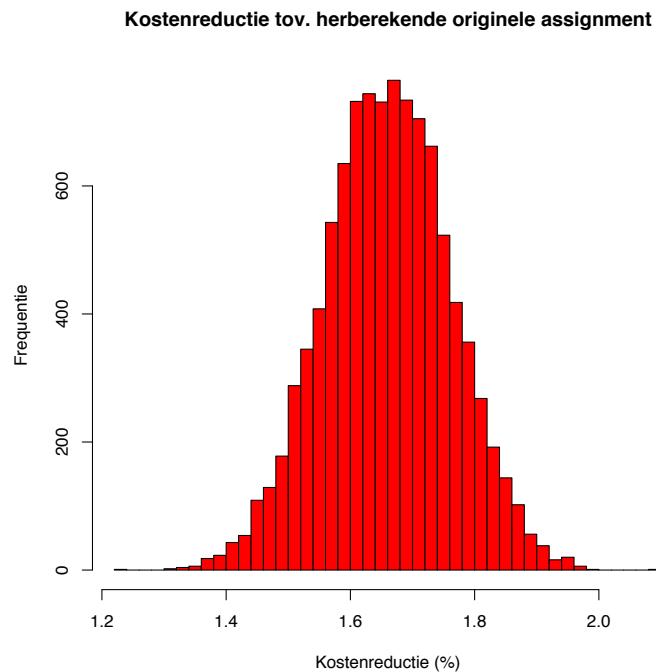


Fig 3.20. Histogram simulatie met herberekende originele kosten

Minimum	1.227%
Maximum	2.098%
Average	1.660%
StdDev	0.102%

Tabel 3.13. Statistieken van de simulatie met herberekende originele kosten

Gemiddeld genomen scoort de geoptimaliseerde assignment een kostenreductie van 1,66%, wat goed is voor een besparing 581.047 euro per maand. Dit is weliswaar lager dan wat we voordien schreven, maar het blijft een belangrijke verbetering.

We hebben de Monte-Carlo simulatie toegepast op de geoptimaliseerde assignment van scenario 7 met als doel na te gaan of we nog voordeel halen uit ons model. Zelfs in dit slechtste geval bedraagt de gemiddelde kostenreductie 0,33% (StdDev: 0,05%). Dit is aanzienlijk minder dan wat we in het standaard scenario behalen, maar blijft positief. Op maandbasis betekent dit een besparing van 115.509 euro, wat zeker de moeite waard is.

Met de Monte-Carlo simulatie wordt het relatief zwak voorspellend vermogen van de passagiersvraag op de rooster gelegd. De regressieanalyse die we gebruikt hebben overleeft de uitdaging en bewijst dat het model zelfs met een matig forecastingssysteem, degelijke resultaten kan boeken. We kunnen daardoor stellen dat de assignments die het model voorbrengt robuust zijn.

Verder onderzoek kan aantonen of het gebruik van reserve capaciteit in het model de resultaten na de Monte-Carlo simulatie kan verbeteren. We hebben namelijk een vermoeden dat de robuustheid van de assignment verhoogt kan worden als men het model laat vertrekken van de verwachte vraag +10%. Waarbij de 10% voor een reverse capaciteit staat.

4. ALGEMEEN BESLUIT

De hevige concurrentie, stijgende olieprijzen en stagnerende markt verplichten luchtvaartmaatschappijen te streven naar operationele efficiëntie. In alle planningsdomeinen wordt gezocht naar kostenreducties. Wij hebben in deze scriptie het fleet assignment problem bestudeerd en onderzocht of er een opportunitet tot kostenreductie is.

Na het verkennen van bestaande fleet assignment modellen werd in het derde hoofdstuk een case study uitgevoerd op reële data van Brussels Airlines met als doel het zoeken naar manieren om verbeteringen in de fleet assignment te boeken op korte termijn. We stelden tijdens de dataverzameling een regressieanalyse voor om het verwachte aantal passagiers te voorspellen waarop we onze optimalisatie konden baseren. We ontwikkelden een nieuw fleet assignment model rond de specifieke situatie van Brussels Airlines. Het nieuwe model is gebaseerd op het basismodel van Hane et al (1995) en aangepast aan een "hub and spoke" netwerktopologie met één enkele hub. De grote vereenvoudiging die we gerealiseerd hebben is het samennemen van de vluchten en het achterwege laten van de bestemmingen. Daardoor werd het mogelijk om het hele netwerk samen te vatten in één enkele tijdslijn, zonder in te boeten aan volledigheid. Dit opende de deuren voor aangepaste visualisatiesoftware.

We ontwikkelde een visualisatieprogramma waarmee we gemakkelijk het model konden valideren. Het hielp bij de interpretatie van de resultaten door het zichtbaar maken van de verbeteringen. Voordat de software echter geïntegreerd kan worden in de workflow van een luchtvaartmaatschappij moet ons prototype verder ontwikkeld worden.

