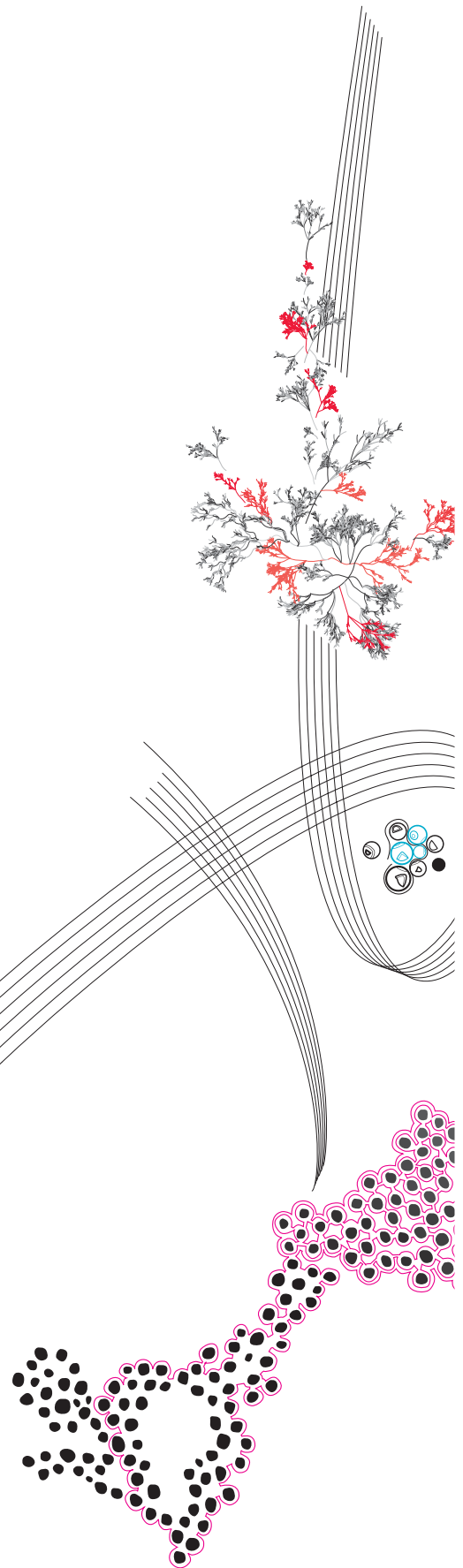




BSc Thesis Applied Mathematics

**Energiezuinige verdeling van
taken over verschillende
processoren in een 'embedded'
computersysteem**

Rianne Arlonne Meenhuis

Supervisor: Dr. Ir. M. de Graaf

3 juli 2020

Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science

Voorwoord

Dit verslag is geschreven ter afsluiting van mijn bachelorstudie "Technische Wiskunde". Dat dit verslag op tijd af is gekomen, voelt voor mij als een overwinning. Ik heb namelijk een tijd lang niet gestudeerd, waardoor ik er eerst weer in moest komen. Wat me wel wat onzekerder maakte. Maar uiteindelijk ging het erin komen, toch beter dan verwacht.

Om mij hierbij te helpen, hebben Lilian Spijker en Judith Timmer ervoor gezorgd dat ik een kamer kreeg op de universiteit. Daarvoor zou ik ze toch graag willen bedanken, ondanks dat ik hier weinig aan heb gehad vanwege de coronamaatregelen. Daarnaast wil ik ze ook nog bedanken voor alle steun en andere hulp die ze mij hierbij geboden hebben.

Daarnaast zou ik ook mijn ouders willen bedanken, vanwege de aanpassingen die ze hebben gedaan om voor mij een goede werksfeer en werkomgeving te creëren. Dat deze werkplek thuis nodig was, kwam doordat ik noodgedwongen thuis moest werken vanwege de coronamaatregelen. Dit was niet ideaal, vooral niet op warme dagen. Daarnaast zou ik mijn ouders ook nog willen bedanken voor het doorlezen van mijn verslag, waardoor er meer taalfouten uit zijn gehaald.

Ook zou ik mijn broer en diens vriendin willen bedanken voor het mogen lenen van een beeldscherm, waardoor ik op warme dagen comfortabeler op een andere plek in huis kon werken. Voor het helpen bij BibTeX, zou ik graag ook nog Nicky Schuermans willen bedanken. Aangezien ik hier erg mee heb zitten worstelen. Verder zou ik ook nog al mijn vriendinnen en een aantal andere mensen willen bedanken, voor de steun en het vertrouwen die ze al die tijd in me hebben gehad. Vooral omdat ik zelf, met name in het begin, hier nog niet zo van overtuigd was.

Tot slot gaat mijn grootste dank uit naar mijn begeleider Maurits de Graaf. Deze heeft me, zeer zeker in het begin, goed op weg geholpen. Daarnaast kon ik altijd met vragen goed bij hem terecht, waardoor hij mij weer verder op weg hielp. Ook heb ik zijn begrip erg op prijs gesteld, aangezien ik de nodige kennis weer moest ophalen, na er een tijd lang tussenuit geweest te zijn. Naarmate de tijd vorderde kreeg mijn begeleider steeds meer vertrouwen erin dat het wel goed zou komen met het verslag. Dat hij dit uitte heb ik als zeer prettig ervaren, en gaf mezelf ook wat meer vertrouwen daarin.

Al met al kijk ik nu terug op een leuk en geslaagd onderzoek, met een verslag waar ik eigenlijk best wel trots op ben. De omstandigheden waren dan wel niet altijd even ideaal, maar de steun om dit voor elkaar te krijgen was groot en heb ik erg gewaardeerd.

Energiezuinige verdeling van taken over verschillende processoren in een 'embedded' computersysteem

R. Arlonne Meenhuis

3 juli 2020

Samenvatting

Computers doen bijna niets anders dan opdrachten verwerken, en het liefst zo snel mogelijk. Om dit te kunnen doen heeft een computer vaak meerdere 'processing units'. Dit maakt dat er keuze is welke 'processing unit' de opdracht uit zal gaan voeren. Deze keuze zou je totaal willekeurig kunnen doen, maar dit is meestal niet de beste optie. Door hiervoor een bepaald beleid op te stellen, zou je deze toewijzing kunnen verbeteren. Zo zou je de snelheid kunnen verbeteren, of bijvoorbeeld een meer recentere issue, het energieverbruik. In dit onderzoek is er gekeken naar de optimalisatie van dit soort toewijzingen bij 2 (verschillende) 'processing units', welke gemodelleerd zijn als een 'processor sharing'-systeem. Qua optimalisatie is er zowel gekeken naar het optimaliseren van de gemiddelde behandelduur, als naar het optimaliseren van het energieverbruik. Daarnaast wordt de 'performance per Watt' gebruikt, om de gemiddelde behandelduur (performance) te combineren met het energieverbruik. Hiervan zijn wiskundige modellen opgesteld, en simulatiemodellen gemaakt in Python. De uitkomsten van de wiskundige modellen zijn daarbij vergeleken met de uitkomsten van de simulatiemodellen. Dit heeft tot bijzondere resultaten geleid, die uiteindelijk erg bruikbaar bleken te zijn.

Keywords: Optimalisatie; Energiezuinigheid; Performance per Watt; Behandelduur; Verdeling van opdrachten; Processing Unit; CPU; GPU; Processor sharing; Bernoulli policy; One-step policy improvement; Wiskundig model; Simulatiemodellen

1 Inleiding

Computers krijgen vaak veel opdrachten te verwerken, en dat kost zowel tijd als energie. Aan de ene kant is het prettig als opdrachten snel worden afgerond, maar aan de andere kant zou dit kunnen leiden tot een zeer ongunstig energieverbruik. Om dit proces te optimaliseren wordt er dus gezocht naar een goede afweging tussen behandelduur en energieverbruik.

Voor het verwerken van opdrachten heeft een computer vaak meerdere 'Processing Units', welke globaal onder te verdelen zijn in 2 groepen: CPU's (Central Processing Unit) en GPU's (Graphics Processing Unit). Sommige opdrachten kunnen alleen door CPU's verwerkt worden, sommige anderen alleen door GPU's. Bij deze 2 type opdrachten is het vrij duidelijk naar welk type 'processing unit' ze gestuurd worden. Daarnaast bestaan er opdrachten die door beide types verwerkt kunnen worden, maar waarbij de behandelduur en het energieverbruik verschilt. Deze laatste groep opdrachten is dus voor dit onderzoek het meest interessant, aangezien hier de vrijheid zit om ze zo optimaal mogelijk over beide type 'processing units' te verdelen.

