

# Revenue management in de luchtvaart met stochastische vraag



Fiorinda Polselli

  
**Universiteit Twente**  
*de ondernemende universiteit*

**ORTEC**

# Revenue management in de luchtvaart met stochastische vraag

Fiorinda Polselli  
str: 9801871  
januari 2004

Werner Scheinhardt  
Universiteit Twente  
Faculteit EWI - Afdeling Toegepaste Wiskunde  
Stochastische Operationele Research  
Drienerlolaan 5  
Enschede

Maurice Bosch  
ORTEC bv  
Afdeling ATM  
Groningerweg 6-33  
Gouda

  
**Universiteit Twente**  
*de ondernemende universiteit*

**ORTEC**

---

## Samenvatting

Om de winst te maximaliseren, maken veel vliegtuigmaatschappijen gebruik van revenue management. Een onderdeel hiervan is het zogenaamde seat-allocation probleem. Dit is het probleem van het toewijzen van zitplaatsen aan verschillende groepen passagiers.

Op dit moment heeft ORTEC een programma waarbij gebruik wordt gemaakt van een deterministisch model om dit laatste probleem op te lossen. In werkelijkheid is echter niet zeker hoe groot de vraag naar de verschillende soorten tickets zal zijn. Daarom is in deze scriptie gekeken naar een methode om deze onzekerheid van de vraag mee te nemen. Naast het stochastische element wordt ook rekening gehouden met het netwerkeffect, de toepassing van nesting en de mogelijkheid tot upgrading.

De modellen waarbij rekening gehouden wordt met de onzekerheid zijn vormen van stochastisch programmeringsmodellen met recourse. Door de vraagverdeling te beschouwen als een eindige discrete verdeling zijn de modellen vervolgens omgeschreven tot lineair programmeringsproblemen.

Met behulp van simulaties is gekeken hoe de nieuwe modellen presteren ten opzichte van elkaar en het originele deterministische model. Bij deze simulaties is gebruik gemaakt van een netwerk dat representatief is voor dat van vier dagen van een bestaande vliegtuigmaatschappij. Uit deze simulaties kan geen eenduidig beeld gegeven worden over het nut van het toevoegen van nesting aan de stochastische modellen. Wel blijkt dat het toevoegen van de onzekerheid van de vraag een positief effect heeft op de behaalde opbrengst.

---

---

## Inhoudsopgave

Voorwoord.....	4
<b>1. Inleiding.....</b>	<b>5</b>
1.1 De context van de opdracht.....	5
1.2 ORTEC.....	5
1.3 Indeling scriptie.....	7
<b>2. Probleemomschrijving.....</b>	<b>8</b>
2.1 Revenue management in het algemeen.....	8
2.2 Revenue management in de luchtvaart.....	8
2.3 De probleemstelling van deze scriptie.....	11
<b>3. Modelling van het seat inventory control probleem.....</b>	<b>12</b>
3.1 Deterministisch model.....	13
3.2 Stochastisch model.....	14
3.2.1 <i>Stochastisch programmeren</i> .....	15
3.2.2 <i>Stochastisch model zonder nesting</i> .....	19
3.2.3 <i>Stochastisch model met volledige nesting</i> .....	21
3.2.4 <i>Model met nesting per pad</i> .....	21
3.3 Deterministisch equivalent.....	22
3.3.1 <i>Model zonder nesting</i> .....	22
3.3.2 <i>Model met volledige nesting</i> .....	24
3.3.3 <i>Model met nesting per pad</i> .....	25
<b>4. Simulatie en Resultaten.....</b>	<b>27</b>
4.1 Betrouwbaarheidsintervallen.....	29
4.2 Simulaties met een statische vraag.....	31
4.2.1 <i>Discreet verdeelde vraag met drie uitkomsten</i> .....	32
4.2.2 <i>Normaal verdeelde vraag</i> .....	35
4.3 Simulaties met een dynamische vraag.....	38
4.3.1 <i>Stationair Poisson-aankomstproces</i> .....	39
4.3.2 <i>Niet-stationair Poisson-proces</i> .....	42
<b>5. Verbeteringen voor het model.....</b>	<b>46</b>
5.1 Discretisering.....	46
5.1.1 <i>Normaal verdeelde vraag</i> .....	47
5.1.2 <i>Poisson-verdeelde vraag</i> .....	49
5.2 Boekingsvolgorde.....	51
5.2.1 <i>Statisch vraagproces met Poisson-verdeelde vraag</i> .....	51
5.2.2 <i>Poisson-aankomstproces, hoge ODF's boeken eerst</i> .....	51
5.2.3 <i>Resultaten</i> .....	52
<b>6. Conclusies en aanbevelingen.....</b>	<b>53</b>
6.1 Conclusies.....	53
6.2 Aanbevelingen.....	54
<b>Veel gebruikte afkortingen en symbolen.....</b>	<b>56</b>
<b>Literatuurlijst.....</b>	<b>57</b>

---

## Voorwoord

Als afronding van mijn studie Toegepaste Wiskunde aan de Universiteit Twente, heb ik het afgelopen jaar een gecombineerde stage/afstudeeropdracht uitgevoerd bij ORTEC. Deze opdracht bestond uit het aanpassen van een model voor revenue management.

Doordat in de opdracht veel onderdelen van mijn studie terugkwamen, heb ik met plezier gewerkt aan deze opdracht. Toen ik voor de keuze stond tussen de verschillende afstudeerrichtingen, heb ik getwijfeld tussen de deterministische en de stochastische OR. Uiteindelijk was het idee dat de stochastische richting meer overeenkomt met de werkelijkheid doorslaggevend voor mijn keuze. Het was dan ook leuk om te werken met een model dat oorspronkelijk deterministisch was en te zien in hoeverre de stochastische component effect heeft op de prestaties.

Bij deze wil ik Maurice Bosch bedanken voor de begeleiding vanuit ORTEC. Hij gaf me een uitleg over revenue management en hielp me op weg met het programma waarin het bestaande model wordt opgelost. Daarnaast was hij bijna altijd beschikbaar voor vragen en voorzag hij mijn werk van commentaar.

Daarnaast wil ik Joaquim Gromicho van ORTEC bedanken voor zijn hulp. Hij heeft me op weg geholpen met het begrip stochastisch programmeren en verwees me naar literatuur waarin de theorie duidelijk beschreven werd. Daarnaast was hij bereid me te helpen met een alternatieve methode voor het oplossen van stochastische programmeringsmodellen. Hiervoor was echter niet genoeg tijd.

Ook wil ik mijn begeleider van de Universiteit Twente, Werner Scheinhardt, bedanken voor zijn opbouwende kritiek en commentaar op de tussenverslagen. Ook wil ik hem bedanken voor de tijd die hij genomen heeft om naar Gouda te komen voor een overleg.

Tenslotte wil ik nog iedereen op de afdeling ATM bedanken voor de gezellige maanden. Ze gaven me al snel het gevoel dat ik volledig deel uit maakte van de afdeling.

Fiorinda Polselli  
Gouda, 5 januari 2004

## 1. Inleiding

In dit hoofdstuk is het probleem dat in deze scriptie behandeld wordt, kort beschreven. Hierbij ligt de nadruk op de context waarbinnen dit probleem ligt. Een uitgebreide beschrijving van het probleem is te vinden in hoofdstuk 2. Daarnaast wordt een beschrijving gegeven van ORTEC, het bedrijf waarbinnen deze opdracht is uitgevoerd. Tenslotte volgt een overzicht van de indeling van de rest van de scriptie.

### 1.1 De context van de opdracht

Sinds de deregulering van de Amerikaanse vliegtuigindustrie in de jaren 70, moesten de bestaande vliegtuigmaatschappijen de concurrentiestrijd aangaan met maatschappijen die goedkope tickets aanboden. Door deze concurrentiestrijd werd het noodzakelijk om naast de dure tickets ook goedkopere aan te bieden. Op deze manier worden extra klanten aangetrokken die lege zitplaatsen opvullen. Dit roept echter een aantal vragen op. De eerste is hoe te voorkomen dat oude klanten die bereid zijn veel te betalen voor een ticket, een goedkoper ticket kopen. Daarnaast moeten de prijzen van de verschillende soorten tickets bepaald worden. Tenslotte is de vraag hoeveel goedkope tickets aangeboden moeten worden. Daarbij is het van belang dat zo min mogelijk zitplaatsen leeg blijven en tegelijkertijd zo min mogelijk klanten die bereid zijn om een duur ticket te kopen, afgewezen worden. Al deze vragen vallen onder het begrip revenue management, dat in hoofdstuk 2 nader toegelicht zal worden.

Sinds de deregulering is veel onderzoek gedaan naar methoden om de verschillende vragen te beantwoorden. Dit onderzoek heeft reeds geleid tot modellen die in de luchtvaartbranche gebruikt worden. Geen van deze modellen neemt echter alle facetten van het probleem mee. Er is dus nog steeds behoefte aan nader onderzoek op dit gebied.

In deze scriptie wordt gekeken naar het model dat gebruikt wordt door ORTEC om voor de KLM een deel van bovengenoemde problemen op te lossen. Dit model gaat er echter vanuit dat de vraag bekend is en niet afwijkt van de voorspelde waarde. De opdracht was nu om dit model dusdanig aan te passen dat onzekerheid van de vraag meegenomen kan worden.

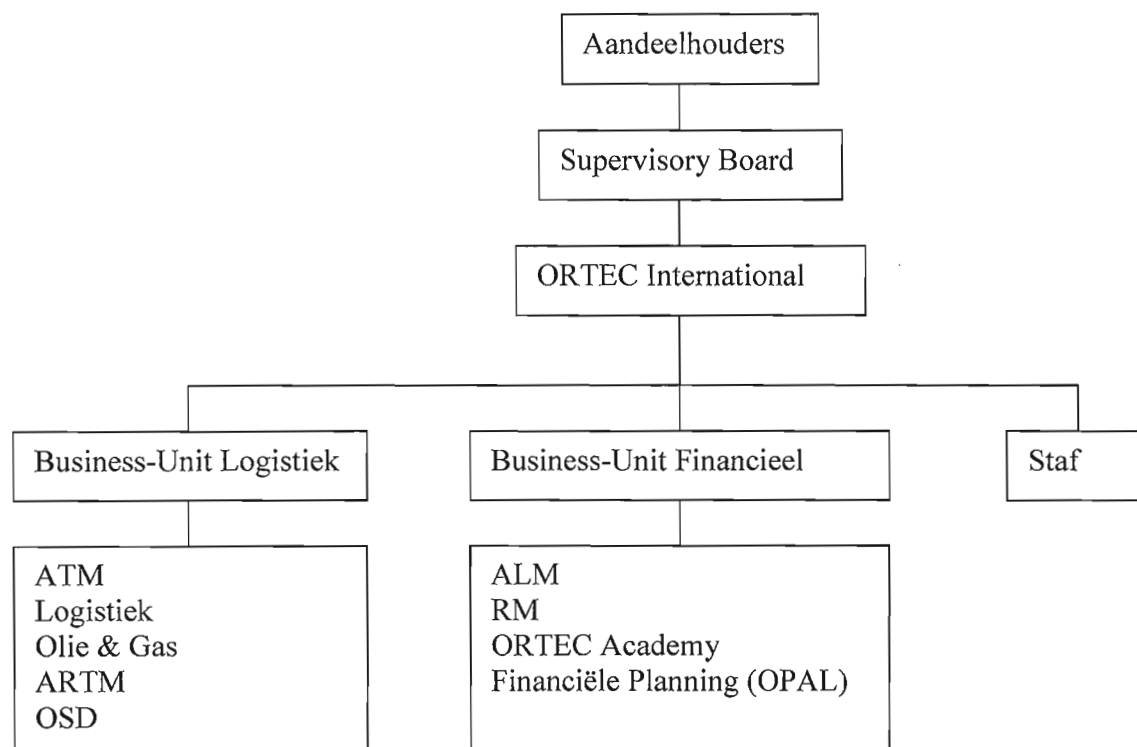
### 1.2 ORTEC

Om een indruk te geven van het bedrijf waarbinnen deze opdracht is uitgevoerd, zal in deze paragraaf een korte beschrijving van ORTEC gegeven worden.

De missie van ORTEC is om het bedrijfsleven en de overheid te ondersteunen bij strategische en operationele besluitvorming, door gebruik te maken van wiskundige modelbouw en de ontwikkeling van geautomatiseerde planningshulpmiddelen.

Het bedrijf is in april 1981 opgericht door vier studenten econometrie van de Erasmus Universiteit in Rotterdam. Inmiddels is het gegroeid van vier werknemers en een studentenkamer tot ongeveer 350 werknemers en 12 vestigingen, verdeeld over vier continenten. Van de vestigingen bevinden zich vier in Nederland, te weten in Gouda (hoofdkantoor), Rotterdam, Amsterdam en Groningen.

In onderstaande figuur wordt de bedrijfsstructuur van ORTEC weergegeven.



**Figuur 1**

De opdracht voor deze scriptie is afkomstig van de afdeling ATM en is daar ook uitgevoerd. ATM staat voor Aviation and Traffic Management. Deze afdeling houdt zich voornamelijk bezig met problemen in de luchtvaart en de spoorwegen en met revenue management. Ook is er een zalenreserveringssysteem ontwikkeld voor universiteiten. Belangrijke klanten zijn de NS, Railion en KLM.

Hieronder zijn in het kort de werkzaamheden van de andere afdelingen van Business Unit Logistiek en Business Unit Financieel beknopt weergegeven.

De afdeling Logistiek houdt zich onder andere bezig met routeringsproblemen, productie planning en load planning. De voornaamste klanten van deze afdeling zijn TPG-post, Deutsche Post, Wehkamp. Ook de (al)locatie van verschillende depots van onder andere groothandelbedrijven en verhuurbedrijven behoort tot de specialiteiten van deze afdeling.

Olie & Gas richt zich op de meest uiteenlopende projecten binnen de Olie en Gas industrie. Het gehele logistieke proces, van het winnen van de benodigde delfstoffen tot aan de aflevering van het eindproduct bij de (particuliere) klant, wordt door deze afdeling onder de loep genomen. Belangrijke klanten zijn Shell, TotalFinaElf en de NAM.

De afdeling ARTM (Arbeidstijdmanagement) is gespecialiseerd in het roosteren van personeel in verschillende sectoren. Het grootste deel van de roosters die gemaakt worden, zijn voor klanten in de gezondheidszorg.

OSD doet aan productontwikkeling, zodat er niet voor iedere klant met hetzelfde probleem een nieuw systeem bedacht hoeft te worden. Hiernaast kunnen ze ook maatwerk leveren voor klanten met specifieke problemen. Hiervoor wordt vaak samengewerkt met andere afdelingen van ORTEC.

De belangrijkste bezigheden van de financiële tak van ORTEC, gevestigd in Rotterdam en Amsterdam zijn:

ALM (Asset Liability Management). Met name voor pensioenfondsen wordt onder andere onderzocht in hoeverre ze liquide moeten zijn, hoe en waarin ze moeten beleggen en welke buffer er aangehouden moet worden, willen ze in de toekomst nog alle pensioenen uit kunnen betalen.

RM (risico management). Deze afdeling richt zich onder andere op verschillende facetten van de wereld van de woningcorporaties, zoals het opstellen van beleidsinstrumenten en het beheer van de onroerendgoedportefeuilles.

OPAL. Dit is een pensioenassetplanner die inzicht geeft in de verwachte financiële situatie bij pensionering, arbeidsongeschiktheid en overlijden. De pensioensituatie wordt zo realistisch mogelijk weergegeven.

ORTEC acadamy. De educatie en training in de geleverde producten en bedrijfsprocessen, met als doelstelling dat de programma's een bijdrage leveren bij het opzetten van planningsprocessen en beleidsoplossingen.

### 1.3 Indeling scriptie

De indeling van deze scriptie is als volgt. In het tweede hoofdstuk zal eerst een nadere beschrijving van het probleem gegeven worden. Naast een beschrijving van revenue management in het algemeen en revenue management in de luchtvaart, is in dat hoofdstuk ook het deelprobleem dat in deze scriptie behandeld wordt, beschreven.

In hoofdstuk 3 worden een aantal manieren beschreven waarop het probleem gemodelleerd kan worden. Daarnaast wordt in dit hoofdstuk een korte inleiding gegeven in stochastisch programmeren.

Vervolgens wordt in hoofdstuk 4 een beschrijving gegeven van simulaties die zijn uitgevoerd om de kwaliteit van de verschillende modellen met elkaar te vergelijken. Naast de resultaten van de simulaties zijn hierin ook de verwachte opbrengsten te vinden die verkregen worden bij het oplossen van de modellen. Daarnaast worden de simulatieresultaten van de verschillende modellen met elkaar vergeleken.

Daarna wordt in hoofdstuk 5 gekeken wat er met de resultaten gebeurt als de benadering van de vraagverdeling wordt verbeterd. Bovendien wordt gekeken wat de gevolgen voor de opbrengst zijn als de veronderstelde boekingsvolgorden niet overeenkomt met de werkelijkheid.

Tenslotte zijn in hoofdstuk 6 de conclusies van deze scriptie te vinden. Bovendien worden een aantal aanbevelingen gedaan voor verder onderzoek op het gebied van het onderwerp van deze scriptie.

Aan het eind van de scriptie (pagina 56) is een overzicht te vinden van de gebruikte afkortingen en symbolen.



## 2. Probleemomschrijving

In dit hoofdstuk is een beschrijving te vinden van het probleem dat in deze scriptie behandeld wordt. Eerst wordt verteld wat het algemene revenue management probleem is. Daarna volgt een uitleg van de toepassing van revenue management in de luchtvaart. Het hoofdstuk eindigt met een beschrijving van het specifieke probleem dat in deze scriptie behandeld wordt.

### 2.1 Revenue management in het algemeen

Revenue management is de kunst van het toewijzen van het juiste product, tegen de juiste prijs, aan de juiste klant, op het juiste moment, met als doel de omzet te maximaliseren. Het wordt toegepast bij de verkoop van producten waarvan gedurende een bepaalde periode een beperkte hoeveelheid beschikbaar is en na die periode niet meer.

In de literatuur over revenue management is sprake van zes karakteristieken waaraan producten waarop het wordt toegepast, over het algemeen voldoen.

1. *Het product is vergankelijk.*  
Dit houdt in dat het product op een bepaald moment beschikbaar komt en na dat moment niet meer beschikbaar of verouderd is.
2. *Het product is geschikt voor marktsegmentatie.*  
Dit wil zeggen dat de mogelijkheid bestaat om, aan de hand van prijsgevoeligheid, onderscheid te maken tussen verschillende groepen klanten.
3. *Het product wordt vooraf verkocht.*  
De klanten kunnen eenheden van het product reserveren voordat deze beschikbaar komen.
4. *Het aantal eenheden van het product dat beschikbaar komt is beperkt.*  
Er is een vast aantal eenheden van het product beschikbaar en aanvullende eenheden zijn niet op korte termijn en zonder hoge kosten toe te voegen.
5. *De variabele kosten van het product zijn in de meeste gevallen laag.*  
Dit houdt in dat de verkoop van een extra eenheid van het product, in verhouding tot de overige vaste kosten, weinig kost. Hierdoor komt vrijwel de gehele omzet uit de verkoop van een extra product direct ten goede aan de winst, waardoor het aantrekkelijk is om alle beschikbare capaciteit te benutten.
6. *In het algemeen varieert de vraag naar het product door de tijd heen.*  
De vraag naar het product is dus niet over de hele periode hetzelfde.

Naast deze karakteristieken blijkt voor veel producten ook te gelden dat de vraag naar het product onzeker is en dat de klanten die bereid zijn een hoge prijs te betalen de markt laat betreden.

### 2.2 Revenue management in de luchtvaart

In de luchtvaart betreft revenue management de verkoop van zitplaatsen van een cabine. Elk vliegtuig bestaat uit één of meerdere cabines (bijvoorbeeld business en economy class) die gerangschikt zijn. In deze scriptie wordt met een cabine één cabine van één vliegtuig op één bepaalde vlucht op één bepaald moment bedoeld.

Elke passagier reist volgens een bepaald pad. Dit pad is de combinaties van cabines waarin de passagier reist. Bij de verkoop van zitplaatsen worden de klanten opgedeeld in groepen met hetzelfde vertrekpunt (origin), bestemming (destination), pad, plaats van aankoop, reisdatum en

prijsklasse (fare class). Een dergelijke combinatie van vertrekpunt, bestemming, pad, aankoopplaats, reisdatum en prijsklasse wordt een ODF genoemd. Deze ODF's kunnen gerangschikt worden volgens bepaalde criteria. Een voorbeeld van een criterium is de prijs van een ticket. Een ODF met een hoge rang heeft een laag rangnummer en wordt meer gewaardeerd wordt dan een ODF met een hoger rangnummer. Als tijdens het boekingsproces de keuze gemaakt moet worden tussen twee klanten uit verschillende ODF's, zal de klant uit een ODF met een laag nummer dus het ticket krijgen, ten koste van de andere klant.