In de case study hebben we een aantal assumpties moeten maken. De operationele kostenfactoren en de kostprijs van een 'verloren' passagier werden geschat. Verder onderzoek kan betere waarden voor deze parameters opleveren waarmee de representativiteit van de case study verhoogt kan worden. De sensitiviteit van de gemaakte assumpties werd door middel van scenario analyse gecontroleerd. Voor de meeste assumpties was de gevoeligheid geen probleem voor het model. Enkel het toevoegen van een strafkost voor wissels had een belangrijke impact op de resultaten. We hebben aan de hand van een Monte-Carlo simulatie kunnen aantonen dat ons fleet assignment model erin slaagt assignments te genereren die in alle situaties beter zijn dan de oorspronkelijke assignment. De resultaten zijn spectaculair. We behalen kostenreducties van 2,3% of 833.000 euro per maand. Zelfs in het slechtste geval zorgt ons model voor een verlaging van de kosten met 0,3 % of 115.000 euro per maand.

Onze case study is uitgevoerd met een aantal beperkingen. We hebben namelijk enkel de data onderzocht van één bepaalde drukke maand. Het is niet zeker of ons model in minder drukke maanden ook voor verbeteringen kan zorgen. Verder onderzoek moet dit uitwijzen. Ook werd de analyse volledig gebaseerd op kwantitatieve verbeteringen. Alle resultaten zijn uitgedrukt in monetaire waarden. Nochtans zijn er ook kwalitatieve maatstaven zoals voorspelbaarheid van de

werkuren voor het personeel en consistentie van ingezette vliegtuigen voor de passagiers. Deze maatstaven zijn moeilijk te berekenen, maar kunnen wel geïntegreerd worden in het model door middel van strafkosten. De probabiliteitsdistributies die gebruikt werden voor de Monte-Carlo simulatie komen uit de verkoopgegevens van één enkele maand. Deze beperking kan makkelijk opgeheven worden door bijvoorbeeld de verkoopgegevens van een heel jaar te gebruiken.

Of de behaalde resultaten van ons model haalbaar zijn voor Brussels Airlines zal vooral afhangen van de mate van crew flexibiliteit. Als men de fleet assignment niet kan wijzigen één maand voor vertrek, zal het niet mogelijk zijn dezelfde kostenreducties te halen als in deze scriptie. Indien het voor Brussels Airlines onmogelijk is om de benodigde flexibiliteit te verkrijgen, hebben we toch kunnen aantonen welke potentiële kostenreducties te laten liggen.

4. GERAADPLEEGDE WERKEN

- Abara, J., 1989. Applying integer linear programming to the fleet assignment problem. *Interfaces* 19 (4), 20–28.
- Berge, M.E., Hopperstad, C.A., 1993. Demand driven dispatch: A method for dynamic aircraft capacity assignment, models and algorithms. *Operations Research* 41, 153–168.
- Bertsimas, D., and de Boer, S., 2005. Simulation-Based Booking Limits for Airline Revenue Management. *Operations Research*, 53 (1), 90-106.
- Boyd, E.A., and Bilegan, I.C., 2003. Revenue Management and E-Commerce. *Management Science*, 49 (10), 1363-1386.
- Chen, V.C.P., Günther, D., and Johnson, E.L., 2003. Solving for an optimal airline yield management policy via statistical learning. *Appl. Statist.*, 52 (1), 19-30.
- Dorndorf, U., Drexel, A., Nikulin, Y., and Pesch, E. (2007). "Flight gate scheduling: State-of-the-art and recent developments." *Omega*, 35, 326-334.
- Fecheyr-Lippens B., 2007. Personeelsplanning in de luchtvaartsector: Case Brussels Airlines.
- Gosavi, A., Ozkaya, E., and Kahraman, A.F., 2007. Simulation optimization for revenue management of airlines with cancellations and overbooking. *OR Spectrum*, 29, 21-38.
- Gu, Z., Johnson, E.L., Nemhauser, G.L., Wang, Y., 1994. Some properties of the fleet assignment problem. *Operations Research Letters* 15, 59–71.
- Hane, C.A., Barnhart, C., Johnson, E.L., Marsten, R.E., Nemhauser, G.L., Sigismondi, G., 1995. The fleet assignment problem: Solving a large-scale integer program. *Mathematical Programming* 70, 211–232.
- Jacobs P.H.M., Verbraeck A., Mulder J.B.P., 2005. Flight scheduling at KLM. *Proceedings of the 2005 Winter Simulation Conference*, Vols 1-4, 299-306.
- Karaesmen, I., and van Ryzin, G., 2004. Overbooking with Substitutable Inventory Classes. *Operations Research*, 52 (1), 83-104.
- Lee, L.H., Lee, C.U., and Tan, Y.P., 2007. A multi-objective genetic algorithm for robust flight scheduling using simulation. *European Journal of Operational Research* 177, 1948-1968.
- McGill, J.I., and van Ryzin, G.J., 1999. Revenue Management: Research Overview and Prospects. *Transportation Science*, 33 (2), 233-256.
- Rushmeier, R.A., Kontogiorgis, S.A., 1997. Advances in the optimization of airline fleet assignment. *Transportation Science* 31, 159–169.

Sherali H.D., Bish, E.K., Zhu, X, 2005. Airline fleet assignment concepts, models, and algorithms. European Journal of Operational Research 172 (2006) 1–30.