1.1 Literatuuronderzoek

Er is op het gebied van 'parallel computing' (meerdere taken tegelijkertijd uitvoeren op verschillende machines) al aardig wat onderzoek gedaan. De aandacht in dit onderzoek zal voornamelijk liggen bij literatuur waarin zowel CPU's als GPU's een rol spelen en waarin de gemiddelde behandelduur centraal staat. Ook is de energiezuinigheid van deze processen onderzocht. Alleen was hier helaas weinig over te vinden. Slechts in één artikel^[8] stond er iets in over energiezuinigheid, namelijk dat de 'performance per Watt' daarbij een belangrijke maatgever zou zijn.

In de meeste onderzoeken wordt MapReduce^[3] gebruikt. Dit is een programma die de taken opsplijst en vervolgens verdeeld (Map) over de beschikbare 'processoren'. De desbetreffende 'processoren' gaan deze taak dan uitvoeren (Reduce). Het programma pretendeert zeer eenvoudig te zijn, zodat mensen die geen ervaring hebben met 'parallel' en 'distributed computing' er ook goed mee kunnen werken. In zo'n systeem is er één machine, de zogenoemde 'Master', die de overige machines aanstuurt. De andere machines worden 'Werkers' genoemd, deze voeren de taken uit. Mocht één van deze werkers het begeven, dan worden zijn taken overgenomen door de andere werkers. In het geval dat de Master ermee op houdt, wordt dit niet opgevangen. Maar dat is iets wat slechts zeer zelden schijnt voor te komen.

De meeste onderzoeken zijn erop gericht om de 'processing time' (de tijd die nodig is om alle taken uit te voeren) zo kort mogelijk te maken.^[2,6,8,11,12] Bij een deel van deze onderzoeken^[2,11] wordt er gewerkt met een verhoudingsgetal α , die het verschil in behandelingsduur weergeeft van de desbetreffende taak. Hierbij wordt er dus vanuit gegaan dat je voor alle taken de behandelingsduur kent bij elk type 'processing unit', of deze aan de hand van kenmerken goed kunt benaderen.

Uit een aantal onderzoeken^[6,8,11] blijkt verder dat door gebruik te maken van zowel CPU's als GPU's de 'processing time' tot maximaal 2 maal sneller kan gaan, dan bij gebruik van alleen maar CPU's (maar wel met evenveel 'processing units', anders is het antwoord wel heel triviaal dat het sneller kan). Dat deze winst variabel is komt door het verschil in taken die uitgevoerd moeten worden. Zo is er bij recursieve processen minder winst te behalen, dan bij processen die allemaal op zichzelf staan. Daarnaast zijn sommige taken meer geschikt voor CPU's en andere weer meer geschikt voor GPU's, dit komt door een verschil in de bouwstructuur.

Ook zijn er nog wat artikelen^[1,7,9] onderzocht, die wat meer wiskundig van aard zijn. Één artikel^[1] daarvan gaat volledig over verschillende soorten multiserver en single-server queues en welke formules je daarbij het best kunt gebruiken. De kostenfuncties worden daarbij opgesplitst voor 3 verschillende situaties, namelijk voor wachten, behouden en verwerpen. Al met al is het dus een goed artikel. De andere twee artikelen^[7,9] gaan over een optimale toewijzing van 'query' aankomsten, met daarnaast veroudering van afgeronde 'report' opdrachten. Dit zijn zeer mooie en interessante artikelen, welke dan ook worden gebruikt om in dit verslag een model op te stellen.

Een belangrijk aandachtspunt, dat verder nog wordt genoemd, is het verschil in taal die een CPU gebruikt en welke een GPU gebruikt.^[8,11] Dit zou mogelijk van invloed kunnen zijn of een GPU de informatie wel zou kunnen verwerken of niet. Uit één artikel^[8] blijkt dat er wel steeds betere 'converters' (programma's die de informatie van de ene taal omzetten naar een andere taal) beschikbaar zijn. Daarnaast kost deze 'vertaling' tijd, iets waar ze in meerdere artikelen^[2,6,11] rekening mee houden. Daarnaast wordt geadviseerd^[8] om de algoritmes bij constructie al vriendelijk te maken voor beide type 'processing units'. En om deze units flexibeler te maken, zodat de delen die niet gebruikt worden tezamen ook een taak gelijktijdig kunnen uitvoeren.

Daarnaast wordt er nog in een ander onderzoek^[12] naast CPU's en GPU's, ook 'Cloud Computing' meegenomen. Hierbij worden er taken via een netwerk verstuurd naar 'Cloud Computers', die een deel van de taken gaan uitvoeren. Hierdoor is het aantal 'processor units' dus onbekend, en wordt optimaliseren van de energiezuinigheid erg complex.

1.2 Onderzoeksvraag

Het doel van dit onderzoek is: Het vinden van een gunstige verdeling van de taken over de 'processing units' zodanig dat de gemiddelde behandelingsduur zo kort mogelijk is, en dat daarbij rekening wordt gehouden met het energieverbruik. Hierdoor luidt de onderzoeksvraag:

Hoe kunnen opdrachten het beste over de 'processing units' worden verdeeld, om tot een zo optimaal mogelijk energieverbruik te komen en waarbij de gemiddelde behandelingsduur nog acceptabel is?

2 Energiezuinigheid

2.1 Controleren vermogen

Om iets te kunnen zeggen over de energiezuinigheid, is het belangrijk om eerst meer te weten te komen over het energieverbruik van een 'processing unit'. In "Power Efficient Distribution and Execution of Tasks upon Hardware Fault with Multiple Processors"^[4] wordt een verband gelegd tussen de processorfrequentie (hierna genoemd als frequentie) en het energieverbruik, namelijk: $E = p_d t$ met $p_d = C_{eff} V_{dd}^2 f$, waarbij E het energieverbruik is in Joule, p_d het vermogen in Watt, t de tijd in seconden, C_{eff} de capaciteit van de processor in Farad, V_{dd} het potentiaal van de processor in Volt en f de frequentie van de processor in Hertz. Verderop in dat artikel wordt een aanname gemaakt dat als de 'processor unit' er 2 keer zolang over zou doen om dezelfde taak te voltooien, het vermogen met een factor 8 zou afnemen. Dit zou dan resulteren in een energieverbruik dat 4 keer lager is. In dit (gedachten)voorbeeld halveren ze namelijk de frequentie, en nemen ze daarbij aan dat daardoor ook het potentiaal zal halveren. Er wordt hierbij dus uitgegaan van een recht evenredig verband tussen frequentie en potentiaal, wat mogelijk toch discutabel is. Maar al met al lijkt er dus wel veel winst te behalen qua energiezuinigheid, ook al gaat dit dan wel ten koste van de snelheid waarop taken worden afgerond. Om toch dezelfde snelheid te kunnen behouden waarop taken worden afgerond, zou je alle taken kunnen opsplitsen. De ene helft van de taken kan dan door de ene CPU verwerkt worden en de andere helft door een andere CPU. Alleen vraagt dit wel extra aanschaf- en onderhoudskosten.

In "Enhanced Intel SpeedStep Technology for the Intel Pentium M Processor"^[5] worden precies dezelfde vergelijkingen gebruikt als het vorige artikel^[4]. Daarnaast wordt er ook een tabel weergegeven wat het potentiaal zou zijn van een CPU bij verschillende frequentie's (zie: Tabel 1). In deze tabel valt meteen op, dat het potentiaal niet halveert als de frequentie gehalveerd wordt.

Frequentie	Potentiaal
1.6 GHz	1.484 V
1.4 GHz	1.420 V
1.2 GHz	1.276 V
1.0 GHz	1.164 V
0.8 GHz	1.036 V
0.6 GHz	0.956 V

Tabel 1: De tabel uit "Enhanced Intel SpeedStep Technology for the Intel Pentium M Processor"^[5]

Hoe hoger de frequentie des te harder de CPU aan het werk is. Nu is de vraag wat energie zuiniger is, weinig CPU's die harder werken (op een hoge frequentie) of meer CPU's die langzamer werken (op een lagere frequentie). Om dit te onderzoeken zijn alle frequentiewaarden in Tabel 1 met elkaar vergeleken. Hierbij zijn de waarden zodanig gekozen, dat de totale frequentie in beide gevallen hetzelfde is. De uitwerkingen hiervan staan weergegeven in Tabel 2.

Wat hierin allereerst opvalt is dat het vermogen van wat de CPU's op hogere frequentie gezamenlijk produceren hoger ligt, dan het vermogen wat de CPU's op lagere frequentie gezamenlijk produceren. Het verschil lijkt hierbij groter te zijn, als het verschil in aantal CPU's ook groter is. Hierdoor lijkt het dus qua energieverbruik en voor een snelle verwerking van taken aantrekkelijk om veel 'processing units' te hebben. In hoeverre dit voordeel op weegt tegen de aanschaf- en onderhoudskosten, zou dan nog wel verder onderzocht moeten worden.