Voor elke ODF geldt dat de reis bestaat uit één of meerdere vluchten, bijvoorbeeld een reis van Amsterdam naar New York met een overstap in Parijs. Op elk van deze vluchten reist de passagier weer in een bepaalde cabine. In het vervolg van deze scriptie zal het begrip pad gebruikt worden om de verzameling cabines weer te geven waarin passagiers van een ODF reizen.

Het product dat verkocht moet worden is een zitplaats op een vlucht. Revenue management is hier toe te passen aangezien het product voldoet aan eerdergenoemde karakteristieken:

1. Het product is vergankelijk, aangezien de zitplaatsen na het vertrek van een vlucht niets meer waard zijn.
2. Uit de praktijk blijkt dat passagiers in verschillende mate gevoelig zijn voor verandering van de prijs en de beperkingen die aan een ticket worden opgelegd. Een beperking kan zijn dat het voor de ene prijsklasse wel is toegestaan om te annuleren en voor een andere (goedkopere) prijsklasse niet. Ook kunnen voorwaarden worden gesteld aan het moment waarop een ticket gereserveerd moet worden. Hieruit blijkt dat het product geschikt is voor marktsegmentatie.
3. Het product wordt vooraf verkocht. Passagiers reserveren immers de tickets voor vertrek.
4. Het aantal eenheden dat beschikbaar is, is beperkt. Bij veel vliegtuigmaatschappijen staat vooraf voor elke vlucht vast met welk vliegtuig gevlogen wordt en hoeveel zitplaatsen daarvoor beschikbaar zijn voor de cabines op die vlucht. Het aantal zitplaatsen is op korte termijn dus niet te wijzigen.
5. De variabele kosten uit de verkoop van een extra zitplaats zijn relatief laag. In verhouding tot de kosten van het laten vliegen van een vliegtuig, zijn de extra kosten van de service voor één extra persoon laag. Hierdoor is de winst maximaal als de omzet maximaal is.
6. Dat de vraag naar het product door de tijd heen varieert, is minder triviaal. In de praktijk geldt vaak dat de vraag naar tickets in het begin van de boekingsperiode niet gelijk is aan de vraag vlak voor vertrek. In het begin van de boekingsperiode bestaat de vraag vaak grotendeels uit klanten die een vakantiereis willen boeken. Deze klanten zijn gevoeliger voor prijsverschillen en hebben geen bezwaar tegen extra beperkingen. Vlak voor vertrek bestaat de vraag in veel gevallen vooral uit klanten die een zakenreis willen boeken en die liever een hogere prijs betalen voor een ticket zonder beperkende voorwaarden.

Aangezien het probleem van het verkopen van zitplaatsen op een vlucht voldoet aan de eerdergenoemde karakteristieken, kan het beschouwd worden als een revenue management probleem. De vraag is alleen hoe het opgelost moet worden. Het probleem is op te splitsen in de volgende vier deelgebieden.

1. *forecasting*  
Dit is het probleem van het voorspellen van de vraag aan de hand van geconditioneerde vraag naar vluchten uit het verleden. Dit houdt in dat alleen bekend is hoeveel tickets daadwerkelijk verkocht zijn en niet hoeveel vraag afgewezen is. Aangezien de overige deelgebieden gebruik maken van de resultaten van de forecasting, is dit een onmisbaar onderdeel. Het is immers onmogelijk om goede resultaten te krijgen als de input niet goed is.
2. *pricing*  
Bij dit probleem gaat het erom passende prijzen te vinden voor de verschillende prijsklassen en te bepalen gedurende welke periode een prijsklasse beschikbaar wordt gesteld.
3. *overboeking*  
Bij dit probleem moet voor elke cabine op een vlucht bepaald worden hoeveel tickets er meer worden verkocht dan het aantal zitplaatsen in die cabine bedraagt. Door dit te doen wordt het nadeel uit het feit dat sommige klanten die een ticket reserveren uiteindelijk niet met de vlucht mee willen, beperkt. Het risico is echter dat meer passagiers komen opdagen dan dat er zitplaatsen zijn. De passagiers die niet met het vliegtuig mee kunnen, krijgen een compensatie voor het ongemak. Dit deelgebied wordt vaak in combinatie met het volgende opgelost.
4. *seat inventory control (SIC)*  
Dit is het probleem waarbij voor iedere ODF het aantal beschikbare zitplaatsen bepaald moet worden. De oplossing hiervan moet leiden tot optimale beslissingen omtrent het accepteren of weigeren van boekingen.

Factoren die van belang kunnen zijn bij de oplossing van de deelproblemen zijn:

- **upgrading**  
Dit wil zeggen dat (een deel van de) klanten die een ticket voor een bepaalde cabine hebben gekocht, voor hetzelfde geld in een duurdere cabine op dezelfde vlucht geplaatst worden. Op deze manier kan het aantal lege zitplaatsen beperkt worden als de vraag naar ODF's in de cabine met de hoge rang laag is. Met upgrading wordt al bij het bepalen van de boekingslimieten rekening gehouden. Er worden dan meer zitplaatsen toegewezen aan de ODF's uit de cabine met de lage rang dan dat er beschikbaar is en minder zitplaatsen aan ODF's uit de cabine met de hoge rang.
- **downgrading**  
Dit is het geval als (een deel van de) klanten een goedkoper ticket kopen dan ze eigenlijk bereid zijn te kopen.
- **nesting**  
Dit houdt in dat zitplaatsen die voor een bepaalde ODF beschikbaar zijn, ook door ODF's met een hogere rang gebruikt kunnen worden.
- **batch-boekingen**  
Hiervan is sprake als één klant meerdere zitplaatsen kan reserveren.
- **netwerkeffect**  
Dit houdt in dat er niet sprake is van losstaande vluchten (single-leg), maar van vluchten die met elkaar verbonden zijn. Hiermee wordt rekening gehouden aangezien er passagiers kunnen zijn die niet rechtstreeks naar hun eindbestemming (kunnen) vliegen. Het accepteren van een dergelijke passagier kan een positieve invloed hebben op de opbrengst van de eerste vlucht, maar tegelijk een grotere negatieve invloed op die van de tweede vlucht, waar

een andere klant geweigerd moet worden. Als het netwerkeffect buiten beschouwing wordt gelaten en alleen de opbrengst van de eerste vlucht wordt geoptimaliseerd, is de totale opbrengst dus lager doordat de klant wel wordt geaccepteerd.

- **annuleringen en no-shows**

Het eerste is het geval als mensen die een ticket gereserveerd hebben, besluiten om af te zien van de reis en daarvoor geld terugkrijgen. Als iemand op het laatste moment niet komt opdagen, en geld terugkrijgt, is sprake van een no-show. Als hier geen rekening mee wordt gehouden, blijven zitplaatsen bij vertrek leeg. Overboeking kan dit echter compenseren.

- **short selling**

Dit houdt in dat een vliegtuigmaatschappij goekope tickets, die al verkocht zijn, terugkoopt om ze vervolgens duurder te verkopen.

### 2.3 De probleemstelling van deze scriptie

Het probleem dat in deze scriptie behandeld zal worden is het zogenaamde seat-allocation probleem (seat inventory control) in de luchtvaart. Dit houdt in dat de zitplaatsen van een cabine in een vliegtuig zodanig over de verschillende ODF's verdeeld moeten worden, dat de omzet maximaal is. Hierbij wordt er vanuit gegaan dat de vraagverdelingen en de prijzen reeds door middel van forecasting en pricing bepaald zijn. Het probleem is nu om voor elke ODF een zogenaamde boekingslimiet te bepalen. Als tijdens het boekingsproces het totaal aantal gereserveerde zitplaatsen boven de boekingslimiet van een ODF komt, kunnen boekingsverzoeken voor die ODF niet meer geaccepteerd worden.

Deze scriptie bevat zowel een deterministische als een stochastische methode om deze boekingslimieten te bepalen. Beide methodes houden rekening met upgrading, het netwerkeffect en nesting. De deterministische methode neemt daarnaast alleen de verwachting van de vraag mee en laat de stochastiek buiten beschouwing. Hiervoor is de methode overgenomen die reeds door ORTEC wordt toegepast en die beschreven is in De Boer (1999). De stochastische methode neemt de stochastiek wel mee. Hiervoor is nog geen methode beschreven waarbij bovengenoemde factoren beschouwd worden.

Vervolgens wordt bepaald of de stochastische methode de beste resultaten oplevert en in hoeverre deze methode beter is dan deterministische. Bovendien wordt gekeken of het zinvol is om nesting aan het model toe te voegen.

Bij de oplossing moet rekening gehouden worden met de rekentijd van het model. In de praktijk is er namelijk sprake van circa 36 miljoen ODF's gedurende een periode van een jaar. Het is daarom van belang om de modellen niet te groot te laten worden.

### 3. Modellering van het seat inventory control probleem

Het probleem is om voor alle ODF's die boekingslimieten te bepalen, waarvoor de verwachte totale opbrengst maximaal is. Voor elke ODF is de opbrengst gelijk aan het aantal verkochte tickets maal de prijs van de betreffende ODF. De totale opbrengst is nu de som van deze opbrengsten over alle ODF's.

Als de boekingslimieten bekend zijn, kan voor elke ODF het maximaal aantal tickets dat verkocht kan worden, bepaald worden. Hierbij zijn drie gevallen te onderscheiden.

1. *geen nesting*

In dit geval mogen zitplaatsen die voor een bepaalde ODF beschikbaar zijn gesteld niet door hogere klassen gebruikt worden. Er geldt dan dat het aantal verkochte tickets voor een bepaalde ODF altijd kleiner of gelijk is aan de boekingslimiet voor die klasse.

2. *volledige nesting*

Hierbij kunnen voor een bepaalde cabine de zitplaatsen die voor een ODF beschikbaar zijn gesteld ook door alle ODF's met een hogere rang in die cabine, gebruikt worden. Als, zoals is deze scriptie aangenomen, hogere klassen altijd na de lagere klassen boeken, geldt dan het volgende. Het aantal verkochte tickets voor een bepaalde ODF is maximaal de boekingslimiet plus het minimum van het aantal zitplaatsen dat de lagere klassen in een cabine onbenut laten, over alle cabines van het pad van de ODF.

3. *nesting per pad*

Dit is een tussenweg tussen het geval zonder nesting en die met gehele nesting. Bij nesting per pad wordt nesting toegepast tussen ODF's die gedurende de hele reis op hetzelfde moment gebruik maken van dezelfde cabines. Als ODF's die over een bepaald pad reizen beschikbare zitplaatsen leeg laten, dan mogen ODF's met een hogere rang die over hetzelfde pad reizen, deze zitplaatsen gebruiken. In dit geval is het aantal tickets dat in een bepaalde ODF verkocht wordt kleiner dan of gelijk aan het aantal zitplaatsen dat voor die ODF beschikbaar is gesteld, plus het aantal zitplaatsen dat lagere klassen op hetzelfde pad onbenut laten. Bij deze vorm van nesting is het niet langer nodig om afzonderlijk naar elke cabine van een ODF te kijken, maar volstaat het te kijken naar het pad als geheel.

Bij de modellering kan men op twee manieren te werk gaan. Ten eerste kan worden aangenomen dat precies bekend is wat de vraag naar de verschillende ODF's zal zijn. Op deze manier ontstaat een deterministisch model. In werkelijkheid is echter vooraf niet bekend wat de vraag naar de verschillende ODF's zal zijn. Een tweede mogelijkheid is om wel rekening te houden met de onzekerheid van de vraag. Het model dat hiermee verkregen wordt is een stochastisch model. In dit stochastische model is het enige stochastische element de vraag naar de ODF's. De prijzen en de capaciteiten, het totale aantal zitplaatsen per cabine, staan vast.

Voor beide mogelijkheden is er vanuit gegaan dat upgrading toegepast wordt.

Aangezien ORTEC op dit moment een model in de vorm van een deterministisch lineair programmeringsmodel (LP) gebruikt, is van een dergelijk model uitgegaan om het stochastisch te maken. In De Boer (1999) is een soortgelijk probleem opgelost waarbij zowel een deterministisch als stochastisch model gegeven is. Aangezien het deterministische model hetzelfde is als dat bij ORTEC gebruikt wordt, is het stochastische geval, net als bij De Boer, als een stochastisch programmeringsmodel gemodelleerd. Bovendien geeft het de aanleiding om door middel van zogenaamde recourse variabelen het stochastische model om te schrijven naar een duidelijk model. In tegenstelling tot De Boer (1999) wordt in deze scriptie nesting wel meegenomen in het model. Hierdoor kan alleen de opzet van het stochastische model met recourse gebruikt worden, maar is het model zelf niet bruikbaar.

In het vervolg van dit hoofdstuk zullen de volgende zeven modellen besproken worden.

3.1 *deterministisch model* (model 1)

3.2 *stochastisch model*

3.2.2 zonder nesting (model 3)

3.2.3 met volledige nesting (model 4)

3.2.4 met nesting per pad (model 5)

3.3 *deterministische equivalenten van de stochastische modellen*

Dit zijn de modellen die verkregen worden door aan te nemen dat de vraag een beperkt aantal waarden aan kan nemen

3.3.1 zonder nesting (model 6)

3.3.2 met volledige nesting (model 7)

3.3.3 met nesting per pad (model 8)

Daarnaast is in paragraaf 3.2 naast een beschrijving van de stochastische modellen ook een korte inleiding te vinden over stochastisch programmeren.

### **3.1 Deterministisch model**

Voor het deterministische model wordt aangenomen dat voor elke ODF de vraag bekend is. Hierdoor heeft het geen zin om meer zitplaatsen beschikbaar te stellen dan deze vraag. Er zou dan immers geen vraag zijn naar die zitplaatsen, waardoor ze leeg blijven. Daarom wordt de bovengrens van het aantal beschikbare zitplaatsen gelijkgesteld aan de vraag, waardoor de vraag altijd groter of gelijk is aan het aantal beschikbare zitplaatsen.

Doordat de vraag altijd groter of gelijk is aan het aantal zitplaatsen dat beschikbaar is, zal het aantal verkochte tickets per definitie gelijk zijn aan de boekingslimiet. Er zullen dus nooit zitplaatsen overblijven voor ODF's met een hogere rang. Hierdoor is het niet nodig om onderscheid te maken tussen het geval waarbij nesting niet wordt toegepast en dat waarbij het wel wordt toegepast.

Het model dat nu verkregen wordt, wordt gegeven in de Boer (1999) en ziet er als volgt uit.

$$\max_x \sum_j f_j x_j \quad (1.1)$$

$$\text{s.t.} \quad \sum_{j \in J(h)} x_j + s_l \leq c_h \quad \forall h \text{ (cabine met een hoge rang)} \quad (1.2a)$$

$$\sum_{j \in J(l)} x_j - s_l \leq c_l \quad \forall l \text{ (cabine met een lage rang)} \quad (1.2b)$$

$$0 \leq x_j \leq d_j \quad \forall j \quad (1.3)$$

- met:  $f_j$  : prijs van ODF  $j$   
 $x_j$  : boekingslimiet van ODF  $j$   
 $c_i$  : aantal beschikbare zitplaatsen op cabine  $i$   
 $d_j$  : netto vraag naar ODF  $j$   
 $J(i)$  : de verzameling van alle ODF's in cabine  $i$   
 $s_l$  : aantal passagiers van cabine  $l$  worden overgeplaatst naar cabine  $h$ .

Doelfunctie (1.1) is gelijk aan de totale opbrengst.

Voorwaarde (1.2a) geeft aan dat er in totaal nooit meer zitplaatsen beschikbaar gesteld mogen worden dan dat er in een cabine met een hoge rang nog beschikbaar is na upgrading.

Voorwaarde (1.2b) geeft aan dat er in totaal nooit meer zitplaatsen beschikbaar gesteld mogen worden dan dat er na upgrading beschikbaar zijn voor ODF's uit de cabine met lage rang.

Voorwaarde (1.3) geeft de onder- en bovengrenzen van de beslissingsvariabelen weer. Het geeft weer dat het aantal zitplaatsen dat beschikbaar gesteld wordt nooit negatief is en altijd kleiner of gelijk is aan de vraag.

De beslissingsvariabelen van het model zijn de boekingslimieten. De optimale boekingslimieten zijn te bepalen door het model op te lossen. Dit kan bijvoorbeeld met het programma CPLEX, een programma waarmee LP-modellen opgelost kunnen worden.

Om te controleren of de modellen niet te groot worden, zal voor de stochastische modellen het aantal voorwaarden en beslissingsvariabelen vergeleken worden met die van het deterministische geval. Daarom zal hieronder bepaald worden wat het aantal variabelen en voorwaarden voor dit model is.

Voor het aantal voorwaarden worden de onder- en bovengrenzen buiten beschouwing gelaten. De voorwaarden in dit model komen nu voort uit door voorwaarde (1.2). Voorwaarde (1.3) levert onder- en bovengrenzen en wordt dus niet beschouwd. Het aantal voorwaarden dat dit model bevat is dus gelijk aan het totaal aantal cabines,  $I$ .

De beslissingsvariabelen zijn de boekingslimieten. Het aantal beslissingsvariabelen is dus gelijk aan  $J$ , het aantal ODF's.

### 3.2 Stochastisch model

Aangezien de vraag in de praktijk niet exact bekend is, is het aannemelijk dat de resultaten van de optimalisatie beter zullen zijn als de onzekerheid van de vraag wel mee wordt genomen. Om dit te bereiken moet het deterministische model aangepast worden tot een stochastisch model.

Hieronder volgt eerst een korte inleiding over stochastisch programmeren. Een uitgebreidere beschrijving van de theorie is te vinden in Kall en Wallace (1994) en Birge en Louvreaux (1997), waarop deze inleiding gebaseerd is. Daarna worden de uiteindelijke stochastische modellen en de wijze waarop deze verkregen zijn, besproken. Hierbij wordt onderscheid gemaakt tussen het model zonder nesting, het model met volledige nesting en het model met nesting per pad.

### 3.2.1 Stochastisch programmeren

Aangezien het deterministische model een speciaal geval is van een mathematisch programmeringsmodel, ligt het voor de hand om het geval met een stochastische vraag te programmeren als een stochastisch programmeringsmodel. Het is dus van belang om te weten wat een stochastisch programmeringsmodel is en hoe het opgelost wordt. Hieronder volgt daarom eerst een beschrijving van stochastisch programmeren en de bijbehorende oplossingsmethoden.

#### Stochastische programmeringsmodellen

Gedurende de laatste vijf decennia heeft een grote ontwikkeling plaatsgevonden op het gebied van het oplossen van mathematische programmeringsmodellen. Dit zijn modellen van de volgen-

$$\begin{aligned} \min_x \quad & g_0(x) \\ \text{s.t.} \quad & g_i(x) \leq 0 \quad i = 1, \dots, m \\ & x \in X \subset \mathbb{R}^n \end{aligned}$$

de vorm.

Hierin geeft  $g_0(x)$  de doelfunctie weer die geminimaliseerd moet worden en  $g_i(x) < 0$  de voorwaarden waaraan oplossing  $x$  moet voldoen.

In de werkelijkheid is het echter in veel gevallen niet reëel om te veronderstellen dat de functies  $g_i(x)$  en de verzameling oplossingen,  $X$ , deterministisch zijn. In die gevallen is het bovengenoemde model waarbij de verwachtingswaarden worden ingevuld, ongeschikt om het probleem goed mee te beschrijven. Om dit te verhelpen kan men overgaan op stochastisch programmeringsmodellen.

Door één of meerdere parameters stochastisch te veronderstellen, ontstaat het volgende probleem.

$$\begin{aligned} \min_{x, \tilde{\xi}} \quad & E_{\tilde{\xi}} g_0(x, \tilde{\xi}) \\ \text{s.t.} \quad & g_i(x, \tilde{\xi}) \leq 0 \quad i = 1, \dots, m \\ & x \in X \subset \mathbb{R}^n \end{aligned}$$

Hierbij is  $\tilde{\xi}$  een stochastische vector en zijn  $g_i(x, \cdot)$  zelf ook stochastische variabelen.

Aangezien de betekenis de voorwaarden in dit probleem niet duidelijk zijn, is dit probleem niet goed gedefinieerd. Het is daarom nodig om het model om te schrijven naar een equivalent model dat wel een duidelijke betekenis heeft. In de volgende paragrafen worden twee manieren gegeven waarop een dergelijk model omgeschreven kan worden naar een model met een duidelijke betekenis. Allereerst zal de probabilistic constrains methode beschreven worden. De tweede methode is een methode waarbij recourse variabelen aan het model worden toegevoegd.