Vanhoucke M., 2008. Applied operations research. Academia Press / Gent.

Walczak D, Brumelle S, 2006. Semi-Markov information model for revenue management and dynamic pricing. OR Spectrum 29 (2007), 61–83.

5.

BIJLAGEN

5.1. Flight schedule bestand

Als voorbeeld worden hieronder enkele data rijen van het "Flight Schedule" bestand getoond.

FlightNr	From	Till	Days	Departure	Arrival	Fleet
SN5129	1-sep-2008	4-sep-2008	12_4__	BRU	07:45	Dest14 09:00
SN5129	3-sep-2008	10-sep-2008	_3_5__	BRU	07:45	Dest14 09:00
SN5129	8-sep-2008	25-sep-2008	12_45__	BRU	07:45	Dest14 09:00
SN5222	1-sep-2008	3-sep-2008	123____	Dest34	17:20	BRU 18:20
SN5222	4-sep-2008	4-sep-2008	___4__	Dest34	17:20	BRU 18:20
SN5222	5-sep-2008	19-sep-2008	_2_5_7	Dest34	17:20	BRU 18:20
SN5260	20-sep-2008	27-sep-2008	____6_	Dest27	16:20	BRU 17:40
SN5260	22-sep-2008	30-sep-2008	12345_7	Dest27	16:20	BRU 17:40
SN5273	1-sep-2008	5-sep-2008	12345__	BRU	18:20	Dest27 19:45
						32S

De file telt in totaal 1587 flight leg groepen. Als men deze samengestelde groepen opsplitst naar individuele flight legs komt men aan 6391 flight legs. Het ontdubbelen van deze flight leg groups wordt gedaan met behulp van een zelfgeschreven ruby script dat de flight legs op een gestructureerde manier opslaat in een database.

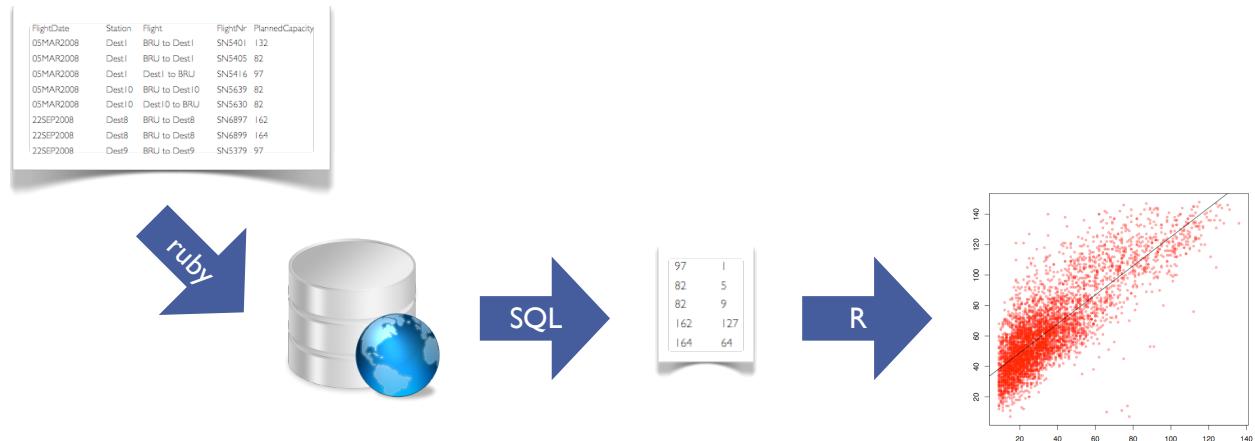
5.2. Verkoophistoriek bestand

Ter illustratie van de dataverwerking die werd gedaan voor het onderzoeken van de verkoopgeschiedenis staat hieronder een uittreksel van het bestand met de verkoophistoriek die ons werd geleverd door Brussels Airlines.

SalesDate	FlightDate	Station	Flight	FlightNr	PlannedCapacity	SeatsSold
01SEP2008	05MAR2008	Dest1	BRU to Dest1	SN5401	132	0
01SEP2008	05MAR2008	Dest1	BRU to Dest1	SN5405	82	1
01SEP2008	05MAR2008	Dest1	Dest1 to BRU	SN5416	97	1
01SEP2008	05MAR2008	Dest10	BRU to Dest10	SN5639	82	5
01SEP2008	05MAR2008	Dest10	Dest10 to BRU	SN5630	82	9
30SEP2008	22SEP2008	Dest8	BRU to Dest8	SN6897	162	127
30SEP2008	22SEP2008	Dest8	BRU to Dest8	SN6899	164	64
30SEP2008	22SEP2008	Dest9	BRU to Dest9	SN5379	97	60
30SEP2008	22SEP2008	Dest9	Dest9 to BRU	SN5370	82	25
30SEP2008	22SEP2008	Dest9	Dest9 to BRU	SN5374	97	27

Ook deze gegevens werden ingelezen met behulp van een ruby script en opgeslagen in de relationele database. Alle datapunten werden eveneens gelinkt aan de bijhorende flight input waarden leg.