Verder valt op dat de factor 4 uit het andere artikel^[4] in deze berekeningen zeer zeker niet behaald worden. Voor een goede vergelijking van hun voorbeeld zou je de volgende 2 paren het best met elkaar kunnen vergelijken: Één CPU op een frequentie van 1.6 GHz met twee CPU's op een frequentie van 0.8 GHz, en één CPU op een frequentie van 1.2 GHz met twee CPU's op een frequentie van 0.6 GHz. Deze geven respectievelijk een factor 2.05 en een factor 1.78. Hierbij wordt het vermogen van de hogere frequenties gedeeld door het vermogen van de lagere.

Hoge Freq.	Lage Freq.	#Hoog	#Laag	Totale Freq.	Vermogen Hoge Freq.	Vermogen Lage Freq.
1.6 GHz	1.4 GHz	7	8	11.2 GHz	24.6653 GW	22.5837 GW
	1.2 GHz	3	4	4.8 GHz	10.5708 GW	7.8152 GW
	1.0 GHz	5	8	8.0 GHz	17.6180 GW	10.8392 GW
	0.8 GHz	1	2	1.6 GHz	3.5236 GW	1.7173 GW
	0.6 GHz	3	8	4.8 GHz	10.5708 GW	4.3869 GW
1.4 GHz	1.2 GHz	6	7	8.4 GHz	16.9378 GW	13.6767 GW
	1.0 GHz	5	7	7.0 GHz	14.1148 GW	9.4843 GW
	0.8 GHz	4	7	5.6 GHz	11.2918 GW	6.0105 GW
	0.6 GHz	3	7	4.2 GHz	8.4689 GW	3.8385 GW
1.2 GHz	1.0 GHz	5	6	6.0 GHz	9.7691 GW	8.1294 GW
	0.8 GHz	4	6	4.8 GHz	7.8152 GW	5.1518 GW
	0.6 GHz	1	2	1.2 GHz	1.9538 GW	1.0967 GW
1.0 GHz	0.8 GHz	4	5	4.0 GHz	5.4196 GW	4.2932 GW
	0.6 GHz	3	5	3.0 GHz	4.0647 GW	2.7418 GW
0.8 GHz	0.6 GHz	3	4	2.4 GHz	2.5759 GW	2.1934 GW

Tabel 2: *Vergelijking tussen weinig CPU's op een hoge frequentie t.o.v. meer CPU's op een lagere frequentie. Hierbij is de totale frequentie in beide gevallen steeds gelijk, waardoor ze een taak gelijktijdig afronden als ze gelijktijdig zouden beginnen.*

Opmerking : De factor 4 in de vorige alinea houdt verband met de factor 8 in de eerste alinea. Het verschil tussen deze factoren komt, doordat bij de factor 4 gecorrigeerd wordt voor behoud van behandelduur. Het aantal 'processing units' wordt dan verdubbeld, en de factor halveert.

2.2 Optimalisatie naar 'performance per Watt'

Bij het optimaliseren van de energiezuinigheid, wil je niet te veel inleveren op de gemiddelde snelheid waarmee een taak afgerond wordt. Om dit te optimaliseren zou je gebruik kunnen maken van de 'performance per Watt', zoals dat omschreven staat in "A Survey on Parallel Computing and its Applications in Data-Parallel Problems Using GPU Architectures"^[8]

Om te kunnen optimaliseren naar 'performance per Watt', is het belangrijk om te weten wat dit precies inhoudt. Het meest optimale is een zo hoog mogelijke 'performance' bij een zo laag mogelijk verbruik ('Watt'). Ofwel de maximale waarde van de 'performance per Watt' is het meest optimaal. Aangezien er gewerkt zal worden met een kostenfunctie (minimale waarde is optimaal), wordt de 'performance per Watt' omgeschreven naar 'Watt per performance'.

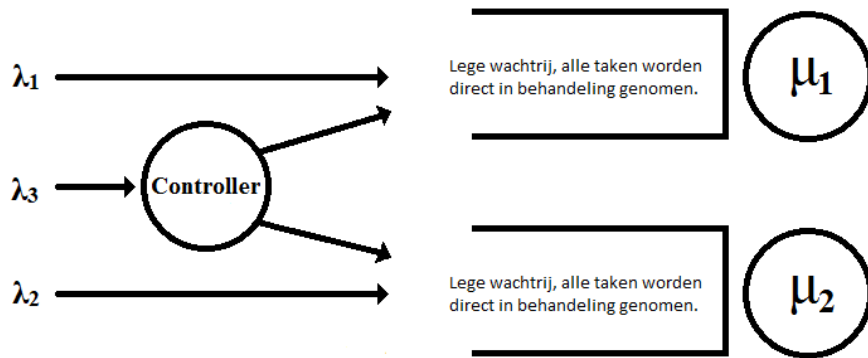
De 'Watt' is het vermogen, wat uitgedrukt wordt in de eenheid Watt. Het optimaliseren van het vermogen geschiedt inderdaad door het te minimaliseren. Qua performance wordt er gekozen voor de gemiddelde behandelduur. Om de gemiddelde behandelduur te optimaliseren is een minimalisatie de beste keuze, terwijl bij de performance juist een maximalisatie optimaal is. Daarom is er gekozen voor: $\text{Performance} = \frac{1}{\text{gemiddelde behandelduur}}$

$$\begin{aligned}
 \max \left(\frac{\text{performance}}{\text{Watt}} \right) &\iff \min \left(\frac{\text{Watt}}{\text{performance}} \right) = \min [\text{gemiddelde behandelduur} \times \text{vermogen}] \\
 &= \min (t P_d) \\
 &= \min (\tilde{E})
 \end{aligned}$$

Hieruit volgt dus dat de optimalisatie van de 'performance per Watt' precies hetzelfde is als de optimalisatie van het energieverbruik. Aangezien de optimalisatie van het energieverbruik al apart wordt behandeld in Paragraaf 3.3.2, wordt er niet meer apart aandacht besteed aan de optimalisatie naar de 'performance per Watt'.

3 Modelbeschrijving

We zijn dus op zoek naar een optimale toewijzing van opdrachten over de beschikbare 'processing units'. Om dit te kunnen doen, heb je allereerst grip nodig op deze toewijzing. Hiervoor heb je dus iets nodig, zoals bijv. een 'controller'. Met de instellingen van zo'n 'controller' wordt het beleid (politiek) vastgelegd, hoe de opdrachten over de 'processing units' worden verdeeld. Maar hoe bepaal je nu welke instelling je het beste kunt gebruiken? Voor dit probleem is het volgende model opgesteld, welke weergegeven wordt in Figuur 1. In dit model zijn 3 type aankomsten te zien, welke overzichtelijk zijn weergegeven in Tabel 3. Opdrachten van type 1 (λ_1) gaan hierbij altijd naar het eerste wachtrijsysteem en opdrachten van type 2 (λ_2) gaan hierbij altijd naar het tweede wachtrijsysteem. Deze 2 type opdrachten stellen de opdrachten voor die slechts bij één van beide wachtrijsystemen behandeld kunnen worden, dit zijn bijvoorbeeld opdrachten die of alleen door een CPU kunnen worden afgehandeld of alleen door een GPU. Opdrachten van type 3 (λ_3) kunnen wel door beide wachtrijsystemen afgehandeld worden en zullen hierdoor eerst aankomen bij de 'controller'. Deze 'controller' bepaald dan, voor iedere aankomst bij de 'controller', naar welk wachtrijsysteem deze opdracht gaat.



Figuur 1: Het Model, met daarin een 'controller' en 2 wachtrijsystemen. Voor beide wachtrijsystemen geldt 'processor sharing'. Dit houdt in dat elk wachtrijsysteem slechts één server heeft, en elke opdracht die daarbij arriveert direct in behandeling wordt genomen. Dit resulteert in lege wachtrijen en een variabele behandelsnelheid.

Type	Parameter	Uitleg
1	λ_1	Gaan strikt naar server 1.
2	λ_2	Gaan strikt naar server 2.
3	λ_3	Worden door een controller verdeeld over beide servers.

Tabel 3: De verschillende type aankomsten, met de bijbehorende parameter en uitleg.

Ook zijn er in dit model 2 wachtrijsystemen te zien, die beide bestaan uit slechts 1 server. Daarnaast is te zien, dat voor beide wachtrijsystemen, de wachtrij leeg is en dat alle taken/opdrachten direct in behandeling worden genomen. Beide wachtrijsystemen zijn namelijk gemodelleerd als een 'processor sharing'-systeem, wat inhoudt dat elke opdracht direct in behandeling wordt genomen en de aandacht van de server gelijkmatig verdeeld is over alle aanwezige opdrachten. Zodra er dus meerdere opdrachten bij één server aanwezig zijn, dan is dit dus van invloed op de behandelduur van elke opdracht afzonderlijk bij deze server.