Probabilistic constraints

Bij deze methode mag de voorwaarde met een bepaalde kans  $\alpha$  overtreden worden en staat er geen boete op deze overtreding. Een voorbeeld van een toepassing van deze methode is als een leverancier wil dat maximaal een bepaald percentage van de gevraagde producten niet geleverd kan worden. Deze methode is niet geschikt voor het onderwerp van deze scriptie, aangezien geen rekening wordt gehouden met de gevolgen voor de opbrengst als de vraag lager is dan verwacht. Daarom worden de oplossingsmethoden hiervan niet hier beschreven en wordt alleen een korte beschrijving gegeven van het algemene model dat deze methode oplevert. Voor oplossingsmethoden wordt verwezen naar Kall en Wallace (1994) en Birge en Louvreaux (1997).

Bij de probabilistic constraints methode wordt de voorwaarde  $g_i(x, \tilde{\xi}) \leq 0$  in het oorspronkelijke stochastische model vervangen door voorwaarde  $P(\{\tilde{\xi} \mid g_i(x, \tilde{\xi}) \leq 0\}) \geq \alpha$ .

Een punt  $\hat{x}$  aan de voorwaarden voldoet als geldt:  $\hat{x} \in X$  en  $\hat{x} \in \bigcap_{\zeta \in G} \{x \mid g(x, \zeta) \leq 0\}$  waarbij  $G$  een deelverzameling is van alle mogelijke realisaties voor  $\tilde{\zeta}$  met  $P(G) \geq \alpha$ . In woorden staat hier dat  $\hat{x}$  een element is uit de verzameling waarvoor voor elke realisatie uit  $G$  geldt dat aan de oorspronkelijke voorwaarde wordt voldaan. Het punt voldoet dan aan de voorwaarden aangezien de stochastische variabele met een kans groter of gelijk aan  $\alpha$  een waarde uit  $G$  behaalt, waarmee aan de voorwaarde wordt voldaan. In de overige gevallen is het niet zeker of aan de voorwaarde wordt voldaan, maar dat maakt niet meer uit.

De verzameling van zichtbare punten kan dus als volgt weergegeven worden:

$$B(\alpha) = \bigcup_{G \in \tilde{\zeta}} \bigcap_{\zeta \in G} \{x \mid g(x, \zeta) \leq 0\}.$$

Het model dat nu verkregen wordt, is:

$$\min_x \quad E_{\tilde{\xi}} g_0(x, \tilde{\xi})$$

$$\text{s.t.} \quad x \in B(\alpha) \subset X \subset \mathbb{R}^n$$

Toevoeging van recourse variabelen

Bij deze methode worden zogenaamde tweede-fase recourse variabelen ingevoerd om een probleem te krijgen waarvan de betekenis wel duidelijk is. Deze variabelen,  $y_i$ , geven aan in hoeverre niet aan de voorwaarde is voldaan en voldoen daarom aan de volgende ongelijkheid:

$$g_i(x, \xi) - y_i(\xi) \leq 0$$

Indien niet aan de voorwaarden wordt voldaan, brengen deze recourse variabelen kosten met zich mee. Als de kosten van één eenheid  $y_i$  gelijk zijn aan  $q_i$ , dan wordt het probleem met recourse kosten  $Q(x, \xi)$  als volgt:

$$\min_x \quad E_{\tilde{\xi}} g_0(x, \tilde{\xi}) + Q(x, \tilde{\xi})$$

$$\text{s.t.} \quad x \in X \subset \mathbb{R}^n$$

$$\text{met:} \quad Q(x, \tilde{\xi}) = \min_y \left\{ \sum_{i=1}^m q_i y_i(\tilde{\xi}) \mid y_i(\tilde{\xi}) \geq g_i(x, \tilde{\xi}), i = 1, \dots, m \right\}$$

Het twee-fasen (two-stage) probleem kan, indien nodig, uitgebreid worden tot een multi-fase recourse probleem. Multi-fase recourse wordt toegepast als er meerdere stadia zijn waarin beslissingen worden genomen en waarbij de beslissingen afhangen van de realisaties van voorgaande stadia. In plaats van de beslissingen  $x$  en  $y$ , is er nu sprake van  $K+1$  opeenvolgende beslissingen  $x_0, x_1, \dots, x_K$  die genomen moeten worden in fase  $\tau = 0, 1, \dots, K$ . In fase  $\tau$  ( $\tau \geq 0$ ) zijn zowel de realisaties  $\xi_1, \xi_2, \dots, \xi_\tau$  als de vorige beslissingen  $x_0, x_1, \dots, x_{\tau-1}$  bekend en moet beslissing  $x_\tau$  bepaald worden. Voor deze fase krijg je dan de volgende recourse functie.

$$Q_\tau(x_0, x_1, \dots, x_{\tau-1}, \xi_1, \dots, \xi_\tau) = \min_{x_\tau} \{q_\tau(x_\tau) \mid q_\tau(x_0, \dots, x_{\tau-1}, \xi_1, \dots, \xi_\tau)\}$$

$$\min_x \quad E_{\xi} g_0(x, \xi) + \sum_{\tau=1}^K Q_\tau(x_0, \hat{x}_1, \dots, \hat{x}_{\tau-1}, \tilde{\xi}_1, \dots, \tilde{\xi}_\tau)$$

$$\text{s.t.} \quad x \in X \subset \mathbb{R}^n$$

$$\text{met:} \quad Q_\tau(x_0, x_1, \dots, x_{\tau-1}, \tilde{\xi}_1, \dots, \tilde{\xi}_\tau) = \min_{x_\tau} \{q_\tau(x_\tau) \mid q_\tau(x_0, \dots, x_{\tau-1}, \tilde{\xi}_1, \dots, \tilde{\xi}_\tau)\}$$

Het multi-fase probleem is dan als volgt.

Met behulp van de recourse variabelen wordt dus een model verkregen waarvan de minimalisatie en de voorwaarden wel duidelijk zijn. De vraag is nu nog hoe dit model opgelost kan worden.

Stochastisch lineair programmeringsprobleem met twee fasen

Een speciaal geval van een stochastisch programmeringsprobleem is een stochastisch lineair programmeringsprobleem. Dit is een stochastisch programmeringsprobleem dat in de volgende

$$\text{"min"} \quad E_{\xi} f^T x$$

$$\text{s.t.} \quad Ax \leq c$$

$$T(\tilde{\xi})x \leq h(\tilde{\xi})$$

$$x \geq 0$$

$$\text{met:} \quad T(\xi) = \hat{T}^0 + \xi_1 \hat{T}^1 + \dots + \xi_k \hat{T}^k \quad \text{en } \hat{T}^0, \hat{T}^1, \dots, \hat{T}^k \text{ deterministische matrices}$$

$$h(\xi) = \hat{h}^0 + \xi_1 \hat{h}^1 + \dots + \xi_k \hat{h}^k \quad \text{en } \hat{h}^0, \hat{h}^1, \dots, \hat{h}^k \text{ deterministische vectoren}$$

vorm te schrijven is.

De verzameling  $X \subset \mathbb{R}^n$  is dan dus de verzameling gegeven door  $X = \{x \in \mathbb{R}^n \mid Ax = c, x \geq 0\}$

Als door middel van lineaire recourse ervoor gezorgd wordt dat overtreding van de voorwaarden extra geld kost, wordt het volgende probleem verkregen:

$$\min_x \quad E_{\xi} (f^T x + Q(x, \xi))$$

$$\text{s.t.} \quad Ax \leq c$$

$$x \geq 0$$

$$\text{met} \quad Q(x, \xi) = \min_y \{q^T y \mid Wy \leq h(\xi) - T(\xi)x, y \geq 0\}$$

De matrix  $W$  geeft hierin aan hoeveel van een eenheid van de recourse nodig is om een overtreding van de voorwaarde goed te maken.

In tegenstelling tot het oorspronkelijke model hebben het minimum en de voorwaarden in het bovenstaande model wel een duidelijke betekenis. Het is nu dus mogelijk om te zoeken naar een oplossing voor bovenstaand probleem.

Oplossen van een twee-fasen stochastisch lineair program met discrete verdeling

Als de verdeling van de stochastische variabelen discreet is met een eindig aantal uitkomsten, dan is het stochastische lineair programmeringsmodel om te schrijven tot een deterministisch LP-model.

Om het model om te schrijven wordt voor elke mogelijke realisatie van het rechterlid een verzameling met recourse variabelen gemaakt. Bij elk van deze verzamelingen horen de bijbehorende voorwaarden van het stochastische model, waarin voor het rechterlid de realisatie ingevuld wordt die hoort bij die recourse variabelen. In de doelfunctie worden de oorspronkelijke recourse variabelen vervangen door de som van de kans op een uitkomst maal de bijbehorende recourse variabele, over alle mogelijke uitkomsten.

Hieronder is een voorbeeld gegeven van een stochastisch model in matrixnotatie met het bijbehorende deterministisch equivalent.

Stochastisch model		Deterministisch equivalent
$\min_x \quad f^T x + E_{\xi} q^T y$	→	$\min_x \quad f^T x + \sum_{k=1}^r p_k q^T y^k$
<p>s.t. <math>Ax = c</math></p> <p style="margin-left: 2em;"><math>Wy = h(\xi) - T(\xi)x</math></p>		<p>s.t. <math>Ax = c</math></p> <p style="margin-left: 2em;"><math>Wy^k = h(\xi^k) - T(\xi^k)x \quad k = 1, \dots, r</math></p>
<p>met: <math>r</math> : het aantal uitkomsten</p> <p style="margin-left: 2em;"><math>p_k</math> : de kans op realisatie <math>k</math></p> <p style="margin-left: 2em;"><math>y^k</math> : de recourse variabelen die horen bij realisatie <math>\xi^k</math>.</p>		

Indien de verdeling van de stochastische variabele niet deterministisch maar continu is, dan kan deze verdeling benaderd worden door een discrete verdeling met een eindig aantal uitkomsten. Dit kan gedaan worden op de volgende manier.

- 1 Bepaal een 99 %-interval. Dit is dat interval waarvoor geldt dat met een zekerheid van 99 % een trekking uit de gegeven verdeling er binnen valt.
- 2 Genereer een groot aantal  $N$ , bijvoorbeeld 10.000, trekkingen,  $\xi^n$ , uit de gegeven verdeling, beperkt tot het 99 %-interval.
- 3 Kies een opdeling van het 99 %-interval in gelijke subintervallen.
- 4 Bereken van elk subinterval het gemiddelde van de trekkingen in dat interval.
- 5 Bereken voor elk interval de relatieve frequentie, het aantal trekkingen in het subinterval gedeeld door het totaal aantal trekkingen. Dit is een schatting voor de kans dat een willekeurige trekking in het subinterval zit.

Als de discrete benadering van de verdeling van de stochastische variabele bekend is, kan deze ingevuld worden in het bovenstaande model. Op deze manier wordt een LP-model verkregen dat opgelost kan worden.

Voor het onderwerp van deze scriptie geldt dat het overtreden van de voorwaarde leidt tot het leeg blijven van zitplaatsen. Dit heeft weer tot gevolg dat de opbrengst lager is, aangezien minder tickets verkocht worden dan verwacht. Er is daarom voor gekozen om de methode met recourse variabelen toe te passen op het revenue management probleem. In de paragrafen 3.2.2, 3.2.3 en 3.3.3 geven we weer op welke manier deze methode is toegepast.

### 3.2.2 Stochastisch model zonder nesting

Zoals al vermeld in paragraaf 3.2.1, moeten recourse variabelen toegevoegd worden om ervoor te zorgen dat een model ontstaat met een duidelijke betekenis. De recourse variabelen moeten ervoor zorgen dat rekening wordt gehouden met de kans dat beschikbare zitplaatsen niet gebruikt worden. In eerste instantie is daarom gekozen voor het gebruik van de recourse variabele  $\tilde{y}_j$ , het aantal zitplaatsen dat beschikbaar is gesteld voor ODF  $j$ , maar niet door die ODF gebruikt wordt. In de doelfunctie moet nu de verwachting van de opbrengst die gemist wordt, meegenomen worden. Later zal dit model aangepast worden, om ervoor te zorgen dat nesting eenvoudig toegevoegd kan worden.

Het model dat ontstaat door recourse variabelen  $\tilde{y}_j$  toe te voegen is als volgt.

$$\max \quad \sum_j f_j x_j - E \sum_j f_j \tilde{y}_j \quad (2.1)$$

$$\text{s.t.} \quad \sum_{j \in J(h)} x_j + s_h \leq c_h \quad \forall h \text{ (cabine met een hoge rang)} \quad (2.2a)$$

$$\sum_{j \in J(l)} x_j - s_l \leq c_l \quad \forall l \text{ (cabine met een lage rang)} \quad (2.2b)$$

$$\tilde{y}_j \geq x_j - D_j \quad \forall j \quad (2.3)$$

$$0 \leq x_j \quad \forall j \quad (2.4)$$

$$0 \leq \tilde{y}_j \leq x_j \quad \forall j \quad (2.5)$$

met:

$\tilde{y}_j$ : aantal zitplaatsen in ODF  $j$  die niet verkocht worden.

$D_j$ : de stochastische vraag.

Doelfunctie (2.1) is gelijk aan de verwachting van de totale opbrengst.

Voorwaarde (2.2a) en (2.2b) geven aan dat men niet meer zitplaatsen beschikbaar kan stellen dan er na upgrading voor klanten uit de hoge of lage cabine beschikbaar is.

Voorwaarde (2.3) geeft aan dat het aantal zitplaatsen dat leeg blijft groter of gelijk is aan het aantal beschikbare zitplaatsen min de vraag. Ongelijkheid is van toepassing indien de vraag groter is dan het aantal zitplaatsen, waardoor er geen zitplaatsen leeg blijven.

Voorwaarde (2.4) en (2.5) geven aan dat het aantal beschikbare zitplaatsen en het aantal zitplaatsen dat leeg blijft nooit negatief is en dat het aantal lege zitplaatsen nooit groter is dan het aantal zitplaatsen dat beschikbaar is gesteld.

Voor de invoering van nesting is het echter makkelijker om te kijken naar het aantal verkochte tickets in plaats van het aantal zitplaatsen dat leeg blijft. Hierdoor is voor het geval met nesting in één oogopslag te zien wat de betekenis van de doelfunctie en de voorwaarden is. Dit in tegenstelling voor het wiskundig gelijke model dat verkregen zou worden als het aantal niet gebruikte zitplaatsen als recourse variabele gebruikt zou worden. Na de aanpassing wordt voor het geval zonder nesting het volgende equivalente model verkregen.

$$\max \quad E \sum_j f_j y_j \quad (3.1)$$

$$\text{s.t.} \quad y_j \leq x_j \quad \forall j \quad (3.2)$$

$$\sum_{j \in J(h)} x_j + s_l \leq c_h \quad \forall h \text{ (cabine met een hoge rang)} \quad (3.3a)$$

$$\sum_{j \in J(l)} x_j - s_l \leq c_l \quad \forall l \text{ (cabine met een lage rang)} \quad (3.3b)$$

$$0 \leq y_j \leq D_j \quad \forall j \quad (3.4)$$

$$x_j \geq 0 \quad \forall j \quad (3.5)$$

met  $y_j$ : aantal zitplaatsen die verkocht worden in ODF  $j$ . Dit is gelijk aan het minimum van de vraag  $D_j$  en het aantal beschikbare zitplaatsen  $x_j$ .

In bovenstaande model geeft doelfunctie (3.1) de verwachte totale opbrengst weer. Daarnaast geeft voorwaarde (3.2) de voorwaarde dat nooit meer tickets verkocht mogen worden dan dat er beschikbaar zijn. Voorwaarde (3.3) is de capaciteitsrestrictie van de cabines na upgrading en tenslotte geven voorwaarden (3.4) en (3.5) de onder- en bovengrenzen van de beslissingsvariabelen weer.

Om dit model te verkrijgen is gebruik gemaakt van het volgende gegeven:

$$y_j = x_j - \tilde{y}_j \Rightarrow \tilde{y}_j = x_j - y_j$$

Vervolgens wordt model 3 op de volgende manier verkregen.

Doelfunctie (3.1) wordt verkregen door doelfunctie (2.1) te herschrijven:

Voorwaarde (3.2) wordt verkregen door het eerste deel van voorwaarde (2.5) om te schrijven.

Voorwaarde (3.3) en (3.5) zijn dezelfde voorwaarden als (2.2) respectievelijk (2.4)

Het eerste deel van voorwaarde (3.4) wordt verkregen door het tweede deel van voorwaarde (2.5) om te schrijven.

Tenslotte kan het tweede deel van voorwaarde (3.4) worden verkregen door het omschrijven van voorwaarde (2.3).

Hieruit volgt dat model 3 daadwerkelijk equivalent is aan model 2.

### 3.2.3 Stochastisch model met volledige nesting

De enige aanpassing van het model met nesting ten opzichte van dat zonder nesting is dat beschikbare zitplaatsen die niet gebruikt worden, nu wel door een ODF met een hogere rang gebruikt mogen worden. Hierdoor moet voorwaarde (3.2) dusdanig aangepast worden dat het aantal verkochte tickets nooit boven het aantal beschikbare zitplaatsen plus het aantal zitplaatsen dat bij lagere klassen leeg blijven. De overige voorwaarden blijven ongewijzigd.

Het model ziet er dan als volgt uit.

$$\max \quad E \sum_j f_j y_j \quad (4.1)$$

$$\text{s.t.} \quad y_j \leq \sum_{\substack{k \in J(i) \\ k > j}} (x_k - y_k) + x_j \quad \forall j \quad \forall i \in I(j) \quad (4.2)$$

$$\sum_{j \in J(h)} x_j + s_h \leq c_h \quad \forall h \text{ (cabine met een hoge rang)} \quad (4.3a)$$

$$\sum_{j \in J(l)} x_j - s_l \leq c_l \quad \forall l \text{ (cabine met een lage rang)} \quad (4.3b)$$

$$0 \leq y_j \leq D_j \quad \forall j \quad (4.4)$$

$$x_j \geq 0 \quad \forall j \quad (4.5)$$

met  $I(j)$ : alle cabines die ODF  $j$  bevatten.

In vergelijking met het derde model is alleen voorwaarde (3.2) aangepast. De overige voorwaarden en de doelfunctie zijn gelijk gebleven.

### 3.2.4 Model met nesting per pad

Aangezien bij volledige nesting voor elke ODF naar elke cabine afzonderlijk gekeken moet worden, kan dit resulteren in een te groot aantal voorwaarden. Als dit het geval is, kan besloten worden om een tussenweg te zoeken tussen het model met volledige nesting en het model zonder nesting. Een oplossing kan zijn om een model te maken waarbij nesting alleen wordt toegepast tussen ODF's die vliegen op precies dezelfde vluchten en op die vluchten gebruikmaken van dezelfde cabines. Al deze ODF's maken dus op hetzelfde moment gebruik van hetzelfde pad.

Door in plaats van volledige nesting, nesting per pad toe te passen, is het niet meer nodig om voor elke afzonderlijke cabine te bekijken wat het resterend aantal beschikbare lege zitplaatsen is. Het volstaat nu om te kijken naar het aantal zitplaatsen dat de lagere ODF's op hetzelfde pad ongebruikt laten. Deze aanpassing kan gedaan worden door in voorwaarde (4.2) te eisen dat  $k$  over hetzelfde pad gaat als  $j$ . De voorwaarde komt er nu als volgt uit te zien:

$$y_j \leq \sum_{\substack{k > j \\ k \in P(j)}} (x_k - y_k) + x_j \quad \forall j$$

met  $P(j)$  = het pad van ODF  $j$ .

Het volledige model ziet er nu als volgt uit.

$$\max \quad E \sum_j f_j y_j \quad (5.1)$$

$$\text{s.t.} \quad y_j \leq \sum_{\substack{k>j \\ P(k)=P(j)}} (x_k - y_k) + x_j \quad \forall j \quad (5.2)$$

$$\sum_{j \in J(h)} x_j + s_l \leq c_h \quad \forall h \text{ (cabine met een hoge rang)} \quad (5.3a)$$

$$\sum_{j \in J(l)} x_j - s_l \leq c_l \quad \forall l \text{ (cabine met een lage rang)} \quad (5.3b)$$

$$0 \leq y_j \leq D_j \quad \forall j \quad (5.4)$$

$$x_j \geq 0 \quad \forall j \quad (5.5)$$

### 3.3 Deterministisch equivalent

Als de vraag slechts een eindig aantal waarden aan kan nemen, dan zijn de stochastische modellen om te schrijven naar equivalente deterministische modellen zonder stochastische variabelen. In Kall en Wallace (1994) wordt beschreven hoe deze omschrijving er uitziet voor een stochastisch model waarbij het enige stochastische element van het model de capaciteitskant van de voorwaarden (het rechterlid) is. Deze methode is ook beschreven in de algemene beschrijving van stochastisch programmeren in paragraaf 3.2.1.