5.3. Data visualisatie



Eenmaal de ruwe data ingelezen is in de database, wordt de nodige informatie met behulp van specifieke Structured Query Language (SQL) statements opgevraagd. Deze informatie wordt geanalyseerd met de "R" statistical language¹ die de uiteindelijke grafiek genereert en wegschrijft als pdf. Om dit te stroomlijnen wordt gebruik gemaakt van embedded R scripting in de PostgreSQL² zodat we functies kunnen schrijven voor elk grafiek type. De functie voor het genereren van een scatter plot met regressielijn staat hieronder:

```
CREATE OR REPLACE FUNCTION r_scatter(title text, sql text)
RETURNS text AS
$BODY$
data <- pg.spi.exec(sql);
pdf('/tmp/scatter_plot.pdf');
plot(data,main=title,cex=0.3,
      pch=20,col=rgb(255,0,0,maxColorValue=255),ylim=c(0,100));
regressie <- lm(data);
abline(regressie);
dev.off();
print('done');
```

Code 5.1. Functie voor het renderen van een scatterplot met regressielijn³

Alle gebruikte SQL querries staan op het digitale bestand querries.sql

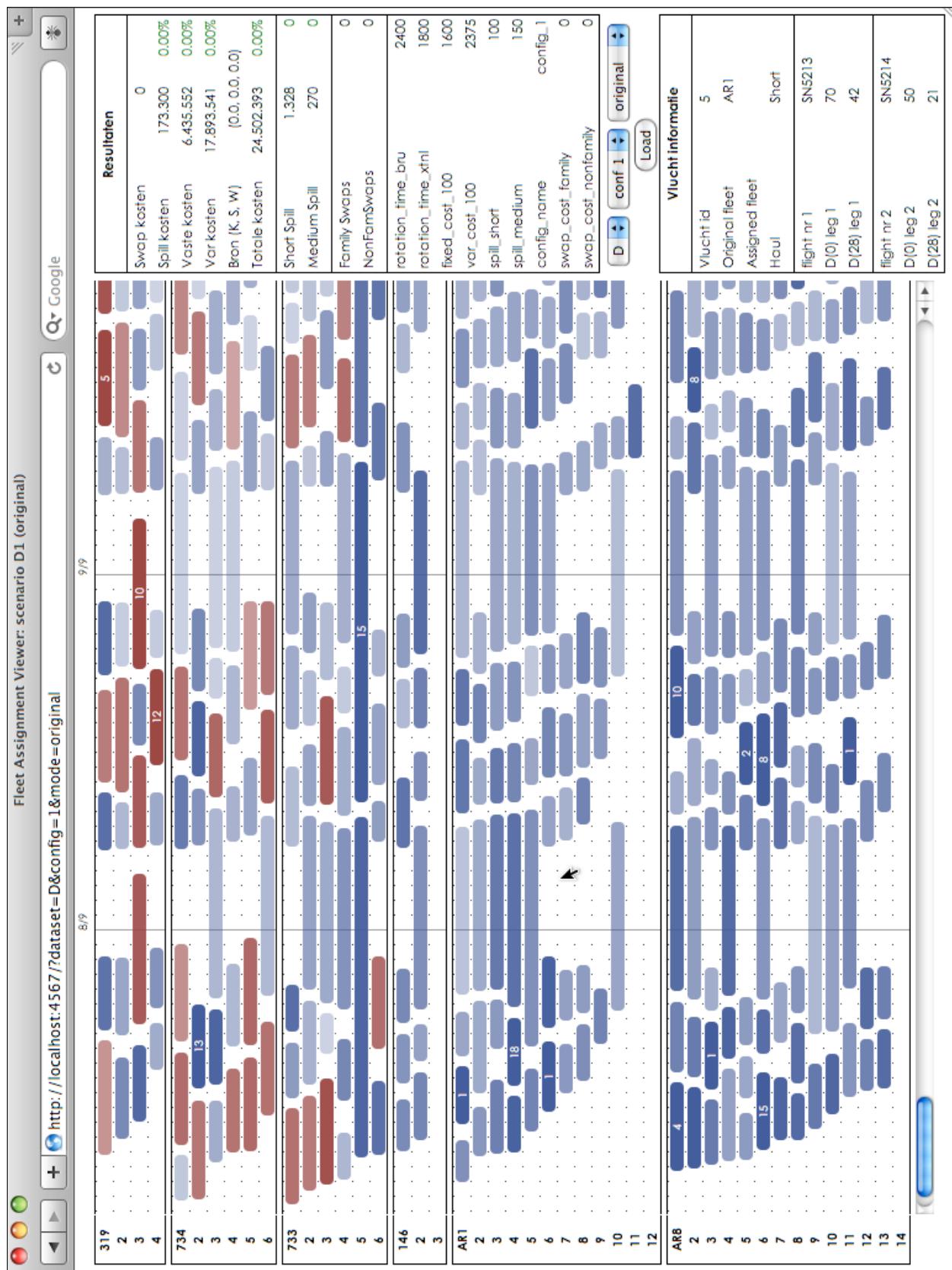
1 <<http://www.r-project.org/>>

2 <<http://www.joeconway.com/plr/>>

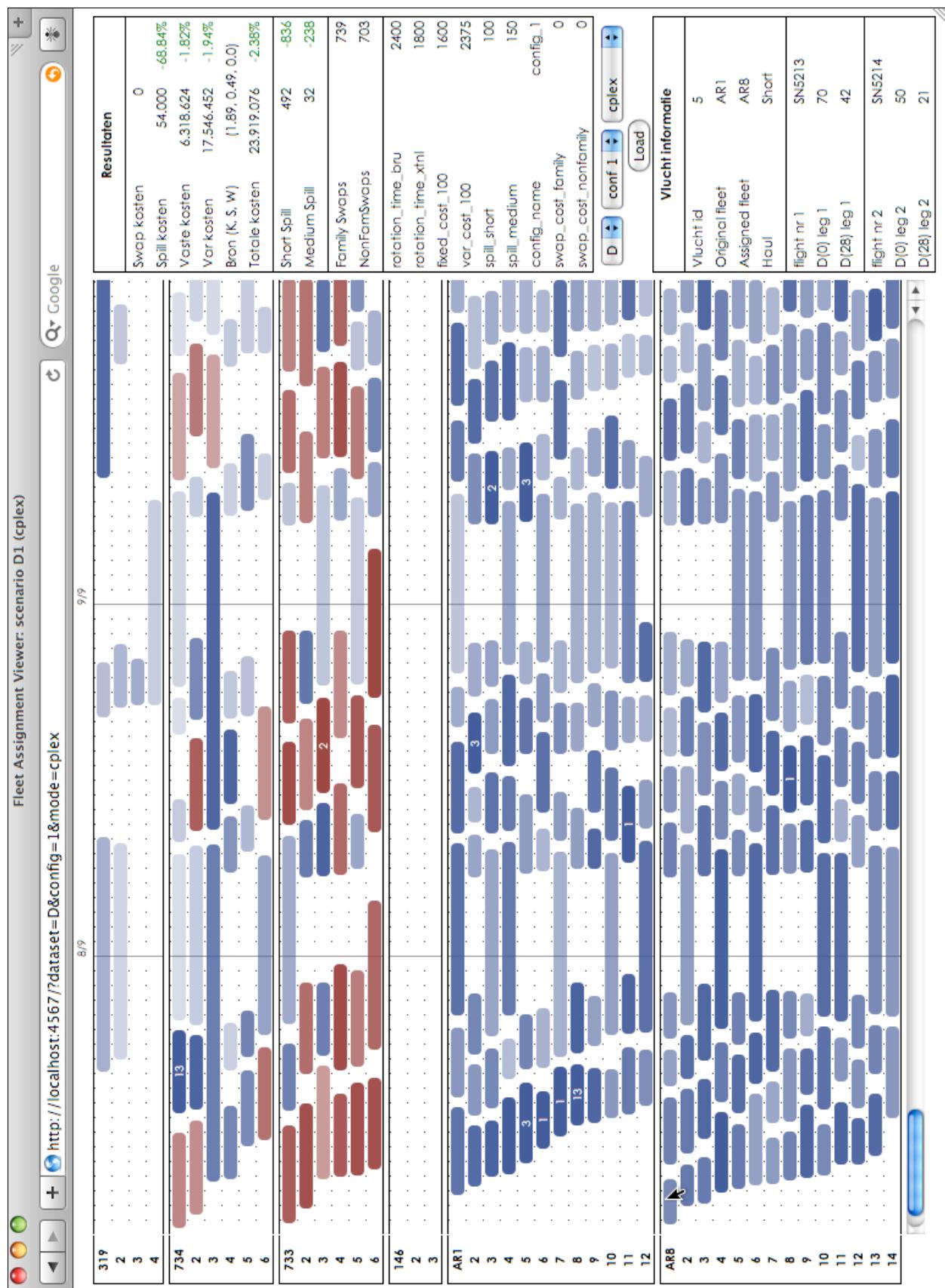
3 met hulp van <http://www.varlena.com/GeneralBits/Tidbits/bernier/art_66/graphingWithR.html>

5.4. Screenshots Assignment Viewer

Originele assignment. Let op het aantal vluchten toegewezen aan grotere vliegtuigen.



Geoptimaliseerde assignment. Meer vluchten worden aan de kleinere vloottypes toegewezen.



5.5. Totale kostencijfers en statistieken

In de volgende kostentabellen staan alle cijfers in euros. Deze cijfers werden berekend per scenario voor dataset D (drie weken)

ORIGINELE ASSIGNMENT	1	2	3	4	5	6	7
Opportunitetskosten	173,300	259,950	126,600	173,300	173,300	173,300	173,300
Vaste kosten	6,435,552	6,435,552	6,435,552	7,722,662	5,148,442	6,435,552	6,435,552
Variabele kosten	17,893,541	17,893,541	17,893,541	21,472,249	14,314,833	17,893,541	17,893,541
Totale kosten	24,502,393	24,589,043	24,455,693	29,368,211	19,636,575	24,502,393	24,502,393