Verder wordt er aangenomen dat alle opdrachten die uitgevoerd moeten worden onafhankelijk van elkaar zijn, de tussenaankomsttijden (X_k) exponentieel verdeeld zijn met parameter λ_k ($X_k \sim \text{Exp}(\lambda_k)$, voor $k = 1, 2, 3$), en dat de bedieningsduren (Y_k) ook exponentieel verdeeld zijn met parameter μ_k ($Y_k \sim \text{Exp}(\mu_k)$, voor $k = 1, 2$). Hierbij is Y_1 de stochast die de bedieningsduur

bij de eerste server definieert in het geval dat er geen andere opdrachten bij deze server aanwezig zouden zijn. En Y_2 is de stochast die de bedieningsduur bij de tweede server definieert. Dit is net als bij de eerste server, alleen de bedieningsduur in het geval dat er geen andere opdrachten bij de tweede server aanwezig zouden zijn.

Daarnaast stelt elke server in dit geval een 'processing unit' voor. Dit kan een CPU en een GPU zijn, maar ook 2 dezelfde 'processing units', die allebei op een verschillende snelheid werken. Daarnaast kan dit model ook gebruikt worden om twee clusters van 'processing units' met elkaar te vergelijken. Hierbij moet iedere 'processing unit', binnen hetzelfde cluster, dan wel van hetzelfde type zijn. Ook moeten ze even hard werken, wat inhoudt dat de opdrachten gelijkmatig over de 'units' worden verdeeld en dat ze op dezelfde snelheid (frequentie) ingesteld staan. Verder moet gelden dat het aantal servers (s_k) per cluster eindig is, ofwel $s_k < \infty$, voor $k = 1, 2$. Elk cluster 'processing units' wordt hierbij dan gemodelleerd als zijnde één server met 'processor sharing'. De parameter μ_k ($k = 1, 2$) wordt in dit geval dan het aantal servers in het k^e cluster (s_k), vermenigvuldigd met de parameter (μ) van één server binnen dit cluster. Aangezien elke server binnen een cluster identiek wordt verondersteld, is er dus maar 1 waarde voor μ_k mogelijk.

De toestandsruimte S bestaat uit de verzameling $\{i, j\}$, waarvoor geldt dat: $0 \leq i < \infty$ en $0 \leq j < \infty$. Hierbij staat i voor het aantal opdrachten die op dat moment door server 1 verwerkt worden, en j voor het aantal opdrachten die op dat moment door server 2 verwerkt worden. Er zit dus geen limiet aan het aantal opdrachten bij beide servers.

3.1 Gemiddelde behandelduur bij alleen aankomsten van type 3

Om niet te complex te beginnen wordt er nu eerst gekeken naar alleen aankomsten van type 3, ofwel $\{X_1\} = \emptyset$, $\{X_2\} = \emptyset$ en $\lambda_3 > 0$. Daarnaast moeten beide servers de vraag aankunnen, waardoor moet gelden dat: $\lambda_3 \alpha < \mu_1$ en $\lambda_3 (1 - \alpha) < \mu_2$. Aangezien elke waarde van α in dit verslag onderzocht moet kunnen worden, wordt de strengere eis gehanteerd, dat: $\lambda_3 < \mu_1$ en $\lambda_3 < \mu_2$. Door de geheugenloosheid van de exponentiële verdeling kan dit model gezien worden als een Markov Beslissingsprobleem. Aan de hand van "Optimal Query Assignment for Wireless Sensor Networks" [7] wordt er een wiskundig model opgesteld.

3.1.1 Het CTMDP-model

Het Continue Tijd Markov BeslissingsProces(CTMDP)-model komt er als volgt uit te zien:

- Toestandsruimte: $S = \mathbb{N}_0 \times \mathbb{N}_0$, waar $(i, j) \in S$ beschrijft de toestand waarbij er i opdrachten zijn bij de eerste server en er j opdrachten zijn bij de tweede server.
- Actieruimte: Bij een nieuwe taak van type 3 bestaan er de volgende twee actiemogelijkheden d in de actieruimte $A = \{Q_1, Q_2\}$, waarbij $d = Q_1$ de taak wordt verwezen naar de eerste server en bij $d = Q_2$ de taak wordt verwezen naar de tweede server.
- De transitieverhoudingen, wanneer het systeem zich in toestand $(i, j) \in S$ bevindt en actie $d \in A$ wordt gekozen, zien er als volgt uit:

$$\mathbf{q}^d[(i, j), (i', j')] = \begin{cases} \lambda_3, & (i', j') = (i + 1, j), d = Q_1 \\ \lambda_3, & (i', j') = (i, j + 1), d = Q_2 \\ \mu_1, & (i', j') = (i - 1, j), i > 0 \\ \mu_2, & (i', j') = (i, j - 1), j > 0 \\ 0, & \text{anders} \end{cases} \quad (1)$$

Deze transitieverhoudingen ($\mathbf{q}^d$) geeft alle mogelijke transities weer, die je vanuit toestand (i, j) kunt maken. De nieuwe toestand (i', j') is bereikbaar onder de voorwaarden die achteraan staan.

Het voorste getal geeft een waarde aan, die de waarschijnlijkheid van deze transitie weergeeft. Dit zijn dus de verhoudingen tussen de verschillende transities die kunnen plaatsvinden. Bij de onderste regel is dit een 0, waardoor alle andere toestanden, die niet expliciet daarboven beschreven staan, onbereikbaar zijn vanuit toestand (i, j) . In dit geval geldt dus dat voor alle toestanden (i, j) , alleen de naburige toestanden bereikbaar zijn. De eerste regel geeft een aankomst weer die naar de eerste server wordt gestuurd, de tweede regel is een aankomst die naar de tweede server wordt gestuurd. De derde regel is een vertrek bij de eerste server, en de vierde regel is een vertrek bij de tweede server. De laatste regel bevat alle andere mogelijke toestandstransities, welke onbereikbaar zijn vanwege de 0. Verder geldt er, dat als er meer opdrachten bij een server aanwezig zijn, dat dit van invloed is op de snelheid dat een opdracht afgerond wordt. Dit is niet terug te zien in de transitieverhoudingen, aangezien de kans dat een opdracht afgerond wordt vergroot is met precies diezelfde factor. Hierdoor heft dit elkaar op. Ter verduidelijking:

$$i \cdot \frac{\mu_1}{i} = \mu_1 \quad \& \quad j \cdot \frac{\mu_2}{j} = \mu_2$$

Hierin stelt het delen van μ_1 en μ_2 , door respectievelijk i en j , het 'processor sharing' aspect voor. Naarmate er namelijk meer opdrachten bij een server aanwezig zijn, zal de snelheid waarmee een individuele opdracht (bij die server) afgerond wordt afnemen. De vermenigvuldiging, met respectievelijk i en j , stelt de verhoogde kans voor. Als er namelijk meer opdrachten bij een server aanwezig zijn, dan vergroot dit de kans dat er een opdracht (bij die server) afgerond wordt.

Dit CTMDP-model heeft steeds verschillende tijden tussen twee opeenvolgende gebeurtenissen. Aangezien dit de uitwerking van het model uiterst complex maakt, worden de tijden tussen twee opeenvolgende gebeurtenissen geüniformeerd. De tijd wordt hierbij in hele kleine uniforme stukjes opgedeeld, zodanig dat er maximaal maar 1 gebeurtenis van het CTMDP-model, per tijdsinterval van het 'nieuwe' model, plaatsvindt. Dit 'nieuwe' geüniformeerde model kan gezien worden als een Discrete Tijd en Discrete Ruimte Markov BeslissingsProbleem (DTMDP). Voor het bewijs dat het CTMDP-model en het DTMDP-model stochastisch equivalent zijn, verwijs ik naar Paragraaf 2.2 en Appendix A van "Optimal Query Assignment for Wireless Sensor Networks"! [7]

3.1.2 Het DTMDP-model

De toestands- en actieruimte van het Discrete Tijd en Discrete Ruimte Markov BeslissingsProbleem (DTMDP) zijn hetzelfde als die van het CTMDP-model. De transitieverhoudingen veranderen wel, aangezien er nu stapjes tussen zitten waarbij de toestand niet verandert. In het CTMDP-model is er namelijk bij elke stap een nieuwe aankomst geweest of er is een taak afgerond, maar in het DTMDP-model zijn er ook stapjes waarin geen van deze gebeurtenissen plaatsvindt. Je blijft dus daarbij in dezelfde toestand. Het DTMDP-model ziet er als volgt uit:

- Toestandsruimte: $S = \mathbb{N}_0 \times \mathbb{N}_0$, waar $(i, j) \in S$ beschrijft de toestand waarbij er i opdrachten zijn bij de eerste server en er j opdrachten zijn bij de tweede server.
- Actieruimte: Bij een nieuwe taak van type 3 bestaan er de volgende twee actiemogelijkheden d in de actieruimte $A = \{Q_1, Q_2\}$, waarbij $d = Q_1$ de taak wordt verwezen naar de eerste server en bij $d = Q_2$ de taak wordt verwezen naar de tweede server.
- De transitiekansen, wanneer het systeem zich in toestand $(i, j) \in S$ bevindt en actie $d \in A$ wordt gekozen, zien er als volgt uit:

$$\mathbf{P}^d[(i, j), (i', j')] = \begin{cases} \lambda_3^*, & (i', j') = (i + 1, j), d = Q_1 \\ \lambda_3^*, & (i', j') = (i, j + 1), d = Q_2 \\ \mu_1^*, & (i', j') = (i - 1, j), i > 0 \\ \mu_2^*, & (i', j') = (i, j - 1), j > 0 \\ 1 - (\lambda_3^* + \mu_1^* \mathbb{1}_{i>0} + \mu_2^* \mathbb{1}_{j>0}), & (i', j') = (i, j) \\ 0, & \text{anders} \end{cases} \quad (2)$$

- Kostenfunctie: De kosten worden bepaald aan de hand van in welke toestand je zit. Deze kosten moet de tijd weergeven (per tijdstapje) die nodig is om opdrachten te voltooien. Hierdoor ziet de kostenfunctie $Cost(i, j)$ er als volgt uit:

$$Cost(i, j) = \gamma_1 i + \gamma_2 j \quad (3)$$

De eerste twee regels van Vergelijking 2 modelleren de nieuwe taken die verwerkt worden bij de verschillende actiemogelijkheden d . De derde en vierde regel modelleren respectievelijk de afronding van een taak bij de eerste server en bij de tweede server. De vijfde regel is een dummy transitie als gevolg van het uniformeren (dit zijn de extra stappen waarbij de toestand niet verandert). En de laatste regel bevat alle andere mogelijke toestandstransities. Verder geldt: $\lambda_3^* = \lambda_3 B^{-1}$ en $\mu_k^* = \mu_k B^{-1}$ (voor $k = 1, 2$), waarbij B een groot eindig getal is. Deze B is het getal, die de oorspronkelijke tijd van het CTMDP-model opdeelt in B uniforme tijdstapjes, zodanig dat er maximaal maar 1 gebeurtenis per tijdstapje plaatsvindt. Hierdoor moet B minimaal voldoen aan de eis dat $B \geq \lambda_3 + \mu_1 + \mu_2$. Verder is het zo dat dit transitiekansen zijn, waardoor alle kansen bij elkaar opgeteld samen 1 moeten zijn en elke afzonderlijke kans tussen 0 en 1 zou moeten liggen. Hieruit volgt dus dat: $0 \leq \lambda_3^*, \mu_1^*, \mu_2^* \leq 1$ en $\lambda_3^* + \mu_1^* + \mu_2^* \leq 1$. Als parameters van exponentiële verdelingen zijn λ_3, μ_1 en μ_2 al groter dan 0. Dit verandert niet als ze door een groot eindig getal B gedeeld worden.

Bij de kostenfunctie is het zo, dat er i opdrachten aanwezig bij de eerste server, die elk γ_1 aan kosten hebben per tijdstapje en bij de tweede server zijn er j opdrachten, die elk γ_2 aan kosten hebben per tijdstapje. Dit betekent dat, na het voltooien van een opdracht bij de eerste server, het systeem γ_1 aan kosten heeft betaald voor elk tijdstapje dat de opdracht aanwezig was bij deze server. Ofwel de gemaakte kosten houden verband met de behandelduur van elke opdracht bij de eerste server. Voor voltooide opdrachten bij de tweede server geldt iets gelijks, maar dan met γ_2 i.p.v. γ_1 . Aangezien in deze paragraaf de gemiddelde behandelduur geoptimaliseerd wordt, is het zeer aannemelijk dat in dit geval γ_1 en γ_2 gelijk aan elkaar zijn. Anders zou namelijk het behandelen door de ene server zwaarder wegen dan het behandelen door de andere server. Iets wat zeer onlogisch is aangezien 'tijd' iets is wat bij beide servers hetzelfde is.

Opmerking : Je kunt beide servers ook een verschillend gewicht meegeven, om bijvoorbeeld de behandelduur bij de ene server zwaarder te laten wegen dan bij de andere server. In die gevallen zou dan $\gamma_1 \neq \gamma_2$. Zoals verderop te zien is bij de optimalisatie van het energieverbruik.

Aanname : Voor de optimalisatie van de gemiddelde behandelduur, wordt aangenomen dat de behandelduur bij de ene server even zwaar weegt als bij de andere server. Het voelt namelijk niet logisch om de behandelduur bij de ene server zwaarder te tellen dan bij de andere, aangezien je daarmee niet de gemiddelde behandelduur optimaliseert. De duur van elk tijdstapje is namelijk voor beide servers even groot. En het verschil in snelheid zit al verwerkt in de μ 's.

Opmerking : Als $\gamma_1 = \gamma_2 = 1$, dan wordt deze kostenfunctie: $Cost(i, j) = i + j$. Dit is precies het aantal opdrachten dat op dat moment aanwezig is in het systeem.

3.1.3 Standaard Stochastisch Dynamisch Programmeren

Deze $(S, A, P, Cost)$ zoals hierboven geïntroduceerd, beschrijven volledig de Discrete Tijd en Discrete Toestand Markov BeslissingsProbleem. Om een optimale toewijzingspolitiek te bepalen en om gebruik te maken van standaard dynamisch programmeren, wordt de volgende waardefunctie gedefinieerd:

$$V_n(i, j) := \begin{cases} \text{De verwachte minimale kosten na } n \text{ stappen,} \\ \text{gegeven dat het proces is begonnen in toestand } (i, j). \end{cases}$$

Deze $\mathbf{V}_n(i, j)$ wordt recursief berekend door gebruik te maken van het zo geheten ‘means of the value’ iteratie, en gaat als volgt. Allereerst wordt er gekeken naar $\mathbf{V}_0(i, j)$. Hierbij zijn er nog geen stappen gemaakt, en dus ook nog geen kosten. Wat leidt tot: $\mathbf{V}_0(i, j) = 0$. Vervolgens wordt, middels het waarde iteratie algoritme en de volgende achterwaardse recursieve vergelijking, geïtereerd:

$$\mathbf{V}_{n+1}(i, j) = \begin{cases} \gamma_1 i + \gamma_2 j + \lambda_3^* \min \begin{cases} \mathbf{V}_n(i+1, j) \\ \mathbf{V}_n(i, j+1) \end{cases} \\ + \mu_1^* \mathbf{V}_n(i-1, j) \mathbb{1}_{i>0} \\ + \mu_2^* \mathbf{V}_n(i, j-1) \mathbb{1}_{j>0} \\ + [1 - (\lambda_3^* + \mu_1^* \mathbb{1}_{i>0} + \mu_2^* \mathbb{1}_{j>0})] \mathbf{V}_n(i, j) \end{cases} \quad (4)$$

Vergelijking (8.4.2) uit het boek "Markov Decision Processes: Discrete Stochastic Dynamic Programming";^[10] stelt het volgende.

$$\mathbf{g} + \mathbf{V}(s) = \min_{d \in A} \{ \text{Cost}(s, d) + \sum_{k \in S} p(k|s, d) \mathbf{V}(k) \}$$

Hierin is $\mathbf{g}$ de tijd-gemiddelde kosten, $\mathbf{V}(s)$ de relatieve waardefunctie (het asymptotisch verschil in kosten als je begint in toestand s t.o.v. een referentietoestand $\bar{s}$), s de toestand waarin je zit, $\text{Cost}(s, d)$ de kosten als je in toestand s zit en actie d uitgevoerd wordt en $p(k|s, d)$ is de kans dat als je in toestand s bent en actie d uitgevoerd wordt dat je naar toestand k gaat.

Opmerking : De $\mathbf{V}(s)$ in deze vergelijking is niet hetzelfde als de $\mathbf{V}_n(i, j)$ in Vergelijking (4).