In het geval van het hier beschreven revenue management probleem, is het stochastische element de vraag naar de verschillende ODF's. Als aangenomen wordt dat alle combinaties van realisaties voor de vraag in de verschillende ODF's mogelijk zijn, dan is het totaal aantal realisaties voor de vraag van alle ODF's gelijk aan het product van  $N_j$ , het aantal uitkomsten voor ODF  $j$  afzonderlijk, over alle ODF's. Aangezien bij elke uitkomst evenveel recourse variabelen horen als  $J$ , het aantal ODF's, komt het totaal aantal recourse variabelen op:  $J * \prod_{j=1}^J N_j$ . Om het deterministische

model hanteerbaar te houden, is het daarom nodig om een manier te vinden om het aantal recourse variabelen te beperken.

Hieronder zal voor zowel het geval zonder als de gevallen met nesting beschreven worden hoe het aantal uitkomsten beperkt is en wat de uiteindelijke deterministische equivalente modellen zijn.

#### 3.3.1 Model zonder nesting

In het model zonder nesting zijn de recourse variabelen uitsluitend afhankelijk van de uitkomst van de vraag van de bijbehorende ODF. Dit houdt in dat de recourse variabelen die horen bij ODF  $j$  voor alle realisaties met dezelfde vraag naar deze ODF, gelijk zijn. Hierdoor is het voldoende om te kijken naar de marginale kansverdeling van  $D_j$  en is de simultane kansverdeling van  $D_i$  en  $D_j$  niet nodig. Het aantal recourse variabelen is daardoor gelijk aan de som van het aantal mogelijke uitkomsten van de vraag van een ODF, over alle ODF's:  $\sum_j N_j$ . Het deterministische equivalent van model (3) ziet er nu als volgt uit.

$$\max \sum_j f_j \sum_{n=1}^{N_j} p_j^n y_j^n \quad (6.1)$$

$$\text{s.t.} \quad y_j^n \leq x_j \quad \forall j \quad n = 1, \dots, N_j \quad (6.2)$$

$$\sum_{j \in J(h)} x_j + s_h \leq c_h \quad \forall h \text{ (cabine met een hoge rang)} \quad (6.3a)$$

$$\sum_{j \in J(l)} x_j - s_l \leq c_l \quad \forall l \text{ (cabine met een lage rang)} \quad (6.3b)$$

$$0 \leq y_j^n \leq d_j^n \quad \forall j \quad n = 1, \dots, N_j \quad (6.4)$$

$$x_j \geq 0 \quad \forall j \quad (6.5)$$

met:  $N_j$ : aantal waarden die de vraag naar ODF  $j$  kan aannemen.

$p_j^n$ : kans dat de vraag naar ODF  $j$  de  $n^e$  waarde aanneemt.

$y_j^n$ : aantal zitplaatsen dat verkocht wordt in ODF  $j$ .

$d_j^n$ :  $n^e$  waarde die de vraag naar ODF  $j$  kan aannemen.

Dit model is in de vorm van een LP-model en kan op dezelfde manier als het deterministische model opgelost worden.

Om te zien of dit model niet veel te groot is in verhouding tot het oorspronkelijke deterministische model, zal hieronder het aantal voorwaarden en beslissingsvariabelen bepaald worden.

Het aantal voorwaarden in dit model is gelijk aan het aantal voorwaarden dat voortkomt uit voorwaarden (6.2) en (6.3) van het model. Voorwaarde (6.4) vormt een onder- en bovengrens voor de recourse variabelen en worden daarom niet meegeteld. Hetzelfde geldt voor voorwaarde (6.5), dat een bovengrens vormt voor gevraagde boekingslimieten.

Het aantal voorwaarden dat voortkomt uit voorwaarde (6.2) is gelijk aan de som van het aantal recourse variabelen in het model. Zoals eerder gemeld, is dat gelijk aan  $\sum_j N_j$

Voorwaarde (6.4) is voor elke cabine  $i$  de capaciteitsrestrictie. Het aantal voorwaarden dat hieruit voortkomt is dus gelijk aan het aantal cabines,  $I$ .

In totaal bevat dit model naast de onder- en bovengrenzen van de beslissingsvariabelen,  $I + \sum_j N_j$  voorwaarden waaraan de beslissingsvariabelen moeten voldoen.

De beslissingsvariabelen van dit model zijn de boekingslimieten  $x_j$  en de recourse variabelen  $y_j$ .

Het aantal beslissingsvariabelen is dus gelijk aan  $\sum_j (N_j + 1)$ .

Als voor alle ODF's geldt dat het aantal mogelijke uitkomsten gelijk is aan  $N$ , dan neemt het aantal beslissingsvariabelen in vergelijking met het deterministische model toe met factor  $N$ . Het aantal voorwaarden neemt toe met  $JN$ .

Als het aantal uitkomsten niet groot is, is dit model dus niet veel groter als het oorspronkelijke model en zal de rekentijd niet uit de hand lopen.



### 3.3.2 Model met volledige nesting

Voor het eerder genoemde stochastische model met volledige nesting, model 4, geldt dat de recourse variabelen afhankelijk zijn van de vraag van meerdere ODF's. Hierdoor is het niet mogelijk om het aantal recourse variabelen te beperken op de manier waarop dat is gebeurd bij het geval zonder nesting. Het is daarom nodig om het aantal uitkomsten van de vraag over alle ODF's op een andere manier te beperken.

Een methode is het aantal mogelijke uitkomsten per ODF kleiner te maken. Als alle combinaties van uitkomsten mogelijk zijn, dan is het aantal uitkomsten gelijk aan het product van de uitkomsten per ODF:  $\prod_j N_j$ . Als nu voor elke ODF het aantal uitkomsten gehalveerd wordt, komt het

$$\text{totaal aantal uitkomsten op: } \prod_j \binom{N_j}{2} = \left(\frac{1}{2}\right)^J \prod_j N_j$$

Een halvering van het aantal uitkomsten per ODF leidt dus tot een verkleining van het aantal recourse variabelen met factor  $2^J$ . Een nadeel is echter dat de nieuwe discrete benadering voor de vraag een minder goede benadering is voor de werkelijke verdeling van de vraag.

Een andere manier is om aan te nemen dat bepaalde combinaties van vraag niet voorkomen. Zo kan men aannemen dat als van een ODF de vraag relatief laag is, dat dan de vraag van andere ODF's ook relatief laag is. Het is echter de vraag of het wel reëel is om aan te nemen dat alle ODF's gelijktijdig een bepaald niveau behalen.

Een minder zware aanname is om de ODF's in groepen te verdelen waarvoor geldt dat als de vraag van een ODF in een groep relatief laag is, de rest van de ODF's in die groep ook een relatief lage vraag heeft, maar dat de niveaus van de groepen wel van elkaar mogen verschillen.

In de praktijk zijn er veel factoren die de vraag naar alle ODF's van een bepaalde groep tegelijk beïnvloeden. Te denken valt aan SARS en 11 september waardoor de vraag naar tickets naar bepaalde bestemmingen afnam. Het lijkt daarom reëel om deze aanname te maken.

Als deze methode wordt toegepast is het aantal uitkomsten gelijk aan het product van het aantal uitkomsten in een groep, over alle groepen:  $\prod_g N_g$ . Hierbij is het aantal uitkomsten binnen de groep,  $N_g$ , gelijk aan het aantal niveaus dat de vraag van die groep kan aannemen.

Het aantal recourse variabelen is dan gelijk aan het totaal aantal uitkomsten maal het totaal aantal ODF's:  $J \prod_g N_g$ .

Een nadeel van deze methode is dat door deze aanname niet langer elke combinatie van vraag mogelijk is. Bovendien zorgt deze methode ervoor dat de vraag naar de verschillende ODF's binnen een groep sterk afhankelijk van elkaar zijn.

Voor het geval met volledige nesting is gekozen voor de aanname dat voor elke ODF de vraag een bepaald aantal,  $N$ , niveaus kan aannemen en dat alle ODF's gelijktijdig een bepaald niveau bereiken. Voor het modellen met nesting per pad, wat in de volgende paragraaf wordt besproken, is een soortgelijke aanname gedaan. Voor de overige modellen was het niet noodzakelijk om aannames te maken over de afhankelijkheid van de vraag naar de verschillende ODF's. Door het doen van de aanname zijn er nog maar  $N$  uitkomsten mogelijk. Er is nu één groep die alle ODF's bevat en deze kan  $N_g = N$  niveaus bereiken. Het aantal recourse variabelen is dan gelijk aan

$J \prod_g N_g = JN$ . Deze keuze is gemaakt om het aantal recourse variabelen klein te houden zodat de rekentijden klein blijven. Het model dat op deze manier verkregen wordt, ziet er als volgt uit.

$$\max \sum_j f_j \sum_{n=1}^N p_j^n y_j^n \quad (7.1)$$

$$\text{s.t. } y_j^n \leq \sum_{\substack{k \in J(i) \\ k > j}} (x_k - y_k^n) + x_j \quad \forall j, \forall i \in I(j), n = 1, \dots, N \quad (7.2)$$

$$\sum_{j \in J(h)} x_j + s_h \leq c_h \quad \forall h \text{ (cabine met een hoge rang)} \quad (7.3a)$$

$$\sum_{j \in J(l)} x_j - s_l \leq c_l \quad \forall l \text{ (cabine met een lage rang)} \quad (7.3b)$$

$$0 \leq y_j^n \leq d_j^n \quad \forall j, n = 1, \dots, N \quad (7.4)$$

$$x_j \geq 0 \quad \forall j \quad (7.5)$$

Om te zien hoeveel groter is, worden ook hier weer het aantal voorwaarden en beslissingsvariabelen bepaald.

De voorwaarden voor dit model worden gevormd door voorwaarden (7.2) en (7.3). De andere drie voorwaarden vormen, net als in model 6, de onder- en bovengrenzen van de beslissingsvariabelen.

Voor elke waarde van  $j$  worden in voorwaarde (7.2)  $N$  voorwaarden voortgebracht voor elke  $i$ . In totaal is het aantal voorwaarden uit (7.2) dus gelijk aan  $N \sum_j I_j$ . Het aantal voorwaarden dat uit

voorwaarde (7.3) voortkomt is net als in model 6 gelijk aan  $I$ . Het totaal aantal voorwaarden komt dus uit op  $N \sum_j I_j + I$ .

Het aantal beslissingsvariabelen is gelijk aan  $J$ , het aantal  $x$ -variabelen, en  $JN$ , het aantal recourse variabelen.

In vergelijking met het geval zonder nesting waarbij elke ODF  $N$  mogelijke uitkomsten kan hebben, is het aantal beslissingsvariabelen dus gelijk aan  $JN+J$  gebleven en het aantal voorwaarden toegenomen met  $(\bar{I}-1)JN$  waarbij  $\bar{I}$  het gemiddelde aantal legs per ODF is. Aangezien  $J$  in de praktijk vaak erg groot is, kan dit erin resulteren dat het gehele LP-probleem te groot is om in een computermodel te zetten. Daarom kan het nodig zijn om te kiezen voor een model met nesting per pad.

### 3.3.3 Model met nesting per pad

Ook voor het stochastische model voor nesting per pad geldt dat de recourse variabelen afhankelijk zijn van meerdere ODF's. Het is dus ook hier niet mogelijk om het aantal recourse variabelen te beperken op de wijze waarop dat gedaan is voor het geval zonder nesting. Daarom wordt voor het model met nesting per pad een soortgelijke methode als bij het model met volledige nesting toegepast om het aantal beslissingsvariabelen te verkleinen. Hierbij is echter aangenomen dat de vraag naar ODF's die op hetzelfde pad reizen gelijktijdig een bepaald niveau aanneemt en dat de vraag naar ODF's op andere paden een ander niveau aan kan nemen. Voor elke ODF wordt het

aantal mogelijke uitkomsten dus beperkt tot  $N$  en kunnen de ODF's met hetzelfde pad niet gelijktijdig verschillende niveaus bereiken. Het model ziet er nu als volgt uit.

$$\max \sum_j f_j \sum_{n=1}^N p_j^n y_j^n \quad (8.1)$$

$$\text{s.t.} \quad y_j^n \leq \sum_{\substack{k>j \\ P(k)=P(j)}} (x_k - y_k^n) + x_j \quad \forall j \quad n=1, \dots, N \quad (8.2)$$

$$\sum_{j \in J(h)} x_j + s_h \leq c_h \quad \forall h \text{ (cabine met een hoge rang)} \quad (8.3a)$$

$$\sum_{j \in J(l)} x_j - s_l \leq c_l \quad \forall l \text{ (cabine met een lage rang)} \quad (8.3b)$$

$$0 \leq y_j^n \leq d_j^n \quad \forall j \quad n=1, \dots, N \quad (8.4)$$

$$x_j \geq 0 \quad \forall j \quad (8.5)$$

Ook voor dit model kan het aantal beslissingsvariabelen en voorwaarden bepaald worden. Het aantal voorwaarden is dit keer gelijk aan de som van het aantal voorwaarden per ODF over alle ODF's plus een capaciteitsvoorwaarde voor elke cabine. Voor een bepaalde waarde van  $j$  is het bijbehorende aantal voorwaarden gelijk aan het aantal niveaus. Het totaal aantal voorwaarden is nu gelijk aan  $JN + I$

Aangezien ten opzichte van het model met volledige nesting alleen de voorwaarden zijn veranderd, geldt dat het aantal beslissingsvariabelen gelijk is gebleven. Het model bevat dus  $JN$  beslissingsvariabelen.

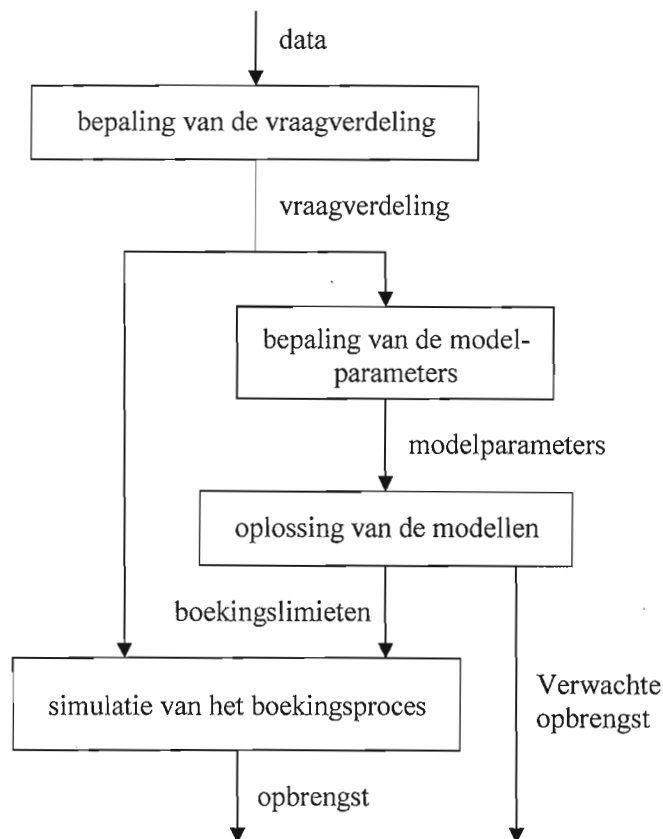
Ten opzichte van het model met volledige nesting is het aantal voorwaarden afgenomen met  $(\bar{I} - 1)N$ . Hierdoor is het mogelijk om problemen die niet in het model met volledige nesting gemodelleerd kunnen worden, toch te modelleren in een model met een beperkte vorm van nesting.

Ten opzichte van het model zonder nesting is het aantal voorwaarden en beslissingsvariabelen gelijk gebleven. Hierdoor kan voor een praktijkgeval waarvoor het model zonder nesting opgelost kan worden, ook het model met nesting per pad opgelost worden.

## 4. Simulatie en Resultaten

Aan de hand van een aantal simulaties is onderzocht hoe de prestaties van de verschillende oplossingsmethoden van elkaar verschillen. Hierbij is gekeken naar de resultaten van het deterministische model en de deterministische equivalenten van de gevallen zonder nesting en met nesting per pad. Het deterministische equivalent voor het geval met volledige nesting is buiten beschouwing gelaten, aangezien dit model te groot is om op te lossen.

In figuur 2 is te zien uit welke stappen het gehele proces bestaat. Daaronder is een beschrijving van de verschillende stappen te vinden.



**Figuur 2** het simulatieproces

### 1. *bepaling van de verdeling van de vraag*

Allereerst moet bepaald worden welk type verdeling de vraag heeft. Aangezien niet bekend is wat voor verdeling de vraag heeft, is het niet mogelijk om één verdeling te nemen die zeker representatief is voor het vraagproces. Daarom zijn een aantal verschillende verdelingen voor de vraag bekeken. De verdelingen die gekozen zijn, zijn als volgt onder te verdelen:

#### - *statische vraag*

In dit geval is de vraag voor een ODF geen dynamisch proces en komt de vraag in volgorde van rang, beginnend met klassen met een lage rang (hoog rangnummer). In dat geval is het dus niet nodig om met het aankomstproces rekening te houden en hoeft alleen geke-

ken te worden naar de totale vraag voor elke ODF. Voor dit geval worden twee simulaties gedaan.

- discrete verdeling
- normale verdeling
- *dynamische vraag*  
Nu is het vraagproces wel een dynamisch proces en komen de klanten van de verschillende ODF's wel door elkaar. Er moet dus wel gekeken worden naar de momenten waarop de klanten komen. Ook voor dit geval zijn verschillende simulaties uitgevoerd, waarbij de aankomstprocessen verschillen.
  - stationair Poisson-proces  
Hierbij is de aankomstintensiteit stationair en blijft dus door de tijd heen gelijk.
  - niet-stationair Poisson-proces  
Hierbij is de aankomstintensiteit niet stationair maar een functie van de tijd.

Als de keuze van de verdeling van de vraag bekend is, moeten de parameters van de verdeling nog bepaald worden. Dit is zodanig gedaan dat de verwachting van de verdeling gelijk is aan de voorspelling van de vraag die wel bekend is.

2. *bepaling van de parameters die in de modellen ingevuld moeten worden*

Aan de hand van de verdeling van de vraag naar tickets voor de verschillende ODF's moeten de parameters voor de modellen bepaald worden. Voor het deterministische model moet de verwachte vraag bepaald worden en voor de stochastische modellen een discrete benadering van de verdeling van de vraag. Voor de discrete benadering is een discrete verdeling met drie uitkomsten gebruikt. Dit is gedaan om te voorkomen dat het aantal voorwaarden groot wordt, waardoor het oplossen lang duurt. In hoofdstuk 5 is te zien in hoeverre de resultaten verbeteren als de discrete benadering nauwkeuriger wordt gemaakt door het verhogen van het aantal mogelijke uitkomsten.

3. *oplossing van de modellen*

De modellen moeten opgelost worden om de boekingslimieten te verkrijgen die bij de verschillende modellen horen. In de modellen worden de parameters ingevuld die verkregen zijn in stap 2. Voor elke vraagverdeling moeten de modellen dus opnieuw opgesteld en opgelost worden. De verkregen LP-modellen zijn opgelost met behulp van CPLEX. Door de oplossing van de modellen worden de verwachte opbrengst en de boekingslimieten verkregen.

4. *simulatie van het boekingsproces*

Simuleer het vraagproces, waarbij de in stap 3 verkregen boekingslimieten worden gebruikt om te bepalen welke klanten geaccepteerd worden.

De simulaties gaan uit van een netwerk met vluchten en ODF's dat representatief is voor de vluchten van een bestaande vliegtuigmaatschappij voor vier dagen. Bovendien is er bij de simulaties vanuit gegaan dat bij het toepassen van de boekingspolitiek altijd sprake is van nesting per pad. Er wordt dus niet op gelet of het model waarmee de boekingslimieten verkregen zijn, hier al dan niet rekening mee houdt.

Om de verschillende modellen met elkaar te kunnen vergelijken is het nodig om één of meerdere graadmeters te hebben. De graadmeter waarnaar gekeken wordt is de totale opbrengst over alle vluchten. Het doel van revenue management is immers om deze opbrengst te maximaliseren. Aan de hand van de resultaten uit de simulaties is voor elk model een 99 %-betrouwbaarheidsinterval

van de totale opbrengst bepaald. Dit betrouwbaarheidsinterval geeft weer tussen welke waarden de totale opbrengst met 99 % zekerheid zit.

Naast de betrouwbaarheidsintervallen voor de opbrengsten van de afzonderlijke modellen is ook voor elk paar modellen een betrouwbaarheidsinterval gegeven voor het verschil tussen de opbrengst van de betreffende modellen.

Tijdens het modelleringsproces zijn een aantal aannames gemaakt om het probleem te kunnen modelleren en de verkregen modellen werkbaar te maken. Een aantal van deze aannames zijn echter niet geheel overeenkomstig met de werkelijkheid. Daarom zijn deze tijdens de simulatie weggelaten, zodat de verkregen resultaten representatief zijn voor de werkelijkheid. Hierdoor zullen de verwachte opbrengsten niet overeenkomen met de behaalde resultaten uit de simulaties.