Tabel 5.1. De kostencomponenten voor de originele assignment per scenario

CPLEX ASSIGNMENT	1	2	3	4	5	6	7
Opportunitetskosten	54,000	55,650	57,175	62,750	42,500	50,000	113,150
Vaste kosten	6,318,624	6,324,640	6,313,088	7,580,890	5,057,510	6,327,456	6,427,840
Variabele kosten	17,546,452	17,560,212	17,533,129	21,049,979	14,044,399	17,585,529	17,822,802
Strafkost voor wissels	-	-	-	-	-	148,450	-
Totale kosten	23,919,076	23,940,502	23,903,392	28,693,619	19,144,409	24,111,435	24,363,792
Kostenreductie (21d)	583,317	648,541	552,301	674,592	492,166	390,958	138,601
Kostenreductie (maand)	833,310	926,487	789,001	963,703	703,094	558,511	198,001

Tabel 5.2. De kostencomponenten voor de cplex geoptimaliseerde assignment per scenario

CPLEX ASSIGNMENT	1	2	3	4	5	6	7
Short-haul spill	492	326	705	581	377	452	863
Medium-haul spill	32	30	43	31	32	32	179
Family swaps	671	659	648	663	669	412	825
Nonfamily swaps	684	685	690	687	676	410	0

Tabel 5.3. De spill en wissel cijfers voor de cplex geoptimaliseerde assignment per scenario

5.6. Ilog CPLEX model

De broncode van het cplex model "cplex.mod":

```
*****  
* OPL 5.0 Model  
* Author: Andrew Fecheyr Lippens  
* Creation Date: 9/05/2009 at 23:49  
*****  
  
// Setup  
int horizon = 60*24*21-1; // Number of minutes in a week  
int blocktime = 5;  
  
// Inladen van parameter.dat  
int rotation_time_bru = ...;  
int spill_short = ...;  
int spill_medium = ...;  
int fixed_cost_100 = ...;  
int var_cost_100 = ...;  
int swap_cost_family = ...;  
int swap_cost_nonfamily = ...;  
int rotation_time_external = ...;  
  
// Opp bouwen van alle tijdspunten in het netwerk  
int check_count = ftoi((horizon+1) / blocktime);  
{int} timepoints = {i*blocktime | i in 0..check_count};  
  
// Data van elke vloot  
tuple FleetType  
{  
    string ba_code;           // Id vloot  
    int aircrafts;           // Aantal vliegtuigen  
    string haul;              // medium-short haul  
    int seats;                // Aantal zetels  
    int fixed_cost_percent;   // Vaste kosten per vlucht  
    int var_cost_percent;     // Variabele kosten per minuut in de lucht  
    int family;               // vliegtuigfamily: 1 = AR1,AR8,146 en 2 = 733,734  
}  
  
// Invullen van vloot info uit data file  
{FleetType} fleets = ...;  
  
// Vlucht datastructuur  
tuple FlightStruct  
{  
    int id;                  // flight id  
    string original_ba_code; // ba_code van de originele assignment  
    int original_family;    // family of the original aircraft  
    int depT;                // departure time  
    int arrT;                // arrival time  
    int flightT;             // Totale tijd in de lucht  
    string haul;              // medium-short haul  
    int pax_1;                // Passagiersvraag voor 1e leg  
    int pax_2;                // Passagiersvraag voor 2e leg  
}  
  
{FlightStruct} flights = ...;  
  
// Controleer of de tijden kloppen  
assert forall(l in flights) 0 <= l.depT;  
assert forall(l in flights) 0 <= l.arrT;  
assert forall(l in flights) horizon >= l.depT;  
assert forall(l in flights) horizon >= l.arrT;  
assert forall(l in flights) l.depT < l.arrT;  
  
// De operationele kostprijs  
// Voor elk koppel (fleet,flight) definieren we Cost = spill_cost + fixed_cost + var_cost*flightT  
float Cost[fleets][flights];  
  
// Cost Initialization  
execute COST_INIT {  
    for(var l in flights) {  
        for(var f in fleets) {  
            var spill_cost = 0;  
            // bereken de totale spill  
            var spill = 0;  
            if(f.seats < l.pax_1) { spill = spill + l.pax_1 - f.seats }  
            if(f.seats < l.pax_2) { spill = spill + l.pax_2 - f.seats }  
            if(0 < spill) {  
                var spill_price = (l.haul == "Medium" ? spill_medium : spill_short );  
                spill_cost = spill * spill_price;  
            }  
        }  
    }  
}
```

```

// bereken de swap_penalty
var swap_penalty = 0;
if (l.original_ba_code != f.ba_code) {
    // er is geswapped
    if (l.original_family == f.family) {
        swap_penalty = swap_cost_family;
    }
    else
    {
        swap_penalty = swap_cost_nonfamily;
    }
}
Cost[f][l] = spill_cost + 2*(f.fixed_cost_percent/100)*fixed_cost_100 + (f.var_cost_percent/100)*(l.flightT/60)*var_cost_100 + swap_penalty;
}

// Maak arrays van vluchten voor elk tijdstip
{FlightStruct} time_flights_array[t in timepoints] = {l | l in flights : l.depT <= t && t < (l.arrT + rotation_time_bru)};

/*
* Variabelen:
* assignment -- assignment[l][f] betekend dat flights[l] gevlogen
* wordt door een vliegtuig in fleets[f].
*/
dvar boolean assignment[fleets][flights];

minimize
sum(l in flights, f in fleets)
    Cost[f][l] * assignment[f][l];

subject to {
forall(l in flights)
    ctCover:
        sum(f in fleets)
            assignment[f][l] == 1;

forall(t in timepoints,f in fleets)
    ctAvailability:
        sum(l in time_flights_array[t])
            assignment[f][l] <= f.aircrafts;

forall(l in flights: l.haul == "Medium")
    ctHaul:
        sum(f in fleets: f.haul == "Short")
            assignment[f][l] == 0;
}

/*
* OUTPUT
* schrijf de assignments naar een textfile dat makkelijk
* ingelezen kan worden door cplex_to_ruby.rb
* dit is nodig om de visualisatie software te gebruiken
*/
execute DISPLAY {
    var ofile = new IloOplOutputFile("Z:\unief\thesis - fleet assignment\ILOG\shared_data\assignments\assignments.txt");
    forall(l in flights) {
        forall(f in fleets) {
            if(assignment[f][l] == 1){
                ofile.writeln(l.id, ":", f.ba_code);
                writeln(l.id, ":", f.ba_code);
            }
        }
    }
    ofile.close();
}