Daarnaast is er in dit verslag gekozen voor een minimalisatie, in plaats van een maximalisatie, aangezien de opbrengsten in dit geval kosten zijn. Ook is de kostenfunctie onafhankelijk t.o.v. de gekozen actie d , waardoor geldt dat $\forall d \in A: \text{Cost}(s, d) = \text{Cost}(s) = \text{Cost}(i, j) = \gamma_1 i + \gamma_2 j$. Als dit wordt toegepast op het DTMDP-model, dan leidt dit tot de volgende vergelijking.

$$\mathbf{g} + \mathbf{V}(i, j) = \begin{cases} \gamma_1 i + \gamma_2 j + \lambda_3^* \min \begin{cases} \mathbf{V}(i+1, j) \\ \mathbf{V}(i, j+1) \end{cases} \\ + \mu_1^* \mathbf{V}(i-1, j) \mathbb{1}_{i>0} \\ + \mu_2^* \mathbf{V}(i, j-1) \mathbb{1}_{j>0} \\ + [1 - (\lambda_3^* + \mu_1^* \mathbb{1}_{i>0} + \mu_2^* \mathbb{1}_{j>0})] \mathbf{V}(i, j) \end{cases} \quad (5)$$

De $\mathbf{g}$ en $\mathbf{V}(i, j)$ in Vergelijking (5) staan in relatie tot de $\mathbf{V}_{n+1}(i, j)$ (en de $\mathbf{V}_n(i, j)$) in Vergelijking (4), middels de volgende vergelijking:

$$\mathbf{V}_{n+1}(i, j) = (n+1) \mathbf{g} + \mathbf{V}(i, j)$$

3.1.4 Bernoulli policy

De standaard politiek die onderzocht zal worden is de Bernoulli policy. Hierbij wordt een taak met kans α naar de eerste server gestuurd, en met een kans van $(1 - \alpha)$ naar de tweede. Voor het uitwerken van deze politiek wordt gebruik gemaakt van het artikel: "On the Control of a Queueing System with Aging State Information"^[9] Hierdoor ontstaat de volgende vergelijking:

$$\mathbf{g}^\alpha + \mathbf{V}^\alpha(i, j) = \begin{cases} \gamma_1 i + \gamma_2 j + \lambda_3^* [\alpha \mathbf{V}^\alpha(i+1, j) + (1 - \alpha) \mathbf{V}^\alpha(i, j+1)] \\ + \mu_1^* \mathbf{V}^\alpha(i-1, j) \mathbb{1}_{i>0} \\ + \mu_2^* \mathbf{V}^\alpha(i, j-1) \mathbb{1}_{j>0} \\ + [1 - (\lambda_3^* + \mu_1^* \mathbb{1}_{i>0} + \mu_2^* \mathbb{1}_{j>0})] \mathbf{V}^\alpha(i, j) \end{cases} \quad (6)$$

Aangezien $\mathbf{g}^\alpha$ de tijd gemiddelde kosten zijn, zal deze er zeer waarschijnlijk als volgt uitzien.

$$\mathbf{g}^\alpha = \alpha [\text{gemiddelde kosten eerste server}] + (1 - \alpha) [\text{gemiddelde kosten tweede server}] \quad (7)$$

De totale gemiddelde kosten kun je opsplitsen in een deel dat naar de eerste server gaat, en in een deel dat naar de tweede server gaat. De opdrachten die naar de eerste server gaan, zullen gemiddelde kosten hebben die bij de eerste server hoort. En de opdrachten die naar de tweede server gaan, zullen gemiddelde kosten hebben die bij de tweede server hoort. Aangezien α het deel is dat naar de eerste server gaat, en $(1 - \alpha)$ het deel is dat naar de tweede server gaat, is het zeer aannemelijk dat het vermoeden, zoals omschreven in Vergelijking (7), juist is. Dit komt mede omdat een gefilterd poissonproces op zichzelf ook weer een poissonproces is.

3.1.5 Waarden bepalen van $\mathbf{V}^\alpha(i, j)$ en $\mathbf{g}^\alpha$

Door zorgvuldig de stappen in Appendix A te doorlopen, kan er een oplossing gevonden worden voor $\mathbf{V}^\alpha(i, j)$ en $\mathbf{g}^\alpha$ zodanig dat Vergelijking (6) klopt. Dat dit daadwerkelijk ook een oplossing is, zal blijken uit het bewijs van Stelling 3.1.

Stelling 3.1. De volgende $\mathbf{V}^\alpha(i, j)$ en $\mathbf{g}^\alpha$, zijn een oplossing voor Vergelijking (6). Namelijk:

$$\mathbf{V}^\alpha(i, j) = \frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \frac{i(i+1)}{2} + \frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \frac{j(j+1)}{2}$$

$$\mathbf{g}^\alpha = \lambda_3^* \alpha \left[\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] + \lambda_3^* (1 - \alpha) \left[\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right]$$

Bewijs : Door de gevonden $\mathbf{V}^\alpha(i, j)$ en $\mathbf{g}^\alpha$ in te vullen in Vergelijking (6), krijg je:

$$2 \left(\frac{\lambda_3^* \alpha \gamma_1}{\mu_1^* - \lambda_3^* \alpha} + \frac{\lambda_3^* (1 - \alpha) \gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right) + \frac{\gamma_1 i(i+1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\gamma_2 j(j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} = \begin{cases} 2(\gamma_1 i + \gamma_2 j) \\ + \lambda_3^* \alpha \left[\frac{\gamma_1 (i+2)(i+1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\gamma_2 j(j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \\ + \lambda_3^* (1 - \alpha) \left[\frac{\gamma_1 i(i+1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\gamma_2 (j+2)(j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \\ + \mu_1^* \left[\frac{\gamma_1 i(i-1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\gamma_2 j(j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \mathbb{1}_{i>0} \\ + \mu_2^* \left[\frac{\gamma_1 i(i+1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\gamma_2 j(j-1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \mathbb{1}_{j>0} \\ + [1 - \lambda_3^*] \left[\frac{\gamma_1 i(i+1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\gamma_2 j(j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \\ - \mu_1^* \mathbb{1}_{i>0} \left[\frac{\gamma_1 i(i+1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\gamma_2 j(j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \\ - \mu_2^* \mathbb{1}_{j>0} \left[\frac{\gamma_1 i(i+1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\gamma_2 j(j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \end{cases}$$

Dit kun je herschrijven tot:

$$2 \left(\frac{\lambda_3^* \alpha \gamma_1}{\mu_1^* - \lambda_3^* \alpha} + \frac{\lambda_3^* (1 - \alpha) \gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right) = \begin{cases} 2(\gamma_1 i + \gamma_2 j) + \lambda_3^* \alpha \left[\frac{\gamma_1 (i+2)(i+1)}{\mu_1^* - \lambda_3^* \alpha} - \frac{\gamma_1 i(i+1)}{\mu_1^* - \lambda_3^* \alpha} \right] \\ + \lambda_3^* (1 - \alpha) \left[\frac{\gamma_2 (j+2)(j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} - \frac{\gamma_2 j(j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \\ + \mu_1^* \left[\frac{\gamma_1 i(i-1)}{\mu_1^* - \lambda_3^* \alpha} - \frac{\gamma_1 i(i+1)}{\mu_1^* - \lambda_3^* \alpha} \right] \mathbb{1}_{i>0} \\ + \mu_2^* \left[\frac{\gamma_2 j(j-1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} - \frac{\gamma_2 j(j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \mathbb{1}_{j>0} \end{cases}$$

Ofwel:

$$2 \left(\frac{\lambda_3^* \alpha \gamma_1}{\mu_1^* - \lambda_3^* \alpha} + \frac{\lambda_3^* (1 - \alpha) \gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right) = \begin{cases} 2 (\gamma_1 i + \gamma_2 j) + \lambda_3^* \alpha \left[\frac{2 \gamma_1 (i+1)}{\mu_1^* - \lambda_3^* \alpha} \right] \\ + \lambda_3^* (1 - \alpha) \left[\frac{2 \gamma_2 (j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \\ + \mu_1^* \left[\frac{-2 \gamma_1 i}{\mu_1^* - \lambda_3^* \alpha} \right] \mathbb{1}_{i>0} + \mu_2^* \left[\frac{-2 \gamma_2 j}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \mathbb{1}_{j>0} \end{cases}$$

Aangezien er twee stapfuncties in deze vergelijking zitten, zal elke mogelijke situatie afzonderlijk bekeken worden. Te beginnen bij $(i, j) = (0, 0)$, dan krijg je:

$$\frac{\lambda_3^* \alpha \gamma_1}{\mu_1^* - \lambda_3^* \alpha} + \frac{\lambda_3^* (1 - \alpha) \gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} = \lambda_3^* \alpha \left[\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] + \lambda_3^* (1 - \alpha) \left[\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right]$$