- Voor het deterministische model is de belangrijkste aanname die gemaakt is, de aanname dat de vraag vaststaat en bekend is. Doordat dit bij de simulaties niet het geval is, zal de verwachte opbrengst afwijken van de gemiddelde opbrengst uit de simulaties (het midden van het betrouwbaarheidsinterval).
- Daarnaast wordt bij de simulatie altijd nesting toegepast. Doordat het model zonder nesting er vanuit gaat dat dit niet gebeurt, zal ook voor dit model de verwachte opbrengst afwijken van de gemiddelde opbrengst uit de simulaties.
- Bij het model met nesting per pad wordt aangenomen dat de vraag naar de verschillende ODF's strikt afhankelijk is. Deze aanname wordt losgelaten tijdens de simulatie, waardoor ook hier een verschil op zal treden tussen de verwachte opbrengst en de gemiddelde opbrengst uit de simulaties.
- Tenslotte is bij alle modellen aangenomen dat de boekingsperioden van de verschillende ODF's strikt gescheiden zijn en dat de ODF's boeken op volgorde van rang. In de praktijk is dit wel vaak het geval, maar bij de simulaties met een dynamische vraag is dit niet meer het geval. Hierdoor zal een afwijking ontstaan tussen de verwachte opbrengst en de opbrengst uit de simulaties.

Het vervolg van dit hoofdstuk bevat ten eerste een korte uitleg over het opstellen van betrouwbaarheidsintervallen. Daarna is beschreven hoe de simulaties zijn uitgevoerd en wat de resultaten zijn van de verschillende simulaties. Hierbij is een opsplitsing gemaakt tussen simulaties met een statische vraag en simulaties met een dynamische vraag.

#### 4.1 Betrouwbaarheidsintervallen

In deze paragraaf zal een korte uitleg gegeven worden over betrouwbaarheidsintervallen en het opstellen ervan. Voor een uitgebreidere uitleg wordt verwezen naar Law en Kelton (2000).

Betrouwbaarheidsintervallen komen aan de orde bij het kijken naar de onbekende verwachting van een stochastische variabele  $Z$  aan de hand van een steekproef. Een  $\beta$ %-betrouwbaarheidsinterval is dat interval waarvoor geldt dat het  $\beta$  % zeker is dat de verwachting van de stochastische variabele in het interval valt. Voor het bepalen van een betrouwbaarheidsinterval zijn twee gegevens nodig. Allereerst moet bekend zijn wat het steekproefgemiddelde is. Daarnaast moet de steekproefvariantie bekend zijn.

Het steekproefgemiddelde is een schatter voor de onbekende verwachtingswaarde van de stochastische variabelen.

$$\bar{Z}(n) = \frac{1}{n} \sum_{i=1}^n Z_i$$

met:  $Z_i$  de  $i$ -de waarneming van de variabele en  $n$  het aantal waarnemingen.

De steekproefvariantie is een schatter voor de variantie van  $Z$ . De formule voor deze schatter bij  $n$  waarnemingen is als volgt.

$$S^2(n) = \frac{\sum_{i=1}^n (Z_i - \bar{Z}(n))^2}{(n-1)}$$

Wanneer zowel het steekproefgemiddelde als de steekproefvariantie berekend zijn, kan, op basis van de wet van de grote getallen, het betrouwbaarheidsinterval als volgt bepaald worden.

$$\left[ \bar{Z}(n) - t_{n-1, 1-\alpha/2} S(n) / \sqrt{n}, \bar{Z}(n) + t_{n-1, 1-\alpha/2} S(n) / \sqrt{n} \right] \quad \text{met } \alpha = 1 - \frac{\beta}{100}$$

Hierin is  $t_{n-1, 1-\alpha/2}$  de kritieke waarde voor de studentverdeling (t-verdeling) met  $n-1$  vrijheidsgraden, dus die waarde van  $x$  met  $P(X \leq x) \geq 1 - \frac{\alpha}{2}$  als  $X$  een t-verdeelde variabele is met  $n-1$  vrijheidsgraden.

Zoals uit de formule voor het betrouwbaarheidsinterval te zien is, zijn de steekproefvariantie, de kritieke waarde van de studentverdeling en het aantal steekproeven bepalend voor de grootte van het interval. Aangezien uit een groot interval weinig valt op te maken over de werkelijke verwachtingswaarde van de stochastische variabele, is het zinvol om te kijken naar methoden die het interval verkleinen.

De eenvoudigste methode is om simpelweg het aantal simulaties te vergroten. Een vergroting van het aantal simulaties heeft nauwelijks invloed op de kritieke waarden en de waarde van de steekproefvariantie. Het positieve effect van de verhoging van het aantal simulaties wordt dus niet tenietgedaan, waardoor het daadwerkelijk een gunstig effect heeft op de grootte van het interval. Een andere methode is om de steekproefvariantie te verkleinen. Dit kan door middel van variantie-reductie technieken. Een van die technieken is de zogenaamde Common Random Numbers (CRN) methode. Dit houdt in dat voor de te vergelijken modellen tijdens de simulatie dezelfde trekkingen uit de uniforme verdeling tussen 0 en 1 gebruikt worden. Het idee hierachter is dat hierdoor de variantie van  $\bar{Z}(n)$  lager wordt. Voor het verschil tussen opbrengsten  $O^1$  en  $O^2$  is die variantie als volgt.

$$\text{var}(\bar{Z}(n)) = \text{var}\left(\frac{1}{n} \sum Z_i\right) = \frac{\text{var}(Z_i)}{n} = \frac{\text{var}(O_i^1) + \text{var}(O_i^2) - 2 \text{cov}(O_i^1, O_i^2)}{n}$$

Als nu verschillende onafhankelijke trekkingen worden gebruikt, dan zijn  $O_i^1$  en  $O_i^2$  onafhankelijk, zodat de covariantie gelijk is aan 0. Als echter de covariantie groter dan 0 gemaakt kan worden, dan wordt de variantie kleiner. In Kall & Wallace is kort beschreven wanneer het zin heeft om de CRN methode te gebruiken om de variantie te reduceren. Bovendien zijn daarin een aantal referenties te vinden waarin dit onderwerp nader besproken wordt. Kort gezegd komt het erop neer dat het zinvol is om voor de simulaties van twee te vergelijken modellen de CRN me-

thode toe te passen als de resultaten op dezelfde wijze reageren (beide toenemen of beide afnemen) op een toename van de trekking. Als dit niet het geval is, dan zal de covariantie van de resultaten negatief zijn en heeft het toepassen van CRN een negatieve invloed op de variantie (deze wordt groter).

Voor alle modellen geldt dat een verhoging van de trekking dezelfde invloed heeft op de vraag. Deze vraag gaat hetzij omhoog doordat de trekking direct de vraag weergeeft, hetzij omlaag doordat de trekking de tussenaankomsttijd weergeeft.

Daarnaast geldt voor alle modellen dat een verhoging (verlaging) van de vraag een positieve (negatieve) invloed heeft op de opbrengst. De modellen reageren dus op dezelfde manier op het verhogen van de trekking. Daarom is het zinvol om de CRN methode toe te passen bij het vergelijken van de modellen en zijn voor de simulaties van alle modellen dezelfde trekkingen uit de uniforme verdeling gebruikt.

## 4.2 Simulaties met een statische vraag

In deze paragraaf wordt besproken hoe de simulaties gedaan zijn in het geval van een statische vraag. Hieronder zal eerst een algemene beschrijving worden gegeven van de wijze waarop de vierde stap uit de inleiding van dit hoofdstuk gedaan is. Deze stap is de feitelijke simulatie van het boekingsproces. Hierbij wordt aangenomen dat de ODF's met een lage rang eerst boeken en daarna pas die met een hogere rang. Vervolgens bespreken de twee volgende subparagrafen de eerste twee stappen voor de verschillende verdelingen van de vraag. Daarnaast is te vinden op welke manier de vraag van de betreffende verdeling gesimuleerd is. Tenslotte worden de resultaten van de simulaties in deze subparagrafen besproken.

De simulatie begint met het vaststellen van de cumulatieve boekingslimieten. Dit is het minimaal aantal zitplaatsen dat bezet moet zijn, voor een nieuw verzoek voor ODF  $j$  niet meer geaccepteerd wordt. De formule voor de cumulatieve boekingslimiet van ODF  $j$  is als volgt.

$$b_j = \sum_{s \geq j} x_s$$

Vervolgens moet voor de laagste rang een vraag gesimuleerd worden. Aan de hand van deze vraag kan bepaald worden hoeveel daarvan wordt geaccepteerd. Het aantal geaccepteerde klanten is gelijk aan het minimum van de cumulatieve boekingslimiet en de vraag.

Deze stap wordt gevolgd door de aanpassing van de cumulatieve boekingslimieten van de ODF's op hetzelfde pad met een hogere rang. Aangezien de geaccepteerde klanten zitplaatsen bezetten, zijn deze zitplaatsen niet meer beschikbaar voor hogere klassen en moeten de cumulatieve boekingslimieten hiervan verlaagd worden met het aantal geaccepteerde klanten.

Vervolgens wordt een vraag gesimuleerd voor de ODF met de volgende rang. Ook van deze vraag kan weer bepaald worden hoeveel wordt geaccepteerd.

Hierna worden de cumulatieve boekingslimieten van de ODF's op hetzelfde pad en met een hogere rang weer verlaagd met het aantal geaccepteerde klanten.

Dit proces herhaalt zich tot voor elke ODF een vraag is gesimuleerd.

Het algoritme dat doorlopen wordt, is dus als volgt.

### 0. initialisatie:

- bepaal de cumulatieve boekingslimieten,  $b_j$ .
- stel totale opbrengst = 0
- stel  $j$  gelijk aan  $J$ , het hoogste rangnummer van alle ODF's



1. simulatie van de vraag:

- simuleer een vraag voor ODF  $j$  (dit is de ODF met rangnr  $j$ )
- bepaal het aantal passagiers dat geaccepteerd wordt
- verhoog de totale opbrengst met het aantal geaccepteerde passagiers maal  $f_j$
- verminder de cumulatieve boekingslimieten voor de ODF's op hetzelfde pad als  $j$  en met een hogere rang met het aantal geaccepteerde passagiers
- verlaag  $j$  met 1
- als  $j$  niet negatief is, herhaal dan stap 1, anders ga verder met stap 2

2. einde simulatie

- sla de opbrengst van deze simulatie op
- verhoog de totale opbrengst over alle simulaties
- hoog het simulatienummer met 1 op
- als het simulatienummer niet groter is dan het gewenste aantal simulaties (voor de statische vraag is gekozen voor 201), ga verder met stap 0, anders verder met stap 3

3. einde alle simulaties

- bepaal het 99 %-betrouwbaarheidsinterval van de opbrengst

Zoals al eerder vermeld, is deze simulatie uitgevoerd voor twee typen verdelingen voor de vraag. In subparagraaf 4.2.1 wordt de beschrijving gegeven voor de simulatie waarbij de vraag discreet verdeeld is en in subparagraaf 4.2.2 die voor een normaal verdeelde vraag.

**4.2.1 Discreet verdeelde vraag met drie uitkomsten**

In deze subparagraaf is te zien hoe de verschillende stappen, zoals die beschreven zijn op pagina 27, gedaan zijn voor het geval dat de vraag discreet verdeeld is. Tenslotte volgen de resultaten van de simulaties.

1. bepaling van de verdeling van de vraag

Bij de simulaties is uitgegaan van een database waarin de voorspelling voor de vraag,  $ED_j$ , naar elke ODF bekend is, maar de verdeling niet. Daarom is ervoor gekozen om een aantal verschillende simulaties uit te voeren waarbij de verwachtingswaarde gelijk blijft, maar de rest van de verdeling varieert. Bij deze verdelingen is het aantal uitkomsten voor alle ODF's vastgesteld op hetzelfde aantal (drie) als de discrete benadering die in de deterministische modellen ingevoerd moeten worden. Voor het aantal uitkomsten is drie gekozen aangezien een groter aantal leidt tot een groter model. Dit zorgt er op zijn beurt weer voor dat het langer duurt om het model op te lossen, wat onwenselijk is als de rekentijd al groot is. Door het aantal op drie te stellen bleef de rekentijd werkbaar. In paragraaf 5.1 is te zien wat er met de rekentijd en simulatieresultaten gebeurt als een discrete verdeling met negen uitkomsten wordt gekozen.

In tabel 1 is te zien welke verdelingen gekozen zijn.

simulatie	Variatie-coëfficiënt	$d_j^1$	$p_j^1$	$d_j^2$	$p_j^2$	$d_j^3$	$p_j^3$
1	0.20	$0.75 \bar{E}D_j$	0.33	$\bar{E}D_j$	0.34	$1.25 \bar{E}D_j$	0.33
2	0.35	$0.50 \bar{E}D_j$	0.25	$\bar{E}D_j$	0.50	$1.50 \bar{E}D_j$	0.25
3	0.41	$0.50 \bar{E}D_j$	0.33	$\bar{E}D_j$	0.34	$1.50 \bar{E}D_j$	0.33
4	<del>0.45 <math>\bar{E}D_j</math></del>	$0.50 \bar{E}D_j$	0.40	$\bar{E}D_j$	0.20	$1.50 \bar{E}D_j$	0.40
5	<del>0.61 <math>\bar{E}D_j</math></del>	$0.25 \bar{E}D_j$	0.33	$\bar{E}D_j$	0.34	$1.75 \bar{E}D_j$	0.33

Tabel 1 gebruikte discrete verdelingen met bijbehorende variatiecoëfficiënt

De verdeling van simulatie 3 is als uitgangspunt genomen. Vervolgens is de variantie in simulatie 1 en 2 verlaagd door hetzij de mogelijke uitkomsten aan te passen (1), hetzij de kansen aan te passen (2). In de overige twee simulaties is de variantie verhoogd. Wederom door het aanpassen van de kansen (4) en de mogelijke uitkomsten (5).

2. bepaling van de parameters die in de modellen ingevuld moeten worden

Voor het deterministische model moet voor elke ODF bepaald worden wat de verwachting van de vraag is. In het geval van een discrete verdeling is de formule voor de verwachtingswaarde als volgt.

$$\mu_j = \sum_{k=1}^K p_j^k d_j^k \quad \text{met } K \text{ het aantal uitkomsten}$$

Het is echter niet nodig om deze formule te gebruiken, aangezien de keuze van de verdeling dusdanig is dat de verwachting van de verdeling gelijk is aan de voorspelling van de vraag. Voor de deterministische equivalenten van de stochastische modellen moet de discrete verdeling met drie mogelijke uitkomsten bepaald worden. Aangezien de vraag al discreet verdeeld is, is deze gelijk aan de vraagverdeling.

3. oplossing van de modellen

Nu bekend is welke parameters in de modellen ingevuld moeten worden, kunnen de modellen opgelost worden. Dit is gedaan met behulp van het programma CPLEX.

4. simulatie van het vraagproces

Voor de simulatie is het nodig om trekkingen te verrichten uit de discreet verdeelde vraag. Om een trekking te doen uit een discrete verdeling voor vraag  $D_j$ , moeten de volgende stappen gemaakt worden.

1. trek een getal  $U$  uit  $U(0,1)$  (de uniforme verdeling tussen 0 en 1)
2. bepaal de waarde  $I$  waarvoor geldt:  $\sum_{i=1}^{I-1} p_j^i \leq U < \sum_{i=1}^I p_j^i$
3. geef de bijbehorende vraag,  $d_j^I$ , terug

Resultaten

In onderstaande tabellen zijn de resultaten weergegeven die met deze simulatie en de oplossing van de modellen zijn verkregen. In tabel 2 zijn per model en simulatie een waarde en een betrouwbaarheidsinterval weergegeven. De vet en cursief gedrukte waarde geeft de verwachte opbrengst van de boekingspolitiek weer. Het betrouwbaarheidsinterval geeft het interval weer waarin de opbrengst met 99 % zekerheid ligt.

In tabel 3 is per simulatie weergegeven wat de betrouwbaarheidsintervallen zijn voor de verschillen tussen de drie modellen.

	1	deterministische model	2	stochastische model zonder nesting	3	stochastische model met nesting per pad
1		<b><i>1.613.330</i></b> [ 1.514.378 , 1.515.393 ]		<b><i>1.613.009</i></b> [ 1.612.272 , 1.613.974 ]		<b><i>1.613.141</i></b> [ 1.612.124 , 1.613.828 ]
2		<b><i>1.613.330</i></b> [ 1.455.422 , 1.457.201 ]		<b><i>1.612.379</i></b> [ 1.611.076 , 1.614.054 ]		<b><i>1.612.557</i></b> [ 1.610.977 , 1.613.953 ]
3		<b><i>1.613.330</i></b> [ 1.415.285 , 1.417.318 ]		<b><i>1.612.111</i></b> [ 1.610.735 , 1.614.156 ]		<b><i>1.612.310</i></b> [ 1.610.637 , 1.614.059 ]
4		<b><i>1.613.330</i></b> [ 1.382.697 , 1.384.921 ]		<b><i>1.611.890</i></b> [ 1.610.638 , 1.614.362 ]		<b><i>1.612.094</i></b> [ 1.610.593 , 1.614.317 ]
5		<b><i>1.613.330</i></b> [ 1.316.135 , 1.319.186 ]		<b><i>1.610.821</i></b> [ 1.608.923 , 1.614.057 ]		<b><i>1.611.121</i></b> [ 1.609.043 , 1.614.178 ]

**Tabel 2** verwachte opbrengst uit de modellen (vetgedrukt) en 99 %-betrouwbaarheidsintervallen voor de verwachte werkelijke opbrengst uit de simulaties bij verschillende discrete vraagverdelingen

	model 2 – model 1	model 3 – model 1	model 3 – model 2
1	[ 97.750 , 98.725 ]	[ 97.602 , 98.579 ]	[ -167 , -127 ]
2	[ 155.329 , 157.177 ]	[ 155.229 , 157.078 ]	[ -120 , -80 ]
3	[ 195.166 , 197.122 ]	[ 195.067 , 197.026 ]	[ -123 , -72 ]
4	[ 227.654 , 229.728 ]	[ 227.608 , 229.684 ]	[ -73 , -17 ]
5	[ 292.361 , 295.298 ]	[ 292.481 , 295.419 ]	[ 81 , 160 ]

**Tabel 3** betrouwbaarheidsintervallen voor de verschillen tussen de drie modellen bij verschillende discrete vraagverdelingen

Uit de resultaten blijkt, zoals was te verwachten, dat ondanks dat de verwachte opbrengst van het deterministische model het hoogst is, dit model in de simulaties juist het minst goed presteert. Voor het model zonder nesting geldt dat de verwachtingswaarde van de opbrengst iets onder die van het model met nesting per pad komt, terwijl in de meeste simulaties juist het model zonder nesting net iets beter presteert. Aangezien het verschil tussen de resultaten van de stochastische modellen klein is en niet één van beide in alle simulaties beter is, zou op grond van de simulaties geen van beide boven de ander geprefereerd worden.

Bovendien is uit tabel 3 te zien dat het verschil tussen de stochastische modellen en het deterministische model kleiner wordt naarmate de variatiecoëfficiënt kleiner wordt. Dit is te verklaren

doordat een kleine variatiecoëfficiënt duidt op een kleine onzekerheid. Hierdoor zal de fout die gemaakt wordt door de deterministische benadering minder groot zijn en ligt de verkregen oplossing minder ver van de optimale oplossing af. Als de variatiecoëfficiënt klein is, is het dus minder lonend om het model stochastisch te maken.

Daarnaast is te zien dat het verschil tussen het model met nesting en dat zonder nesting kleiner en zelfs negatief wordt als de variatiecoëfficiënt afneemt. Als de variatiecoëfficiënt klein is heeft het rekening houden met nesting dus een negatief effect op de opbrengst.

#### 4.2.2 Normaal verdeelde vraag

Ook in deze subparagraaf worden de vier stappen van het simulatieproces besproken. Dit keer is echter uitgegaan van een vraag die normaal-verdeeld is.

##### 1. *bepaling van de verdeling van de vraag*

Voor het bepalen van de normale verdeling van de vraag is het nodig om een verwachtingswaarde  $\mu$  en een standaardafwijking  $\sigma^2$  te weten. De verwachtingswaarde wordt gelijk gekozen aan de voorspelling van de vraag  $ED_j$ . Aangezien de gebruikte database niet beschikt over gegevens van de standaardafwijking van de vraag, zijn hiervoor een aantal waarden gekozen bij de simulatie. In plaats van de standaardafwijking, is echter de variatiecoëfficiënt  $\frac{\sigma}{\mu}$  gekozen. Per simulatie is de variatiecoëfficiënt gelijk gekozen voor alle ODF's. De waarden die gekozen zijn, zijn 0.2, 0.5 en 0.7.