```

5.7. Swapping algoritme

broncode van het bestand “swap_algorithm.rb”:

```

require 'benchmark'
require 'config'

session_name = '7_27'
config = 1

AssignmentParameters.from_ilog("config_#{config}")

# All the extra methods required
# to run the swapping algorithm

```

```

class Assignment

# check if the swap of both flights fits in the schedule
def swap_fits?(flight_1,flight_2)
  schedule_for(flight_2).remove_flight(flight_2)
  fit = schedule_for(flight_2).fits?(flight_1)
  schedule_for(flight_2).add_flight(flight_2)
  return fit
end

# check if a swap is valid
def swap_valid?(flight_1,flight_2)
  # check range constraint
  return false if (flight_1.haul == "Medium" && flight_2.aircraft.range == "Short")
  return false if (flight_2.haul == "Medium" && flight_1.aircraft.range == "Short")
  # check schedule constraint
  return (swap_fits?(flight_1,flight_2) && swap_fits?(flight_2,flight_1))
end

# winst in kostenreductie als gevolg van de swap
def swap_profits(flight_1,flight_2)
  aircraft_1 = flight_1.aircraft
  aircraft_2 = flight_2.aircraft
  delta_1 = flight_1.cost(aircraft_1) - flight_1.cost(aircraft_2)
  delta_2 = flight_2.cost(aircraft_2) - flight_2.cost(aircraft_1)
  return (delta_1 + delta_2)
end

def do_swap!(flight_1,flight_2)
  @swap_counter += 1
  print '!'
  STDIN.flush
  # save previous schedules
  sched_1 = schedule_for(flight_1)
  sched_2 = schedule_for(flight_2)
  # remove flights
  sched_2.remove_flight(flight_2)
  sched_1.remove_flight(flight_1)
  # add flights
  sched_1.add_flight(flight_2)
  sched_2.add_flight(flight_1)
  # swap assigned aircraft type
  aircraft_2 = flight_2.aircraft
  flight_2.assign_aircraft(flight_1.aircraft)
  flight_1.assign_aircraft(aircraft_2)
end

# return the possible swaps for a flight, sorted by profitability
def swaps_for_flight(flight_1)
  possible_swaps = []
  @flights.each do |flight_2|
    if ((profits = swap_profits(flight_1,flight_2)) > 1) && swap_valid?(flight_1,flight_2)
      possible_swaps << [profits,flight_2]
    end
  end
  return nil if possible_swaps.empty?
  possible_swaps.sort_by(&.first)
end

# execute the most profitable swap for a given flight
def best_swap_for_flight(flight_1)
  swaps = swaps_for_flight(flight_1)
  do_swap!(flight_1,swaps.last.last) if swaps
end

# do a full swap pass and return the number of swaps executed
def full_swap_pass
  @swap_counter = 0
  @flights.each do |flight|
    best_swap_for_flight(flight)
  end
  return @swap_counter
end

@pass_counter = 0
@pass_results = []

# Shortcut to keep the execution time of a swapping pass
def do_a_pass(assignment)
  start_time = Time.now
  swap_count = assignment.full_swap_pass
  @pass_results << { :pass => @pass_counter+=1,:swap_count => swap_count,
    :time_ran => Time.now - start_time,:results => assignment.results}
end

Benchmark.bm(12) do |bench|

```

```

bench.report("Config:") do
  require 'config'
end
bench.report("Loading:") do
  @assignment = Assignment.new(load_from_yaml("data/assignments/#{session_name}_original.yml"))
end
bench.report("Schedule:") do
  @assignment.schedule!
  @pass_results <<{:pass => 'original',:results => @assignment.results}
end
@x = 0
begin
  bench.report("Pass #{@x+=1}\n") do
    do_a_pass(@assignment)
  end
end until @pass_results.last[:swap_count] < 1
bench.report("WriteYaml:") do
  write_to_yaml(@assignment.flights,"data/assignments/#{session_name}_conf#{config}_swapped.yml")
  write_to_yaml(@pass_results,"data/progress_reports/#{session_name}_conf#{config}_swapped.yml")
end
end