Als $(i, j) = (i, 0)$ met $i > 0$, dan is dit:

$$\begin{aligned} \frac{\lambda_3^* \alpha \gamma_1}{\mu_1^* - \lambda_3^* \alpha} + \frac{\lambda_3^* (1 - \alpha) \gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} &= \begin{cases} \gamma_1 i + \lambda_3^* \alpha \left[\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] \\ + \lambda_3^* (1 - \alpha) \left[\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \\ - [\mu_1^* - \lambda_3^* \alpha] \left[\frac{\gamma_1 i}{\mu_1^* - \lambda_3^* \alpha} \right] \end{cases} \\ &= \lambda_3^* \alpha \left[\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] + \lambda_3^* (1 - \alpha) \left[\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \end{aligned}$$

Als $(i, j) = (0, j)$ met $j > 0$, dan wordt dit:

$$\begin{aligned} \frac{\lambda_3^* \alpha \gamma_1}{\mu_1^* - \lambda_3^* \alpha} + \frac{\lambda_3^* (1 - \alpha) \gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} &= \begin{cases} \gamma_2 j + \lambda_3^* \alpha \left[\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] \\ + \lambda_3^* (1 - \alpha) \left[\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \\ - [\mu_2^* - \lambda_3^* (1 - \alpha)] \left[\frac{\gamma_2 j}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \end{cases} \\ &= \lambda_3^* \alpha \left[\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] + \lambda_3^* (1 - \alpha) \left[\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \end{aligned}$$

En als $(i, j) = (i, j)$ met $i > 0$ en $j > 0$, dan krijg je:

$$\begin{aligned} \frac{\lambda_3^* \alpha \gamma_1}{\mu_1^* - \lambda_3^* \alpha} + \frac{\lambda_3^* (1 - \alpha) \gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} &= \begin{cases} \gamma_1 i + \gamma_2 j + \lambda_3^* \alpha \left[\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] \\ + \lambda_3^* (1 - \alpha) \left[\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \\ - [\mu_1^* - \lambda_3^* \alpha] \left[\frac{\gamma_1 i}{\mu_1^* - \lambda_3^* \alpha} \right] \\ - [\mu_2^* - \lambda_3^* (1 - \alpha)] \left[\frac{\gamma_2 j}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \end{cases} \\ &= \lambda_3^* \alpha \left[\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] + \lambda_3^* (1 - \alpha) \left[\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \end{aligned}$$

De gevonden oplossing is dus inderdaad een oplossing voor alle situaties, dus Stelling 3.1 klopt.

Q.E.D.

3.1.6 Waarde bepalen van γ_1 en γ_2

In het geval dat $\gamma_1 = \gamma_2 = 1$, dan zou de kostenfunctie er als volgt uitzien.

$$\text{Cost}(i, j) = i + j$$

Hierbij zijn de kosten dus het aantal opdrachten die in het systeem aanwezig zijn. De bijbehorende $\mathbf{g}^\alpha$, die voor nu even genoteerd wordt als $\mathbf{g}_1^\alpha$, is dan het gemiddelde aantal opdrachten dat aanwezig is in het systeem. Deze ziet er dan als volgt uit.

$$\mathbf{g}_1^\alpha = \lambda_3^* \alpha \left[\frac{1}{\mu_1^* - \lambda_3^* \alpha} \right] + \lambda_3^* (1 - \alpha) \left[\frac{1}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \quad (8)$$

Middels de formule van Little kan, aan de hand van het gemiddeld aantal opdrachten die in het systeem aanwezig zijn, de gemiddelde behandelduur van een opdracht bepaald worden. Hierbij is $\lambda = \lambda_3^*$.

$$\mathbf{g}^\alpha = \frac{\mathbf{g}_1^\alpha}{\lambda_3^*} = \alpha \left[\frac{1}{\mu_1^* - \lambda_3^* \alpha} \right] + (1 - \alpha) \left[\frac{1}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right]$$

Dit komt overeen met het vermoeden zoals omschreven in Vergelijking (7). Deze $\mathbf{g}^\alpha$ wordt tevens verkregen als in de oorspronkelijke vergelijking wordt gekozen voor $\gamma_1 = \gamma_2 = \frac{1}{\lambda_3^*}$. Hiermee zijn dus de waarden van γ_1 en γ_2 gevonden, en het vermoeden dat deze gelijk aan elkaar zouden zijn is ook uitgekomen. Door deze uitkomst weer om te schrijven naar de oude variabelen, door $\mu_k = \mu_k^* B$ voor $k = 1, 2$ en $\lambda_3 = \lambda_3^* B$, is de vergelijking niet meer afhankelijk van de willekeurig gekozen B , waardoor deze nu vast ligt. Hierbij is B het willekeurig gekozen grote getal waar het systeem in opgeknapt was om het huidige systeem te krijgen. De $\tilde{\mathbf{g}}^\alpha$ is dan de gemiddelde kosten van het oorspronkelijke systeem, wat in dit geval de gemiddelde tijd is dat een opdracht aanwezig is in het systeem, ofwel de gemiddelde behandelduur ($t(\alpha)$).

$$t(\alpha) = \tilde{\mathbf{g}}^\alpha = \alpha \left[\frac{1}{\mu_1 - \lambda_3 \alpha} \right] + (1 - \alpha) \left[\frac{1}{\mu_2 - \lambda_3 (1 - \alpha)} \right] \quad (9)$$

3.1.7 Optimale waarde van α bepalen

Door Vergelijking (9) te differentiëren, en deze afgeleide gelijk te stellen aan nul, kan een optimale waarde voor α gevonden worden. De uiteindelijk gevonden waarde ($\hat{\alpha}_t$) is dan de optimale waarde van α die de gemiddelde behandelduur, onder de Bernoulli policy, minimaliseert.

$$t'(\alpha) = \frac{\mu_1}{[\mu_1 - \lambda_3 \alpha]^2} - \frac{\mu_2}{[\mu_2 - \lambda_3 (1 - \alpha)]^2}$$

Met behulp van Mathematica is deze afgeleide gelijk gesteld aan nul, met als resultaat:

$$\alpha_{1,2} = \frac{\lambda_3 \mu_1 [\lambda_3 - 2 \mu_2] \pm \sqrt{(\lambda_3)^2 \mu_1 \mu_2 [\mu_1 + \mu_2 - \lambda_3]^2}}{(\lambda_3)^2 [\mu_1 - \mu_2]}$$

Deze gevonden α_1 en α_2 zijn, naast de grenswaarden $\alpha = 0$ en $\alpha = 1$, kandidaten voor $\hat{\alpha}_t$. Daarnaast moet gelden dat $0 \leq \hat{\alpha}_t \leq 1$. Dit samen maakt:

$$\hat{\alpha}_t = \left\{ x \in \{0, 1, \alpha_1 \mathbb{1}_{0 < \alpha_1 < 1}, \alpha_2 \mathbb{1}_{0 < \alpha_2 < 1}\} : t(x) \leq t(y) \text{ , } \forall y \in [0, 1] \right\} \quad (10)$$

3.2 Gemiddelde behandelduur bij aankomsten van type 1, type 2 en type 3

De kostenfunctie voor deze blijft:

$$Cost(i, j) = \gamma_1 i + \gamma_2 j$$

Hierbij kunnen de waarden van γ_1 en γ_2 verschillen met de waarden die gevonden zijn in Paragraaf 3.1. Na toepassing van de Bernoulli policy, ontstaat de volgende vergelijking.

$$\mathbf{g}^\alpha + \mathbf{V}^\alpha(i, j) = \begin{cases} \gamma_1 i + \gamma_2 j + [\lambda_1^* + \lambda_3^* \alpha] \mathbf{V}^\alpha(i+1, j) \\ + [\lambda_2^* + \lambda_3^* (1-\alpha)] \mathbf{V}^\alpha(i, j+1) \\ + \mu_1^* \mathbf{V}^\alpha(i-1, j) \mathbb{1}_{i>0} \\ + \mu_2^* \mathbf{V}^\alpha(i, j-1) \mathbb{1}_{j>0} \\ + [1 - (\lambda_1^* + \lambda_2^* + \lambda_3^* + \mu_1^* \mathbb{1}_{i>0} + \mu_2^* \mathbb{1}_{j>0})] \mathbf{V}^\alpha(i, j) \end{cases} \quad (11)$$

Opmerking : Deze vergelijking lijkt erg op Vergelijking (6), alleen zijn er naast $\lambda_3^* \alpha$ aankomsten bij de eerste server, nu ook alle aankomsten van λ_1^* bij de eerste server. En naast $\lambda_3^* (1-\alpha)$ aankomsten bij de tweede server, gaan nu ook alle aankomsten van λ_2^* naar deze server.