##### 2. *bepaling van de parameters die in de modellen ingevuld moeten worden*

Indien een variabele  $N(\mu, \sigma^2)$ -verdeeld is, dan is de verwachtingswaarde bekend en gelijk aan  $\mu$ . Het is dus eenvoudig om te bepalen welke waarde ingevuld moet worden in het deterministische model.

De discrete benadering zou verkregen kunnen worden op de wijze zoals beschreven in Kall en Wallace (1994) en in paragraaf 3.2.1 van deze scriptie. Dan zou echter, om de benadering te bepalen, voor elke verdeling een groot aantal trekkingen uit de normale verdeling gedaan moeten worden. Daarom is gekozen voor een methode waarbij de kansen en verwachtingswaarden van de subintervallen analytisch bepaald worden. Dit in tegenstelling tot de methode van Kall en Wallace, waarbij de kansen bepaald worden met behulp van trekkingen uit de verdeling. Stapsgewijs is deze methode als volgt.

##### 1. *Bepaling van het 99 %-interval van de normale $N(\mu, \sigma^2)$*

Om het betrouwbaarheidsinterval te vinden, wordt eerst  $X$  omgeschreven als een functie van een andere stochast die standaardnormaal verdeeld is. Als  $X \sim N(\mu, \sigma^2)$ , dan geldt  $X = \mu + \sigma Y$ , met  $Y \sim N(0,1)$ . Er geldt dan dus

$$P(Y < \alpha) = P\left(\frac{X - \mu}{\sigma} < \alpha\right) = P(X - \mu < \alpha\sigma) = P(X < \alpha\sigma + \mu)$$

Het 99 %-betrouwbaarheidsinterval van de standaardnormale verdeling is te vinden in tabellen en gelijk aan  $[-\alpha, \alpha]$ , met  $\alpha = 2,5758$ . Het betrouwbaarheidsinterval van  $X$  is dan dus gelijk aan  $[\mu - \alpha\sigma, \mu + \alpha\sigma]$ .

2. *De opsplitsing van het betrouwbaarheidsinterval.*

Het betrouwbaarheidsinterval wordt opgesplitst in drie (het aantal uitkomsten van de discrete benadering) gelijke intervallen. Deze intervallen zijn dus gelijk aan

$$\left[ \mu - \alpha\sigma, \mu - \frac{1}{3}\alpha\sigma \right], \left[ \mu - \frac{1}{3}\alpha\sigma, \mu + \frac{1}{3}\alpha\sigma \right] \text{ en } \left[ \mu + \frac{1}{3}\alpha\sigma, \mu + \alpha\sigma \right]$$

3. *Bepaling van de verwachte waarde binnen de subintervallen.*

Voor een interval  $[a, b]$  wordt de verwachtingswaarde gegeven door de volgende formule.

$$E[x | a \leq x \leq b] = \frac{\int_a^b xf(x)dx}{\int_a^b f(x)dx}$$

met  $f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ , de kansdichtheid van  $X \sim N(\mu, \sigma^2)$ .

Voor de drie subintervallen worden zo de volgende verwachtingswaarden verkregen:  $\mu - 1.37\sigma$ ,  $\mu$  en  $\mu + 1.37\sigma$

4. *Bepaling van de kansen op een waarde binnen de subintervallen*

Voor interval  $[a, b]$  geldt dat de kans op een waarde binnen het interval gegeven wordt door de volgende formule.

$$P(a \leq x \leq b | \mu - \alpha\sigma \leq x \leq \mu + \alpha\sigma) = \frac{\int_{\mu+\alpha\sigma}^b f(x)dx}{\int_{\mu-\alpha\sigma}^{\mu+\alpha\sigma} f(x)dx}$$

Voor de drie subintervallen worden zo de volgende kansen verkregen: 0.19, 0.62 en 0.19

De onderstaande tabel geeft weer welke variatiecoëfficiënten gebruikt zijn en wat de bijbehorende discrete verdelingen zijn.

simulatie	$\frac{\sigma}{\mu}$	$d_j^1$	$p_j^1$	$d_j^2$	$p_j^2$	$d_j^3$	$p_j^3$
1	0.2	$0.73 ED_j$	0.19	$ED_j$	0.62	$1.27 ED_j$	0.19
2	0.5	$0.32 ED_j$	0.19	$ED_j$	0.62	$1.68 ED_j$	0.19
3	0.7	$0.04 ED_j$	0.19	$ED_j$	0.62	$1.96 ED_j$	0.19

**Tabel 4** de gebruikte variatiecoëfficiënten en de bijbehorende discrete verdeling

3. oplossing van het model

Ook voor dit geval worden de parameters in de modellen ingevoerd, waardoor drie LP-modellen worden verkregen. Deze modellen zijn weer opgelost met behulp van CPLEX.

4. simulatie van het vraagproces

Voor de simulatie van de vraag moeten trekkingen uit de normale verdeling gedaan worden. Dit is gedaan volgens de methode zoals beschreven in Law en Kelton (2000). Volgens deze methode bestaan de trekkingen bestaan uit de volgende stappen:

- I. trek de getallen  $U_1$  en  $U_2$  uit  $U(0,1)$
- II. neem  $V_i = -\log(U_i)$   $i = 1,2$  en  $W = \frac{1}{2}(V_1 - 1)^2$
- III. als  $V_2 < W$ , ga terug naar stap I  
anders trek  $U_3$  uit  $U(0,1)$
- IV. als  $U_3 \leq 0.5$ , geef  $\mu - \sigma V_1$  terug,  
anders geef  $\mu + \sigma V_1$  terug

Resultaten

In onderstaande tabel zijn de resultaten gegeven van de simulaties waarbij de vraag normaal verdeeld is verondersteld en de oplossing van de modellen met de bijbehorende parameters. Wederom zijn de vet en cursief gedrukte waarden de verwachte opbrengsten uit de modellen en de intervallen de 99 %-betrouwbaarheidsintervallen van de opbrengsten uit de simulatie. In tabel 6 zijn de 99 %-betrouwbaarheidsintervallen gegeven van de verschillen tussen de opbrengsten van de drie modellen.

	1	deterministische model	2	stochastische model zonder nesting	3	stochastische model met nesting per pad
1		<b><i>1.613.330</i></b> [ 1.519.308 , 1.520.434 ]		<b><i>1.613.087</i></b> [ 1.607.531 , 1.609.345 ]		<b><i>1.613.195</i></b> [ 1.608.215 , 1.610.037 ]
2		<b><i>1.613.330</i></b> [ 1.411.315 , 1.413.851 ]		<b><i>1.612.097</i></b> [ 1.642.686 , 1.646.987 ]		<b><i>1.609.253</i></b> [ 1.644.419 , 1.648.748 ]
3		<b><i>1.613.330</i></b> [ 1.405.935 , 1.408.631 ]		<b><i>1.610.933</i></b> [ 1.768.403 , 1.773.542 ]		<b><i>1.611.213</i></b> [ 1.771.682 , 1.776.844 ]

**Tabel 5** verwachte opbrengst en betrouwbaarheidsintervallen voor de verwachte werkelijke opbrengst bij normaalverdelingen met verschillende variatiecoëfficiënten

	model 2 – model 1	model 3 – model 1	model 3 – model 2
1	[ 88.073 , 89.061 ]	[ 88.756 , 89.754 ]	[ 643 , 732 ]
2	[ 231.046 , 233.461 ]	[ 232.776 , 235.226 ]	[ 1.616 , 1.878 ]
3	[ 361.987 , 365.392 ]	[ 365.264 , 368.696 ]	[ 3.127 , 3.455 ]

**Tabel 6** betrouwbaarheidsintervallen voor de verschillen tussen de drie modellen bij normale verdelingen met verschillende variatiecoëfficiënten

Uit de tabel 5 valt te zien dat de resultaten van het stochastische model met nesting het beste zijn en die van het deterministische model het minst goed. Daarnaast is uit tabel 6 net als bij de simulaties met een discreet verdeelde vraag te zien dat het verschil tussen de stochastische modellen en het deterministische model kleiner wordt naarmate de variatiecoëfficiënt afneemt. En dat toevoeging van nesting dan minder nut heeft.

Dat de verwachting van de opbrengst van de stochastische modellen niet bevat is in de betrouwbaarheidsintervallen van de opbrengst, wordt vermoedelijk veroorzaakt door de onnauwkeurigheid van de discrete benadering. De benadering gaat namelijk uit van een normale verdeling die niet beperkt is tot de niet-negatieve waarden. De simulatie gaat echter uit van een niet-negatieve vraag en laat negatieve trekkingen buiten beschouwing. Hierdoor is de gebruikte verdeling niet meer exact gelijk aan de normale verdeling en is de verwachte vraag hoger als bij de echte verdeling. Hierdoor zal de opbrengst ook hoger zijn. Vooral als de variatiecoëfficiënt groot is, speelt dit een rol. Dan is de kans op een negatieve waarde immers groter, waardoor de discrete benadering minder goed is. In hoofdstuk 5 zal gekeken worden naar het effect als het aantal uitkomsten van de discrete benadering verhoogd wordt. Naast een vergelijking van de opbrengst wordt ook gekeken of het verschil tussen de verwachte opbrengst en de verwachting voor de daadwerkelijke opbrengst kleiner wordt.

### 4.3 Simulaties met een dynamische vraag

Net als in de vorige paragraaf, zal hier eerst een beschrijving worden gegeven van de algemene methode waarop het boekingsproces is gesimuleerd. Dit keer is echter uitgegaan van het geval waarbij het vraagproces dynamisch is. Vervolgens staat in de subparagrafen een beschrijving van stappen 1, 2, 3 en 4 voor de specifieke aankomstprocessen van de vraag. Daarnaast zijn de resultaten van de simulaties in deze subparagrafen te vinden. In paragraaf 4.3.1 is de beschrijving terug te vinden van het geval waarbij de aankomstprocessen stationair zijn en in paragraaf 4.3.2 voor het geval waarbij de aankomstintensiteit wel afhangt van de tijd.

Als de vraag dynamisch is en de vraag naar de verschillende ODF's in plaats van na elkaar, door elkaar komt, dan is het nodig om te kijken naar de momenten waarop klanten een ticket voor een bepaalde ODF vragen.

De simulatie geschiedt dan niet langer door het genereren van een waarde die de totale vraag naar een ODF representeert, maar door het genereren van momenten waarop een klant van een ODF arriveert. De simulatie gaat nu van aankomst naar aankomst tot het moment waarop het vliegtuig vertrekt. Op het moment van een aankomst wordt aan de hand van de cumulatieve boekingslimieten bepaald of de passagier al dan niet geaccepteerd moet worden. Vervolgens moet bepaald worden wanneer de volgende aankomst van een klant uit dezelfde ODF is.

Op deze manier hoeft tijdens de simulatie per ODF van slechts één persoon onthouden te worden op welk tijdstip deze arriveert. Op het moment dat de simulatietijd deze tijd bereikt is, wordt de aankomst van deze persoon verwerkt en een aankomst voor de volgende gepland.

De stappen die tijdens de simulatie verricht moeten worden, zijn als volgt.

#### 0. initialisatie:

- stel  $t = 0$
- bepaal voor elke ODF de cumulatieve boekingslimieten voor het betreffende model
- plan het vertrek van de vliegtuigen
- plan de aankomst van de eerste klant voor alle ODF's
- stel totale opbrengst gelijk aan 0

1. opzoeken van de volgende gebeurtenis

- kijk wat de eerstvolgende gebeurtenis is
- stel simulatietijd gelijk aan de tijd van de eerstvolgende gebeurtenis
- voer de gebeurtenis uit (2a of 2b)

2a uitvoeren van een aankomst

- als de cumulatieve boekingslimiet groter of gelijk is aan 0
  - verhoog de totale opbrengst
  - verlaag de cumulatieve boekingslimieten van alle ODF's in dezelfde cabines
- plan een nieuwe aankomst voor de betreffende ODF
- ga verder met stap 1

2b uitvoeren van het vertrek/ einde simulatie

- sla de opbrengst van deze simulatie op
- verhoog de totale opbrengst over alle simulaties
- hoog het simulatienummer met 1 op
- als het simulatienummer kleiner is dan het gewenste aantal simulaties (in deze scriptie is gekozen voor 101 simulaties)
  - ga verder met stap 0
- als het simulatienummer gelijk is aan het gewenste aantal simulaties
  - anders verder met stap 3

3 einde alle simulaties

- bepaal het 99 %-betrouwbaarheidsinterval van de opbrengst

**4.3.1 Stationair Poisson-aankomstproces**

Hieronder is beschreven hoe de stappen 1, 2, 3 en een deel van stap 4 gedaan zijn voor het stationaire Poisson-aankomstproces. Bovendien zijn de resultaten van de simulaties weergegeven.

1. bepaling van de verdeling van de vraag

Voor een Stationair Poissonproces is het nodig om te weten wat het verwachte aantal aankomsten per tijdseenheid, de aankomstintensiteit is. Hoewel de database in principe niet beschikt over deze parameter, is deze wel makkelijk af te leiden uit de voorspelling van de vraag,  $\bar{E}D_j$ , die wel uit de database te halen is. Er geldt dat tijdens de boekingsperiode die  $T$  tijdseenheden duurt,

$\bar{E}D_j$  klanten van ODF  $j$  verwacht worden. De aankomstintensiteit  $\lambda_j$ , is dus  $\frac{\bar{E}D_j}{T}$ . Aangezien het niet uitmaakt welke eenheden gebruikt worden, is de boekingsperiode  $T$  gelijkgesteld aan 1, waardoor deze komt te vervallen. Het vliegtuig vertrekt dus op tijdstip 1 en de aankomstintensiteit van ODF  $j$ ,  $\lambda_j$ , is  $\bar{E}D_j$ .

2. bepaling van de parameters die in de modellen ingevuld moeten worden

Nu bekend is welke parameters het aankomstproces heeft, kunnen de parameters voor de modellen bepaald worden. Zoals hierboven uitgelegd, is de verwachting van de vraag gelijk is aan  $\lambda_j = \bar{E}D_j$ .



Nu moet de discrete verdeling van de vraag met drie uitkomsten bepaald worden. Voor de vraag naar ODF  $j$  gedurende de gehele boekingsperiode geldt dat de verdeling nu Poisson verdeeld is met  $\lambda_j = \bar{E}D_j$ . Dit is dan ook de verdeling die gebruikt moet worden bij het toepassen van het algoritme voor het bepalen van de discrete verdeling.

Indien de methode van Kall en Wallace toegepast zou worden, zou voor elke ODF een groot aantal trekkingen verricht moeten worden uit de Poisson-verdeling. Aangezien het trekken uit een Poisson-verdeling ook weer meerdere trekkingen uit de uniforme verdeling vereist, is ervoor gekozen een andere, snellere, methode toe te passen. Deze methode is als volgt.

1. Bepaal de subintervallen  $[n_1, n_2 - 1]$ ,  $[n_2, n_3 - 1]$  en  $[n_3, n_4 - 1]$  waarbij  $n_1 < n_2 < n_3 < n_4$  de kleinste gehele getallen zijn waarvoor geldt:

$$P(X \leq n_1) \geq 0.005$$

$$P(X \leq n_2) \geq \frac{1}{3}$$

$$P(X \leq n_3) \geq \frac{2}{3}$$

$$P(X \leq n_4 - 1) \geq 0.995$$

$$\text{met: } P(X \leq n) = \sum_{x=0}^n p(x) \text{ en } p(x) = \frac{e^{-\lambda} \lambda^x}{x!}$$

2. Bepaal voor elk subinterval de kans dat  $X$  in het interval valt, gegeven dat het in één van de intervallen valt. Deze kans is gelijk aan:  $p_i = \frac{\sum_{x=n_i}^{n_{i+1}-1} p(x)}{\sum_{x=n_i}^{n_4-1} p(x)}$
3. Bepaal voor elk subinterval de verwachtingwaarde van  $X$  gegeven dat deze in het betreffende interval valt. Deze verwachtingwaarde is gelijk aan:  $d_i = \frac{\sum_{x=n_i}^{n_{i+1}-1} xp(x)}{\sum_{x=n_i}^{n_{i+1}-1} p(x)}$
4. De discrete verdeling wordt nu gegeven door:  
 $d_1$  met kans  $p_1$ ,  $d_2$  met kans  $p_2$  en  $d_3$  met kans  $p_3$

### 3. oplossing van de modellen

Nadat de parameters uit stap 2 in de modellen zijn ingevoerd, zijn deze met behulp van CPLEX opgelost. Op deze manier worden de boekingslimieten verkregen die gebruikt worden bij stap 4.

### 4. simulatie van het vraagproces

Voor het simuleren van een Poisson-aankomstproces is het nodig om te bepalen op welke momenten een klant aankomt. Dit wordt gedaan op de methode zoals beschreven door Law en Kelton (2000), door middel van het bepalen van de tussenaankomsttijden, de tijd tussen de aankomst van twee opeenvolgende klanten.

Voor de tussenaankomsttijden van een Poissonproces met aankomstintensiteit  $\lambda$  geldt dat deze exponentieel verdeeld zijn met parameter  $\lambda$ . De aankomsten kunnen nu bepaald worden door vanaf het begin van de periode steeds een aankomst te plannen over  $t$  tijdseenheden, met  $t \sim \text{Exp}(\lambda)$ . Voor het trekken van een tussenaankomsttijd uit de exponentiële verdeling met parameter  $\lambda_j$  moeten de volgende stappen gedaan worden.

1. trek een getal  $U$  uit  $U(0,1)$

2. geef  $t = -\frac{\ln(U)}{\lambda_j}$  terug

**Resultaten**

In onderstaande tabellen zijn de resultaten gegeven van de simulaties en de oplossingen van de modellen. Hierbij geven in de bovenste tabel wederom de vet en cursief gedrukte waarden de verwachte opbrengst uit de modellen weer en de intervallen de 99 %-betrouwbaarheidsintervallen van de verwachte werkelijke opbrengst. In tabel 8 worden de betrouwbaarheidsintervallen voor de verschillen tussen de resultaten van de modellen weergegeven.

	1	deterministische model	2	stochastische model zonder nesting	3	stochastische model met nesting per pad
1		<b><i>1.613.330</i></b>		<b><i>1.536.675</i></b>		<b><i>1.515.594</i></b>
		[ 1.300.994 , 1.310.951 ]		[ 1.571.288 , 1.584.313 ]		[ 1.580.681 , 1.594.249 ]

**Tabel 7** verwachte opbrengst en betrouwbaarheidsintervallen voor de verwachte werkelijke opbrengst bij een stationair Poisson-aankomstproces

model 2 – model 1	model 3 – model 1	model 3 – model 2
[ 268.569 , 275.087 ]	[ 277.839 , 285.145 ]	[ 8.769 , 10.560 ]

**Tabel 8** betrouwbaarheidsintervallen voor de verschillen tussen de drie modellen bij een stationair Poisson-aankomstproces

Uit de tabellen is te zien dat het stochastische model met nesting per pad de hoogste opbrengst oplevert. Het verschil met het stochastische model zonder nesting is echter gering. De resultaten van het deterministische model zijn wel veel lager. Als in de praktijk blijkt dat de aankomsten daadwerkelijk weer te geven zijn met een stationair aankomstproces met een discrete vraagverdeling, dan loont het dus om rekening te houden met de onzekerheid van de vraag.

Aangezien het verschil tussen het model met nesting per pad en dat zonder nesting niet erg groot is, is het niet zonder meer gunstig om nesting toe te voegen. De toevoeging ervan zorgt er namelijk voor dat de rekentijd van de optimalisatie langer duurt. Het is de vraag of de kleine verhoging van de opbrengst opweegt tegen de langere rekentijd.

Het verschil tussen de verwachte opbrengst van het deterministische model en de simulatieresultaten is te verklaren doordat bij het berekenen van de verwachting uitgegaan wordt van een deterministische vraag. Tijdens de simulaties is deze vraag daarentegen wel stochastisch, waardoor de verwachte opbrengst niet behaald wordt.

Bij de stochastische modellen is het verschil echter niet zo simpel te verklaren. Vermoedelijk wordt het verschil veroorzaakt door de onnauwkeurigheid van de benadering van de vraagverdeling. In hoofdstuk 5 zal gekeken worden of het vergroten van het aantal uitkomsten van de discrete benadering een positieve invloed heeft op de nauwkeurigheid van de verwachte opbrengst. Bovendien wordt dan ook gekeken of het verhogen van het aantal uitkomsten leidt tot een hogere opbrengst.