```

5.8. Monte-Carlo 2 broncode

broncode van het bestand "monte_carlo_2.rb"

```

require 'config'

class MonteCarlo
  attr_accessor :possibility_distribution
  # geef random een waarde uit de array van mogelijkheden
  def mc
    rnd = rand(total_count)
    @possibility_distribution.detect{ |pos| (rnd -= pos.last) <= 0 }.first
  end

  # tel het totaal aantal 'frequencies'
  def total_count
    @total_count ||= @possibility_distribution.inject(0) { |sum, n| sum + n.last }
  end

  # haal de kansverdeling uit de database voor gegeven input range
  def self.from_db(min_seat_sold,max_seat_sold)
    sql = 'select seats_sold(l.id,0) as "0d",count(*) from sales_legs l,sales_ticks t
      where l.id = t.sales_leg_id and t.date = l.date - 28 and t.seats_sold
      BETWEEN '+min_seat_sold.to_s+' and '+max_seat_sold.to_s+' group by "0d" order by "0d"'
    mc = MonteCarlo.new
    mc.possibility_distribution = SalesLeg.connection.select_all(sql).collect{ |t| [{"0d":t.to_i,t["count"]}] }
    return mc
  end
end

# MonteCarloGroup is een object die alle MonteCarlo generators voor elke input range bijhoudt
class MonteCarloGroup
  attr_accessor :monte_carlos, :stap, :aantal

  # haal de probabiliteitsdistributie voor <aantal> groepen van <stap>
  # en sla ze op in de @monte_carlos array
  def initialize(stap=5,aantal=34)
    @stap = stap
    @monte_carlos = []
    aantal.times do |x|
      van = x*stap
      tot = (x+1)*stap-1
      @monte_carlos[tot] = MonteCarlo.from_db(van,tot)
    end
  end

  def rnd_mc(verkochte_zetels_28d)
    # bereken de juiste mc index
    idx = (verkochte_zetels_28d / @stap).ceil*@stap-1
    @monte_carlos[idx].mc
  end
end

# Functies toevoegen aan Array's
class Array
  # bereken het gemiddelde
  def avg
    self.inject(:+) / self.size.to_f
  end
end

```

```

# bereken de standaard deviatie
def std_dev
  count = self.size
  mean = self.avg
  return Math.sqrt( self.inject(0) { |sum, el| sum + (el - mean) ** 2 } / count.to_f )
end
end

puts "Distributies ophalen voor MonteCarlo"
@mc_group = MonteCarloGroup.new

# functie die de procentuele kostenreductie berekent
def gain(new_total_cost,original_total_cost = @original_total_cost)
  ((original_total_cost - new_total_cost) / original_total_cost.to_f) * 100
end

# flights inladen
session_name = '7_27_conf1'
AssignmentParameters.from_ilog('config_1')
flights = load_from_yaml("data/assignments/#{session_name}_cplex.yml")
@assignment = Assignment.new(flights)

# toon de originele results
print "original results: "
original_result = @assignment.results(true)
puts @original_total_cost = original_result[:total_cost]

print "assigned results: "
assigned_result = @assignment.results
puts assigned_total_cost = assigned_result[:total_cost]

print "original gain: "
puts gain(assigned_total_cost)

#####
# do the montecarlo #
#####

numbers = []
10000.times do |counter|
  flights.each do |f|
    # MonteCarlo Randomized het werkelijk aantal passagiers van elke vlucht
    f.pax_1 = @mc_group.rnd_mc(f.pax_1_28d)
    f.pax_2 = @mc_group.rnd_mc(f.pax_2_28d)
  end
  @assignment.flights = flights
  numbers << gain(@assignment.results[:total_cost],@assignment.results(true)[:total_cost])
  puts "#{$(counter)}" if counter % 500 == 0 # progress indicatie
end

# Toon basic statistieken
puts "Minimum: #{numbers.min}"
puts "Maximum: #{numbers.max}"
puts "Average: #{numbers.avg}"
puts "StdDev: #{numbers.std_dev}"

# Schrijf alle nummers naar een textfile (voor de grafieken)
nr_file = 'data/montecarlo/montecarlo_results_2.txt'
puts "writing the numbers to #{nr_file}"
File.open(nr_file, 'w') { |f| f.write(numbers.join("\n")) }
puts "done"

```

5.9. Overige scripts

Ruby code voor het ophalen van de lengtegraad, breedtegraad, stad, land en naam van een luchthaven. Werd gebruikt om figuur 3.7 op p24 te tekenen.

```

def get_additional_data
  require 'hpricot'
  require 'open-uri'
  doc = Hpricot(open("http://gc.xls2.com/airport/#{$(iata_code)}",
    "User-Agent" => "Ruby"))
  self.lat = doc.at("abbr.latitude")["title"].to_f
  self.lng = doc.at("abbr.longitude")["title"].to_f
  self.city = doc.at("span.locality").inner_html
  self.country = doc.at("span.country-name").inner_html
  self.name = doc.at("td.fn").inner_html.gsub(/<[^>]+>/, "")
end

```