### 4.3.2 Niet-stationair Poisson-proces

In deze subparagraaf zal wederom gekeken worden naar de wijze waarop de vraagverdeling en de modelparameters bepaald en de modellen opgelost kunnen worden. Bovendien wordt het trekken de vraagverdeling en de resultaten van de simulaties besproken. Dit keer is echter gekeken naar een dynamisch vraagproces, waarbij de aankomstintensiteit niet constant is, maar afhangt van de tijd.

#### 1. *bepaling van de verdeling van de vraag*

Net als bij een stationair Poisson-proces, moet ook bij een niet-stationair Poisson-proces eerst bepaald worden wat de parameters zijn die het aankomstproces weergeven. Het enige verschil is dat dit keer de aankomstintensiteit niet constant is, maar varieert over de tijd. Er moet dus geen gemiddelde aankomstintensiteit bepaald worden, maar een functie van de tijd die de aankomstintensiteit weergeeft. Aangezien deze functie niet uit de dataset gehaald kan worden, moet deze gekozen worden aan de hand van de voorspelde vraag. Hiervoor is gebruik gemaakt van de functie die gegeven wordt door de Boer (1999). Met behulp van toetsen laat hij zien dat het aannemelijk is dat de aankomsten verdeeld zijn volgens een niet-stationair Poisson-proces, waarvan de aankomstintensiteit weergegeven kan worden met de volgende formule.

$$\lambda_j(t) = \lambda_j \beta_j(t)$$

met:  $\lambda_j$  = het verwachte aantal gevraagde tickets naar ODF  $j$  tijdens de boekingsperiode

$\beta_j(t)$  = onderstaande gestandaardiseerde kansdichtheid van de bètaverdeling

Voor de gestandaardiseerde bètaverdeling met parameters  $\alpha_j^1$  en  $\alpha_j^2$  wordt de kansdichtheid op het interval  $[0,1]$  als volgt gedefinieerd.

$$\beta_j(t) = \frac{t^{\alpha_j^1-1} (1-t)^{\alpha_j^2-1} \Gamma(\alpha_j^1 + \alpha_j^2)}{\Gamma(\alpha_j^1) \Gamma(\alpha_j^2)}$$

waarbij  $\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t} dt$  voor elk positieve reële getal  $z$

$\Gamma(k+1) = k!$  voor elk niet-negatieve gehele getal  $k$

Aangezien de parameters niet bekend zijn, is ervoor gekozen om deze dusdanig te kiezen dat de aankomstintensiteit van ODF's met een lage rang (hoog rangnummer) in het begin hoog is en aan het eind laag. Voor ODF's met een hoge rang moet het omgekeerde gelden. Dit wordt bereikt door voor de aankomstintensiteit van de ODF met de laagste rang  $\alpha^1$  laag en  $\alpha^2$  hoog te kiezen. De keuze van de parameters is nu als volgt. Voor elk pad worden groepen gemaakt van ODF's die op hetzelfde pad reizen. Vervolgens wordt de grootte van deze groep en het rangnummer bepaald. Als dit bekend is, worden de parameters op de volgende manier gekozen.

$$\alpha_j^1 = G_j - j \text{ en } \alpha_j^2 = j + 1$$

met:  $G_j$  : de grootte van de groep waarin ODF  $j$  zit

De aankomstintensiteiten worden nu dus gegeven met de volgende formule:

$$\lambda_j(t) = \lambda_j(t)^{G_j-r_j-1} (1-t)^{r_j} \frac{G_j!}{(G_j-r_j-1)! j!}$$

Door de parameters op deze manier te kiezen, wordt ervoor gezorgd dat als er veel keus is uit tickets voor dezelfde cabines op een pad, dat dan de vraag naar een ODF uit die groep plaatsvindt tijdens korte hoge pieken. De vraag naar een ODF met een lage rang zal dus tijdens een korte periode aan het begin van de boekingsperiode plaatsvinden en daarna volgen weinig tot geen boekingen voor deze ODF. Voor een ODF met een hoge rang geldt het omgekeerde. Hiervoor komen weinig tot geen klanten aan het begin van de boekingsperiode en de rest komt gedurende een korte periode helemaal aan het eind. Als er weinig keus is uit tickets voor hetzelfde pad, zal de vraag naar de ODF met een lage rang langer duren doordat een ticket van een ODF met één rang hoger meteen veel verschilt in prijs en voorwaarden. Hierdoor komt de vraag in brede lage pieken en is het meer gespreid.

Nu moeten alleen nog de gemiddelde aankomstintensiteiten bepaald worden. Deze worden op dezelfde manier gekozen als bij het stationaire geval. Tijdens de simulatie moet  $\lambda_j$  dus gelijk aan  $ED_j$  genomen worden.

### 2. bepaling van de parameters die in de modellen ingevuld moeten worden

Nu de parameters van het aankomstproces bekend zijn, kunnen ook de benodigde parameters voor de modellen bepaald worden. De verwachtingswaarde is gelijk aan de verwachtingswaarde van een stationair Poisson-proces met als parameter het gemiddelde van de aankomstintensiteit tijdens de duur van het proces (de boekingsperiode). Dus geldt:  $\mu_j = \lambda_j = ED_j$ .

Vervolgens moet wederom de discrete benadering bepaald worden. Net als in het stationaire Poisson-proces wordt hiervoor gekeken naar de vraag tijdens de gehele boekingsperiode. Aangezien de boekingsperiode 1 tijdeenheid betreft, is de vraag Poisson( $\lambda_j$ )-verdeeld. De discrete benadering kan nu op dezelfde manier worden gevonden als in paragraaf 4.3.1.

### 3. oplossing van de modellen

In deze stap worden wederom de modellen opgelost met behulp van het programma CPLEX. Doordat de modelparameters hetzelfde zijn als die bij het stationaire Poisson-proces hoeft dit echter niet opnieuw gedaan te worden.

### 4. simulatie van het vraagproces

Ook bij een niet-stationair Poisson-proces worden de aankomsten bepaald aan de hand van de tussenaankomsttijden. In Law en Kelton (2000) worden een aantal methodes beschreven om een niet-stationair Poissonproces te genereren. Eén van de algoritmes in dit artikel beschreven wordt en dat hier gebruikt is voor de simulatie, is het zogenaamde thinning algoritme. Bij dit algoritme worden aankomsten gegenereerd voor een Poissonproces met een aankomstintensiteit die gelijk is aan de maximale aankomstintensiteit. Vervolgens wordt een deel van de aankomsten weggegooid op basis van de verhouding tussen de aankomstintensiteit op het betreffende tijdstip en de maximale aankomstintensiteit. Stapsgewijs ziet het algoritme er als volgt uit.

1. neem  $t = t_{i-1}$ , de aankomsttijd van de vorige klant
2. genereer  $U_1$  en  $U_2$  als onafhankelijke  $U(0,1)$ -verdeelde variabelen
3. vervang  $t$  door  $t - \frac{\ln(U_1)}{\lambda^*}$  met  $\lambda^* = \max_i \{\lambda(t)\}$
4. als  $U_2 \leq \frac{\lambda(t)}{\lambda^*}$  geef dan  $t_i = t$ , anders ga terug naar stap 2

Om dit algoritme uit te kunnen voeren moet bepaald worden wat de maximale aankomstintensiteit is. Per definitie geldt dat de kansdichtheid van een verdeling maximaal is op het punt van de zogenaamde modus. Voor de bètaverdeling geldt voor de modus het volgende.

$$\text{modus} = \begin{cases} \frac{\alpha_1 - 1}{\alpha_1 + \alpha_2 - 2} & \text{als } \alpha_1 > 1, \alpha_2 > 1 \\ 0 \text{ en } 1 & \text{als } \alpha_1 < 1, \alpha_2 < 1 \\ 0 & \text{als } \alpha_1 < 1 \text{ en } \alpha_2 \geq 1 \text{ of } \alpha_1 = 1, \alpha_2 > 1 \\ 1 & \text{als } \alpha_1 \geq 1 \text{ en } \alpha_2 < 1 \text{ of } \alpha_1 > 1, \alpha_2 = 1 \\ [0, 1] & \text{als } \alpha_1 = 1 \text{ en } \alpha_2 = 1 \end{cases}$$

Om de maximale aankomstintensiteit te bepalen, wordt gekeken naar de waarde van  $\alpha_1$  en  $\alpha_2$ . Als deze niet beide gelijk zijn aan 1, dan wordt voor de tijd de waarde genomen die gelijk is aan de modus. In het geval waarbij ze wel beide gelijk zijn aan 1, maakt de tijd die gekozen wordt niet uit, aangezien de verdelingsfunctie een constante functie is. In dat geval wordt de tijd vastgesteld op 0. Vervolgens wordt de maximale aankomstintensiteit verkregen door de gevonden tijd in te vullen in de formule voor de aankomstintensiteit.

Het verkregen aankomstproces is toegepast in de simulaties.

Resultaten

Met de simulatie zijn de resultaten verkregen die in onderstaande tabellen zijn weergegeven. In tabel 9 is te zien wat de verwachte opbrengsten uit de modellen zijn (vet en cursief gedrukt) en wat de betrouwbaarheidsintervallen van de verwachte werkelijke opbrengsten zijn. Tenslotte is in tabel 10 te vinden wat de betrouwbaarheidsintervallen van de verschillen van de opbrengsten van de drie modellen zijn.

	1	deterministische model	2	stochastische model zonder nesting	3	stochastische model met nesting per pad
1		<b><i>1.613.330</i></b>		<b><i>1.536.675</i></b>		<b><i>1.515.594</i></b>
		[ 1.298.968 , 1.309.883 ]		[ 1.578.402 , 1.592.297 ]		[ 1.582.327 , 1.596.467 ]

**Tabel 9** verwachte opbrengst en betrouwbaarheidsintervallen voor de verwachte werkelijke opbrengst bij een niet-stationair Poisson-aankomstproces

model 2 – model 1	model 3 – model 1	model 3 – model 2
[ 277.530 , 284.318 ]	[ 281.458 , 288.485 ]	[ 3.326 , 4.768 ]

**Tabel 10** betrouwbaarheidsintervallen voor de verschillen tussen de drie modellen bij een niet-stationair Poisson-aankomstproces

Ook voor dit vraagproces blijkt uit de resultaten dat het stochastische model met nesting per pad beter presteert dan het deterministische en dat het verschil met het model zonder nesting klein is. Als in de praktijk blijkt dat de aankomsten van klanten te modelleren zijn als een niet-stationair aankomstproces met een zelfde soort verloop van de aankomstintensiteit, is het zinvol om het stochastische element toe te voegen. Voor het toevoegen van nesting valt niet zonder meer te zeggen of het kleine voordeel opweegt tegen de nadelen zoals de langere rekentijd van het model. Voor de verschillen tussen de verwachte opbrengst en de opbrengsten uit de simulaties gelden dezelfde oorzaken. Voor het deterministische geval wordt het verschil veroorzaakt door het feit dat bij het berekenen van de verwachting uitgegaan wordt van een zekere vraag terwijl deze bij de simulaties onzeker is. Voor de stochastische modellen moet de oorzaak waarschijnlijk gezocht worden in de onnauwkeurigheid van de discrete benadering. In hoofdstuk 5 wordt hier dit nader bekeken.

## 5. Verbeteringen voor het model

Om de modellen te kunnen maken, zijn tijdens het modelleringsproces een aantal aannames gedaan. In dit hoofdstuk zal gekeken worden in hoeverre het aanpassen van enkele van deze aannames de resultaten verbeteren. Hiervoor worden de modellen iets aangepast, waarbij de aannames weggelaten worden. Vervolgens worden de modellen opgelost en wordt het boekingsproces gesimuleerd. De nieuwe resultaten worden vergeleken met die van het vorige hoofdstuk, om te zien in hoeverre de gemaakte aanname invloed heeft op de kwaliteit van de oplossing.

Het eerste punt waarnaar gekeken is, is het discretiseren van de vraagverdeling met een discrete verdeling met drie uitkomsten. Dit is gedaan om het deterministische equivalent klein te houden, waardoor het snel op te lossen is. In paragraaf 5.1 zal gekeken worden of de oplossingen beter worden als het aantal uitkomsten verhoogd wordt tot 9. De nieuwe benadering is nog steeds niet exact, maar zal beter blijken te zijn dan de originele benadering.

Daarnaast wordt gekeken naar de invloed van de aanname dat klanten uit een lage klasse het eerst boeken. Met een simulatie worden verschillende aankomstprocessen gesimuleerd en met elkaar vergeleken. In het vorige hoofdstuk is al gekeken naar twee mogelijkheden. Hieraan zullen in dit hoofdstuk nog twee worden toegevoegd.

### 5.1 Discretisering

Tijdens de simulaties is gebruik gemaakt van een discrete verdeling met drie uitkomsten, die de eigenlijke verdeling van de vraag moet benaderen. Hoewel deze benadering niet nauwkeurig is, is het de vraag in hoeverre de onnauwkeurigheid invloed heeft op de kwaliteit van de oplossingen van het model. Daarom zal in deze paragraaf gekeken worden naar de invloed van het vergroten van het aantal uitkomsten van de discrete verdeling. Hierbij wordt het discrete model niet meer bekeken, aangezien een verbetering van de discretisering geen invloed heeft op dit model.

Om de kwaliteit van de nieuwe oplossing te vergelijken met de oorspronkelijke uit hoofdstuk 4, wordt wederom een aantal simulaties uitgevoerd. Ten eerste zal een simulatie worden uitgevoerd voor het geval waarbij de vraag normaal-verdeeld wordt verondersteld. Deze simulatie is vrijwel identiek aan die uit subparagraaf 4.2.2. In subparagraaf 5.1.1 is te zien hoe de oorspronkelijke simulatie aangepast is om het model met 9 uitkomsten op te lossen en het bijbehorende boekingsproces te simuleren.

Daarnaast zal hetzelfde gedaan worden voor de Poisson-verdeling. Deze is gebruikt bij het simuleren van de dynamische vraagprocessen. Aangezien de verschillen tussen de verwachting van de opbrengst iets onder de betrouwbaarheidsintervallen van de opbrengst bleek te liggen, is gekeken of dit niet meer het geval is als het aantal uitkomsten verhoogd wordt tot 9. Daarom wordt een soortgelijke simulatie uitgevoerd als in hoofdstuk 4 voor het geval waarbij de vraag dynamisch is. Hierbij is alleen gekeken naar de simulatie van het statische Poisson-aankomstproces. Voor het niet-statische geval zal de invloed van de onnauwkeurige benadering vergelijkbaar zijn. In 5.1.2 is te zien hoe de simulatie aangepast moet worden.

### 5.1.1 Normaal verdeelde vraag

In deze paragraaf is te zien of en hoe de stappen uit subparagraaf 4.2.2 aangepast moeten worden als de discrete benadering uitgebreid wordt. Daarnaast worden de resultaten van de simulaties gegeven en vergeleken met de eerder gevonden resultaten waarbij uitgegaan is van een benadering met drie uitkomsten.

#### 1. *bepaling van de verdeling van de vraag*

In principe verandert deze stap niet. Dit keer zal echter alleen gekeken worden naar de verdeling met een variatiecoëfficiënt die gelijk is aan 0.7. Hiervoor is gekozen omdat bij deze waarde het verschil tussen de verwachting en de gesimuleerde opbrengst het hoogst was. De verwachting is dat dit verschil kleiner wordt door een betere benadering te nemen. Bovendien zal een betere benadering ook leiden tot een hogere opbrengst, doordat het model de 'werkelijkheid' beter beschrijft.

#### 2. *bepaling van de parameters die in de modellen ingevuld moeten worden*

De enige aanpassing in de tweede stap is de wijze waarop de discrete benadering wordt verkregen. Naast het vergroten van het aantal uitkomsten, zal dit keer in het algoritme ook rekening worden gehouden met het feit dat de normale verdeling beperkt is tot de niet-negatieve waarden. Het algoritme voor het verkrijgen van de discrete benadering met  $K$  uitkomsten, is dan als volgt:

##### 1. *Bepaling van het 99 %-interval van $N(\mu, \sigma^2)$*

Het 99 %-betrouwbaarheidsinterval van de normale verdeling is onveranderd en gelijk aan  $[\mu - \alpha\sigma, \mu + \alpha\sigma]$ , met  $\alpha = 2,5758$ .

##### 2. *Beperking van het interval tot de positieve waarden*

Aangezien de vraagverdeling beperkt is tot de positieve waarden, wordt het betrouwbaarheidsinterval beperkt tot het positieve deel. Het interval wordt dan als volgt:

$$[\mu - \alpha\sigma, \mu + \alpha\sigma] \quad \text{als } \frac{\sigma}{\mu} \leq \frac{1}{\alpha} = 0.39$$

$$[0, \mu + \alpha\sigma] \quad \text{als } \frac{\sigma}{\mu} > \frac{1}{\alpha} = 0.39$$

In het vervolg van dit algoritme wordt uitgegaan van het tweede geval, aangezien een betere benadering gevonden moet worden voor de normale verdeling waarbij de variatiecoëfficiënt 0.7 is.

##### 3. *De opsplitsing van het interval.*

Het interval moet opgesplitst worden in  $K$  gelijke intervallen. Op deze manier wordt het  $k^e$  interval gelijk aan  $\left[ \frac{(k-1) - K}{K}(\mu + \alpha\sigma), \frac{k - K}{K}(\mu + \alpha\sigma) \right]$

##### 4. *Bepaling van de kansdichtheid van de beperkte normale verdeling*

Doordat de verdeling beperkt is tot de positieve waarden, is de kansdichtheid niet meer gelijk aan die van de normale verdeling. Om de juiste kansdichtheid te krijgen moet die van de normale verdeling gedeeld worden door de kans dat de normale verdeling in het interval valt. Deze voorwaardelijke kansdichtheid wordt nu als volgt.



$$\hat{f}(x) = \frac{f(x)}{\int_0^{\mu+\alpha\sigma} f(x) dx} \quad \text{met } f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \text{ de kansdichtheid van } N(\mu, \sigma^2).$$

5. *Bepaling van de verwachte waarde binnen de subintervallen.*

Voor interval  $[a, b]$  wordt de verwachtingswaarde gegeven door de volgende formule.

$$E[X | a \leq X \leq b] = \frac{\int_a^b x \hat{f}(x) dx}{\int_a^b \hat{f}(x) dx} \quad \text{met } \hat{f}(x), \text{ de kansdichtheid uit de vorige stap.}$$

Voor 9 subintervallen worden zo de volgende verwachtingswaarden verkregen:  $0.17\mu$ ,  $0.48\mu$ ,  $0.78\mu$ ,  $1.09\mu$ ,  $1.39\mu$ ,  $1.70\mu$ ,  $2.01\mu$ ,  $2.31\mu$  en  $2.62\mu$

6. *Bepaling van de kansen op een waarde binnen de subintervallen.*

Voor interval  $[a, b]$  geldt dat de kans gegeven wordt door de volgende formule.

$$P(a \leq x \leq b | 0 \leq x \leq \mu + \alpha\sigma) = \int_a^b \hat{f}(x) dx \quad \text{met } \hat{f}(x) \text{ de kansdichtheid uit stap 4.}$$

Voor 9 subintervallen worden zo de volgende kansen verkregen:  $0.10$ ,  $0.14$ ,  $0.18$ ,  $0.19$ ,  $0.16$ ,  $0.12$ ,  $0.07$ ,  $0.03$  en  $0.01$

### 3. oplossing van de modellen

Aan de modellen zoals die uitgeschreven zijn in hoofdstuk 3 veranderd niet. Uitgeschreven zien ze er echter wel anders uit als bij de simulatie met drie uitkomsten. Doordat het aantal uitkomsten groter is, zijn er meer recourse variabelen nodig. De LP-modellen die verkregen worden na het invullen van de parameters zijn op te lossen met behulp van CPLEX.

### 4. simulatie van het vraagproces

Aan de simulatie van het vraagproces verandert niets, aangezien de vraagverdeling hetzelfde blijft als in de simulatie in hoofdstuk 4.

### Resultaten

In onderstaande tabel zijn de resultaten gegeven van de simulaties waarbij de vraag normaal verdeeld is verondersteld. Wederom zijn de vet en cursief gedrukte waarden de verwachte opbrengsten en de intervallen de betrouwbaarheidsintervallen van de opbrengsten van de modellen. Het waarde tussen haakjes geeft de tijd die nodig is om het LP-model op te lossen weer.

	1	stochastische model zonder nesting	2	stochastische model met nesting per pad
simulatie 1 (3-punts benadering)		<b>1.610.933</b> ( 3 min. )		<b>1.611.213</b> ( 12 min.)
		[ 1.768.403 , 1.773.542 ]		[ 1.771.682 , 1.776.844 ]
simulatie 2 (9-punts benadering)		<b>1.762.137</b> ( 17 min.)		<b>1.762.658</b> ( 2,5 dag )
		[ 1.784.679 , 1.790.080 ]		[ 1.784.756 , 1.790.151 ]
simulatie 2 – simulatie 1		[ 16.073 , 16.742 ]		[ 12.885 , 13.497 ]

**Tabel 11** verwachte opbrengst, rekentijd en betrouwbaarheidsintervallen voor de opbrengst bij niet-stationair Poisson-aankomstproces en het verschil in opbrengst bij het verhogen van het aantal uitkomsten

In de tabel is te zien dat door het vergroten van het aantal uitkomsten, de verwachte opbrengst uit de modellen dichter bij de verwachte werkelijke opbrengst komt te liggen. Dit is het geval bij beide stochastische modellen. Bovendien zorgt de verbetering voor een hogere opbrengst uit de simulaties. Het verhogen van het aantal uitkomsten dus een positieve invloed op de resultaten. De vraag is echter of de positieve invloed opweegt tegen de vergroting van de rekentijd. Voor het model zonder nesting valt deze vergroting mee, maar bij het model met nesting per pad is de vergroting onacceptabel groot. Als gekozen wordt voor een model met nesting per pad is het daarom af te raden om het aantal uitkomsten te vergroten.

### 5.1.2 Poisson-verdeelde vraag

Hieronder zal voor de Poissonverdeling bekeken worden welke aanpassingen gedaan zijn om de discrete benadering te verbeteren. Vervolgens zullen de resultaten van de nieuwe simulaties vergeleken worden met die uit hoofdstuk 4. Hierbij is echter alleen gekeken naar het model zonder nesting, aangezien uit de vorige paragraaf bleek dat de rekentijd voor het model met nesting per pad erg lang is. De verwachting is dat voor het model met nesting per pad de veranderingen vergelijkbaar zijn.

#### 1. *bepaling van de verdeling van de vraag*

Deze stap blijft ongewijzigd, aangezien hier niets wordt gedaan met de benadering van de vraagverdeling.

#### 2. *bepaling van de parameters die in de modellen ingevuld moeten worden*

Van de parameters die ingevuld moeten worden, verandert alleen de discrete benadering. Voor het benaderen van een Poissonverdeling met een discrete verdeling met  $K$  uitkomsten is de methode als volgt.

1. Bepaal de subintervallen  $[n_1, n_2 - 1], [n_2, n_3 - 1] \dots [n_K, n_{K+1} - 1]$  waarbij  $n_1 < n_2 < \dots < n_{K+1}$  de kleinste gehele getallen zijn waarvoor geldt:

$$P(X \leq n_1) \geq 0.005$$

$$P(X \leq n_2) \geq \frac{1}{K}$$

$$\vdots$$

$$P(X \leq n_K) \geq \frac{K-1}{K}$$

$$P(X \leq n_{K+1} - 1) \geq 0.995$$

$$\text{met: } P(X \leq n) = \sum_{x=0}^n p(x) \text{ en } p(x) = \frac{e^{-\lambda} \lambda^x}{x!}$$

2. Bepaal voor elk sub-interval de kans dat  $X$  in het interval valt, gegeven dat het in één van de intervallen valt. Deze kans is gelijk aan:  $p_i = \frac{\sum_{x=n_i}^{n_{i+1}-1} p(x)}{\sum_{x=n_i}^{n_{i+1}-1} p(x)}$
3. Bepaal voor elk sub-interval de verwachtingwaarde van  $X$ , gegeven dat deze in het betreffende interval valt. Deze verwachtingwaarde is gelijk aan:  $d_i = \frac{\sum_{x=n_i}^{n_{i+1}-1} xp(x)}{\sum_{x=n_i}^{n_{i+1}-1} p(x)}$
4. De discrete verdeling wordt nu gegeven door  $d_1$  met kans  $p_1, d_2$  met kans  $p_2, \dots, d_K$  met kans  $p_K$

### 3. oplossing van de modellen

Bij het oplossen van de modellen hoeft alleen het aantal recourse variabelen verhoogd te worden. De opzet van het model blijft echter ongewijzigd.

### 4. simulatie van het vraagproces

Net zoals bij het geval van de normale verdeling verandert ook hier niets bij de simulatie.

### Resultaten

In onderstaande tabellen zijn de resultaten gegeven van de simulaties. Hierbij geven wederom de waarden vet en cursief gedrukte waarden de verwachte opbrengst weer en de overige waarden de gemiddelde opbrengst uit de simulaties.

	stochastische model zonder nesting
simulatie 1 3-punts benadering	<b>1.536.675</b> [ 1.571.288 , 1.584.313 ]
simulatie 2 9-punts benadering	<b>1.579.932</b> [ 1.583.516 , 1.596977 ]
simulatie 2 – simulatie 1	[ 11.548 , 13.344 ]

**Tabel 12** verwachte opbrengst en betrouwbaarheidsintervallen voor de verwachte werkelijke opbrengst bij stationair Poisson-aankomstproces

Uit de tabel is te zien dat het verhogen van het aantal realisaties een positieve invloed heeft op de opbrengst uit de simulatie. De vraag is echter of de positieve invloed opweegt tegen het langer duren van de optimalisatie. Daarnaast is de verwachte opbrengst een betere benadering van de verwachting voor de daadwerkelijke opbrengst bij het toepassen van de gevonden boekingsstrategie.

## 5.2 Boekingsvolgorde

In deze paragraaf wordt bekeken wat voor invloed de boekingsvolgorde heeft op de daadwerkelijk behaalde opbrengst. Er verandert dus uitsluitend iets aan de simulaties en niet aan de modellen en de parameters die in de modellen ingevoerd worden. In de rest van deze paragraaf zullen een aantal verschillende simulaties gedaan worden. Ten eerste zal een vraag gesimuleerd worden waarbij de vraag Poisson-verdeeld is, maar het aankomstproces niet dynamisch is. Op deze manier wordt een vraagproces gesimuleerd waarbij de lage ODF's als eerste boeken en er geen andere ODF's boeken voor de laatste klant uit die ODF's. Daarnaast wordt een vraagproces gesimuleerd die tegenovergesteld is aan het niet-statische aankomstproces uit subparagraaf 4.3.2. Dan boeken de hoge ODF's dus vroeg, maar zijn de aankomsten niet strikt gescheiden. Dit is gedaan om te bekijken in hoeverre de opbrengst lager uitvalt als blijkt dat de aanname dat ODF's met een lage rang eerst boeken niet juist blijkt te zijn. De resultaten van deze simulaties worden vergeleken met die van de simulaties die in paragraaf 4.3 gevonden zijn.

### 5.2.1 Statisch vraagproces met Poisson-verdeelde vraag

In deze subparagraaf is beschreven hoe de simulatie voor dit geval gedaan is. In tabel 13 zijn de resultaten van de simulaties te vinden.

In het geval van een statisch vraagproces met een Poisson-verdeelde vraag, wordt een statische simulatie uitgevoerd zoals beschreven in paragraaf 4.2. In de simulatie moeten trekkingen verricht worden uit de Poisson( $\lambda$ )-verdeling. Hiervoor moeten de volgende stappen gedaan worden:

- I. neem aantal = 0, tijd = 0
- II. trek de getallen  $U$  uit  $\exp(\lambda)$  (zoals beschreven in subparagraaf 4.3.1)
- III. tijd = tijd +  $U$ , aantal = aantal + 1
- IV. als tijd < 1, ga terug naar stap II  
anders geef aantal - 1 terug

### 5.2.2 Poisson-aankomstproces, hoge ODF's boeken eerst

Hieronder wordt beschreven op welke manier de simulatie waarbij de hoge ODF's gemiddeld het eerst boeken, gedaan is. Ook van deze simulaties zijn de resultaten in tabel 13 te vinden.

De enige aanpassing ten opzichte van de simulatie voor het Poisson-aankomstproces waarbij de lage ODF's eerst boeken (zie paragraaf 4.3.2), is de functie die de aankomstintensiteit weergeeft. Door het aanpassen van de parameters van de beta-verdeling ( $\alpha_j^1$  en  $\alpha_j^2$ ) kan ervoor gezorgd worden dat de pieken net omgekeerd komen. In de simulaties zijn de volgende waarden gekozen voor de parameters.  $\alpha_j^1 = j + 1$  en  $\alpha_j^2 = G_j - j$ . Naast deze aanpassing hoeft verder niet aangepast te worden

### 5.2.3 Resultaten

In onderstaande tabel zijn de resultaten te vinden van de verschillende simulaties. De simulaties die uitgevoerd zijn, zijn de volgende:

simulatie 1: statisch vraagproces met Poisson-verdeelde vraag (§ 5.2.1)

simulatie 2: Poisson-aankomstproces met stationaire aankomstintensiteit (§ 4.3.1)

simulatie 3: niet-stationair Poisson-aankomstproces, lage ODF's boeken eerst (§ 4.3.2)

simulatie 4: niet-stationair Poisson-aankomstproces, hoge ODF's boeken eerst (§ 5.2.2)

	1	deterministische model	2	stochastische model zonder nesting	3	stochastische model met nesting per pad
verwachting	<b>1.613.330</b>		<b>1.536.675</b>		<b>1.515.594</b>	
simulatie 1	[ 891.644 , 897.320 ]		[ 1.544.883 , 1.557.560 ]		[ 1.550.569 , 1.563.359 ]	
simulatie 2	[ 1.287.338 , 1.297.156 ]		[ 1.571.288 , 1.584.313 ]		[ 1.580.681 , 1.594.249 ]	
simulatie 3	[ 1.298.968 , 1.309.883 ]		[ 1.578.402 , 1.592.297 ]		[ 1.582.327 , 1.596.467 ]	
simulatie 4	[ 1.258.925 , 1.269.398 ]		[ 1.532.929 , 1.546.809 ]		[ 1.557.903 , 1.572.251 ]	

**Tabel 13** verwachte opbrengst en betrouwbaarheidsintervallen voor de opbrengsten bij verschillende aankomstprocessen

Uit een vergelijking van simulatie 1 en 3 is te zien dat het in alle gevallen ongunstig is als de boekingsperioden strikt gescheiden zijn. Vooral voor het deterministische model is dit verschil groot. Dit is te verklaren doordat in de eerste simulatie de ODF's strikt gescheiden arriveren en bij de derde niet. Daardoor kunnen hogere klassen bij de derde simulatie gebruik maken van stoelen van lagere klassen die nog wel leeg zijn, maar waar later wel vraag naar komt. Hierdoor worden de stoelen gebruikt voor een hogere prijsklasse en is de opbrengst hoger.

Daarnaast is uit de vergelijking van simulatie 2,3 en 4 te zien dat hoe meer de volgorde afwijkt van datgene wat gebruikt is bij de modellering, hoe slechter de resultaten zijn. Bij simulatie 3 komt het het best overeen (lage ODF's boeken aan het begin), bij simulatie 2 minder goed (ODF's boeken verspreid over de gehele periode) en bij simulatie 4 het minst goed (lage ODF's boeken aan het eind). Als de boekingsperiode beïnvloed kan worden, verdient het dus de voorkeur om ervoor te zorgen dat deze overeenkomt met datgene wat gebruikt is tijdens de modellering. Dit kan bijvoorbeeld door het instellen van een boekingsperiode waarbinnen tickets van een bepaalde ODF enkel gekocht kunnen worden en daarbuiten niet.

## 6. Conclusies en aanbevelingen

In deze scriptie zijn een aantal modellen gemaakt voor het zogenaamde seat-allocation probleem. Door de oplossing van deze modellen worden boekingslimieten verkregen voor de verschillende ODF's. Aan de hand van deze boekingslimieten kan de optimale boekingspolitiek bepaald worden. Bij de modellering is met verschillende factoren rekening gehouden. Het eerste model is een deterministisch model. Hierbij is rekening gehouden met het netwerk-effect en de toepassing van upgrading. Bij dit model is er vanuit gegaan dat de vraag naar de verschillende ODF's bekend is. Het tweede model is een deterministisch equivalent van het stochastische model zonder nesting. Bij dit model is wederom rekening gehouden met het netwerk-effect en de toepassing van upgrading. Daarnaast is rekening gehouden met de onzekerheid van de vraag. Om het deterministische equivalent te krijgen is aangenomen dat de verdeling van de vraag een discrete verdeling heeft. Daarnaast is nog een deterministisch equivalent voor het stochastische model met volledige nesting gemaakt. Het enige verschil van het vorige model is dat het rekening houdt met de toepassing van volledige nesting.

Tenslotte is nog een deterministisch equivalenten voor het stochastische model met nesting per pad gemaakt. Dit model houdt rekening met nesting per pad, maar is verder gelijk aan de vorige twee modellen.

Naast een beschrijving van de modellen is in deze scriptie ook een simulatie te vinden. Hiermee is gekeken hoe de verschillende modellen ten opzichte van elkaar presteren in de werkelijkheid. Hierbij is het model met volledige nesting buiten beschouwing gelaten, aangezien dit model te groot was om op te lossen. Bij simulaties is gebruik gemaakt van een netwerk van een bestaande vliegtuigmaatschappij voor vier dagen.

Aangezien geen gegevens beschikbaar waren over de verdeling van de vraag en het aankomstproces van de klanten, zijn simulaties met verschillende vraagverdelingen een aankomstproces uitgevoerd. Voor elke simulatie is de opbrengst uit de simulatie voor de drie modellen met elkaar vergeleken en zijn ze vergeleken met de verwachte opbrengst uit de oplossing van de modellen. Hieronder zijn in het kort de conclusies uit deze vergelijking gegeven. Naast de conclusies van het onderzoek zullen in dit hoofdstuk ook een aantal aanbevelingen gedaan worden voor nader onderzoek.

### 6.1 Conclusies

Uit de simulaties valt een aantal conclusies te maken over het deterministische model en de stochastische modellen zonder nesting en met nesting per pad.

Allereerst blijkt dat de verwachte opbrengst uit het deterministische model geen goede indicatie is voor de verwachte werkelijke opbrengst. Voor de overige modellen is de verwachting uit de modellen weliswaar nog steeds niet exact de verwachte werkelijke waarde, maar geeft het wel een goede indicatie van de verwachting van de totale opbrengst.

Daarnaast blijkt uit de simulatie dat het toevoegen van de stochasticiteit van de vraag aan het model een positief effect heeft op de opbrengst. De boekingsstrategieën die verkregen wordt bij de nieuwe modellen leveren een stijging van de opbrengst van 5 tot 20 % op, afhankelijk van de verdeling van de vraag.

Over het nut van het toevoegen van nesting is geen eenduidige uitspraak te doen. Hoewel de opbrengst bij het model met nesting per pad wel het hoogst is, is het verschil in alle gevallen slechts klein en soms zelfs negatief. Het is daarom niet zonder meer te zeggen of de toevoeging van nesting per pad al dan niet zinvol is.

Bovendien valt het nut van uitbreiding van de discretisering ook te betwijfelen. Een vergroting van het aantal uitkomsten van 3 naar 9 heeft een verhoging van de opbrengst uit de simulaties met minder dan 1% tot gevolg, terwijl de rekentijd aanzienlijk toeneemt. Het is de vraag welke van deze twee gevolgen het zwaarst meeweegt en beslissend is.

Tenslotte is te zien dat de volgorde van het boeken geen invloed heeft op de prestaties van de modellen ten opzichte van elkaar. Uit simulaties van verschillende volgorden blijkt dat het model dat bij een volgorde de hoogste (laagste) opbrengst heeft, ook bij de andere volgorden de hoogste (laagste) opbrengst heeft. Wel heeft de boekingsvolgorde invloed op de opbrengst. Het gunstigst is de situatie waarbij de klanten met een lage rang over het algemeen aan het begin van de boekingsperiode boeken, maar waarbij overlap met andere ODF's ook wel is toegestaan.

## 6.2 Aanbevelingen

In deze paragraaf worden een aantal aanbevelingen gedaan voor nader onderzoek. Hierbij ligt de nadruk op de controle van de gemaakte aannames en het verbeteren van het model.

### Vraagverdeling

Allereerst is het belangrijk om meer gegevens te hebben over de werkelijke verdeling van de vraag. Deze is nodig voor het bepalen van de parameters die ingevuld moeten worden in de modellen. Zonder juiste parameters is het onmogelijk om een model te maken dat met zekerheid een optimale boekingspolitiek vindt.

Daarnaast bleek uit de simulaties dat de vraagverdeling van invloed is op het nut van het toevoegen van nesting. Als een keuze gemaakt moet worden over het al dan niet toevoegen van nesting, is het dus zinvol om eerst meer informatie te hebben over de werkelijke vraagverdeling.

### Afhankelijkheid

Bij de modellering van het model met nesting per pad is aangenomen dat de vraag naar de verschillende ODF's gelijktijdig een hoge, lage of gemiddelde waarde aannemen. Hoewel deze aanname reëel lijkt, is deze niet gecontroleerd. Er zou daarom nog naar gekeken kunnen worden in hoeverre deze aanname overeenkomt met de werkelijkheid.

Een andere optie is om te kijken in hoeverre de resultaten veranderen als deze aanname onterecht blijkt te zijn. Als de resultaten niet veel veranderen, kan de aanname blijven staan zonder dat deze gecontroleerd wordt.

### Oplossing van de modellen

In deze scriptie is uitgegaan van de oplossingsmethode die op dit moment bij ORTEC gebruikt wordt voor het oplossen van de deterministische modellen. Met deze oplossingsmethode was het niet mogelijk om het model met volledige nesting op te lossen, aangezien dit te groot werd en niet in CPLEX ingevoerd kon worden. Er zou daarom gekeken kunnen worden naar methoden om grotere modellen op te lossen. Wellicht dat daarmee het model met volledige nesting ook

opgelost kan worden.

Daarnaast zou gekeken kunnen worden naar een methode om de grotere LP-problemen sneller op te lossen. Hierdoor kunnen de modellen die verkregen worden door de discrete verdelingen uit te breiden, sneller worden opgelost, waardoor de geringe verhoging van de opbrengst alsnog opweegt tegen de verhoging van de rekentijd.



## Veel gebruikte afkortingen en symbolen

- $b_j$  : cumulatieve boekingslimiet van ODF  $j$   
 $c_i$  : aantal zitplaatsen in cabine  $i$   
 $D_j$  : de stochastische vraag naar ODF  $j$   
 $d_j$  : verwachte vraag naar ODF  $j = \overline{ED}_j$   
 $d_j^n$  : de  $n^e$  realisatie van de vraag naar ODF  $j$   
 $f_j$  : prijs van ODF  $j$   
 $G_j$  : Het aantal ODF's op het pad van ODF  $j$   
 $i$  : een bepaalde cabine  
 $I$  : aantal cabines  
 $I_j$  : aantal cabines van ODF  $j$   
 $I(j)$  : alle cabines die ODF  $j$  bevatten.  
 $j$  : een bepaalde ODF  
 $J$  : aantal ODF's  
 $J(i)$  : de verzameling van alle ODF's in cabine  $i$   
 $n$  : een bepaalde uitkomst  
 $N_j$  : aantal waarden dat de vraag naar ODF  $j$  kan aannemen  
 $N$  : aantal waarden dat de vraag naar een ODF kan aannemen als dat voor alle ODF's gelijk is  
 ODF : combinatie van vertrekpunt, bestemming, plaats van aankoop, reisdatum en prijsklasse  
 $P(j)$  : het pad van ODF  $j$   
 $p_j^n$  : kans van ODF  $j$  op de  $n^e$  realisatie van de vraag  
 $s_l$  : aantal passagiers van cabine  $l$  worden overgeplaatst naar cabine  $h$ .  
 s.t: subject to = onder de voorwaarde  
 $T$  : de duur van de boekingsperiode  
 $x_j$  : boekingslimiet van ODF  $j$   
 $y_j$  : aantal zitplaatsen die verkocht worden in ODF  $j$   
 $y_j^n$  : aantal zitplaatsen die verkocht worden in ODF  $j$  bij realisatie  $n$

## Literatuurlijst

- [1] Bertsimas, D. en Boer, S. de (2000), A stochastic booking-limit control policy for airline network revenue management, working paper, Operations Research Center, Massachusetts Institute of Technology.
- [2] Birge, J.R. en Louvreaux, F (1997), *Introduction tot stochastic programming*, eerste druk, New York, Springer.
- [3] Boer, S.V. de (1999), *Mathematical programming in airline seat inventory control*, afstudeerscriptie vrije universiteit Amsterdam, afdeling wiskunde.
- [4] Boer, S.V. de, e.a. (1999), Stochastic programming for multiple-leg network revenue management, *Econometric Institute Report EI-9935/A*, Rotterdam.
- [5] Kall, P. en Wallace, S.W. (1994), *Stochastic Programming*, Wiley, John & Sons, Incorporated.
- [6] Lautenbacher, C.J. en Stidham, S. jr. (1999), The underlying markov decision process in the single-leg airline yield-management problem, *transportation science*, vol. 33, afl. 2, pag. 136-146.
- [7] Law, A.M en Kelton, W.D. (2000), *Simulation modeling and analysis*, derde editie, Singapore, McGraw-Hill Co.