Door zorgvuldig de stappen in Appendix A te doorlopen, wordt de volgende oplossing gevonden.

$$\mathbf{V}^\alpha(i, j) = \frac{1}{2} \left(\frac{\gamma_1 i (i+1)}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{\gamma_2 j (j+1)}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} \right)$$

$$\mathbf{g}^\alpha = \frac{\gamma_1 [\lambda_1^* + \lambda_3^* \alpha]}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{\gamma_2 [\lambda_2^* + \lambda_3^* (1-\alpha)]}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]}$$

Stelling 3.2. Deze gevonden $\mathbf{g}^\alpha$ en $\mathbf{V}^\alpha(i, j)$ (zie hierboven) zijn een oplossing van Vergelijking (11).

Bewijs : Zie Appendix F.

Q.E.D.

Opmerking : Als $\lambda_1 = \lambda_2 = 0$, dan is dit precies hetzelfde als alleen bij type 3 aankomsten.

3.2.1 Waarde bepalen van γ_1 en γ_2

Als er opnieuw gekeken wordt naar $\gamma_1 = \gamma_2 = 1$, dan is ook nu de kostenfunctie het aantal opdrachten die aanwezig zijn in het systeem. De bijbehorende $\mathbf{g}^\alpha$, die opnieuw genoteerd wordt als $\mathbf{g}_1^\alpha$, is dan het gemiddelde aantal opdrachten die aanwezig zijn in het systeem. En is:

$$\mathbf{g}_1^\alpha = \frac{\lambda_1^* + \lambda_3^* \alpha}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{\lambda_2^* + \lambda_3^* (1-\alpha)}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} \quad (12)$$

Middels de formule van Little kan, aan de hand van het gemiddeld aantal opdrachten die in het systeem aanwezig zijn, de gemiddelde behandelduur bepaald worden. Hierbij is $\lambda = \sum_{k=1}^3 \lambda_k^*$.

$$\mathbf{g}^\alpha = \frac{\mathbf{g}_1^\alpha}{\lambda_1^* + \lambda_2^* + \lambda_3^*} = \begin{cases} \frac{\lambda_1^* + \lambda_3^* \alpha}{\lambda_1^* + \lambda_2^* + \lambda_3^*} \left[\frac{1}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} \right] \\ + \frac{\lambda_2^* + \lambda_3^* (1-\alpha)}{\lambda_1^* + \lambda_2^* + \lambda_3^*} \left[\frac{1}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} \right] \end{cases}$$

Dit ziet er herkenbaar uit, want opnieuw is dit het gedeelte dat naar de eerste server gaat vermenigvuldigd met de gemiddelde behandelduur van de eerste server opgeteld bij het gedeelte

dat naar de tweede server gaat vermenigvuldigd met de gemiddelde behandelduur van de tweede server. Deze $\mathbf{g}^\alpha$ wordt tevens verkregen als in de oorspronkelijke vergelijking wordt gekozen voor $\gamma_1 = \gamma_2 = \frac{1}{\lambda_1^* + \lambda_2^* + \lambda_3^*}$. Hiermee zijn dus de waarden van γ_1 en γ_2 gevonden, en het vermoeden dat deze gelijk aan elkaar zouden zijn is ook hier weer uitgekomen. Door deze uitkomst weer om te schrijven naar de oude variabelen, door $\mu_k = \mu_k^* B$ voor $k = 1, 2$ en $\lambda_3 = \lambda_3^* B$, is de vergelijking niet meer afhankelijk van de willekeurig gekozen B , waardoor deze nu vast ligt. Hierbij is B het willekeurig gekozen grote getal waar het systeem in opgeknapt was om het huidige systeem te krijgen. De $\tilde{\mathbf{g}}^\alpha$ is dan de gemiddelde kosten van het oorspronkelijke systeem, wat in dit geval dus de gemiddelde behandelduur ($t(\alpha)$) is.

$$t(\alpha) = \tilde{\mathbf{g}}^\alpha = \frac{1}{\lambda_1 + \lambda_2 + \lambda_3} \left[\frac{\lambda_1 + \lambda_3 \alpha}{\mu_1 - [\lambda_1 + \lambda_3 \alpha]} + \frac{\lambda_2 + \lambda_3 (1 - \alpha)}{\mu_2 - [\lambda_2 + \lambda_3 (1 - \alpha)]} \right] \quad (13)$$

3.2.2 Optimale waarde van α bepalen

Door Vergelijking (13) te differentiëren, en deze afgeleide gelijk te stellen aan nul, kan een optimale waarde voor α gevonden worden. De uiteindelijk gevonden waarde ($\hat{\alpha}_t$) is dan de optimale waarde van α die de gemiddelde behandelduur, onder de Bernoulli policy, minimaliseert.

$$t'(\alpha) = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \left(\frac{\mu_1}{[\mu_1 - [\lambda_1 + \lambda_3 \alpha]]^2} - \frac{\mu_2}{[\mu_2 - [\lambda_2 + \lambda_3 (1 - \alpha)]]^2} \right)$$

Met behulp van Mathematica is deze afgeleide gelijk gesteld aan nul, met als resultaat:

$$\alpha_{1,2} = \frac{\lambda_3 \mu_1 [\lambda_2 + \lambda_3 - 2 \mu_2] + \lambda_1 \lambda_3 \mu_2 \pm \sqrt{(\lambda_3)^2 \mu_1 \mu_2 [\lambda_1 + \lambda_2 + \lambda_3 - \mu_1 - \mu_2]^2}}{(\lambda_3)^2 [\mu_1 - \mu_2]}$$

Deze gevonden α_1 en α_2 zijn, naast de grenswaarden $\alpha = 0$ en $\alpha = 1$, kandidaten voor $\hat{\alpha}_t$. Daarnaast moet gelden dat $0 \leq \hat{\alpha}_t \leq 1$. Dit samen maakt:

$$\hat{\alpha}_t = \left\{ x \in \{0, 1, \alpha_1 \mathbb{1}_{0 < \alpha_1 < 1}, \alpha_2 \mathbb{1}_{0 < \alpha_2 < 1}\} : t(x) \leq t(y) \text{ , } \forall y \in [0, 1] \right\} \quad (14)$$

3.3 Energiezuinigheid

Om naar de energiezuinigheid te kunnen kijken, is het belangrijk om te weten hoe het energieverbruik wordt bepaald. Zoals eerder vermeld in Paragraaf 2.1 is de formule voor energieverbruik $E = p_d t$, met $p_d = C_{eff} V_{dd}^2 f$, waarbij E het energieverbruik in Joule, p_d het vermogen in Watt, t de tijd in seconden, C_{eff} de capaciteit van de processor in Farad, V_{dd} het potentiaal van de processor in Volt en f de frequentie van de processor in Hertz. Aangezien de frequenties die gebruikt worden in GHz zijn, zal deze met een F genoteerd worden om verwarring te voorkomen. Ook het vermogen dat hierdoor in GW is, zal hierdoor genoteerd worden met P_d . En het energieverbruik met $\tilde{E}$, aangezien deze nu in GJ zal zijn.

$$\tilde{E} = p_1 \tilde{E}_1 + p_2 \tilde{E}_2 = p_1 P_{d,1} t_1 + p_2 P_{d,2} t_2$$

In deze vergelijking is $\tilde{E}_k$ ($k = 1, 2$) het energieverbruik in GJ bij het k^e cluster 'processing units', p_k de fractie taken van het geheel die naar het k^e cluster worden gestuurd, $P_{d,k}$ het vermogen in GW die bij het k^e cluster hoort en t_k de gemiddelde tijd in s die een opdracht aanwezig is bij het k^e cluster. De $P_{d,k}$ hangt daarnaast ook af van het aantal 'processing units' binnen het k^e cluster, waardoor $P_{d,k} = s_k C_{eff,k} V_{dd,k}^2 F_k$. Hierbij is s_k het aantal 'processing units' bij het k^e cluster, F_k de frequentie in GHz, waarop elke 'processing unit' binnen het k^e cluster staat afgesteld, $C_{eff,k}$

de capaciteit in F van het type 'processing units' bij het k^e cluster en $V_{dd,k}$ het potentiaal in V dat bij het type 'processing unit' hoort bij een frequentie van F_k . Dit alles geldt dus voor $k = 1, 2$.

Daarnaast hangen de bedieningsduren (μ_1 en μ_2) nog af van de gekozen frequentie, het type 'processing unit' en het aantal 'processing units' bij elk cluster. Hierdoor wordt $\mu_k = s_k F_k \delta_k \mu$ voor $k = 1, 2$. Met s_k het aantal 'processing units' bij het k^e cluster, F_k de frequentie in GHz , waarop elke 'processing unit' binnen het k^e cluster staat afgesteld, δ_k een factor die afhankelijk is van het type 'processing units' dat bij het k^e cluster aanwezig is en μ de gemiddelde bedieningsduur bij 1 standaard ($\delta = 1$) server die ingesteld staat op een frequentie van 1 GHz .

Opmerking : In dit verslag is $s_1 = s_2 = 1$, en is er gekozen om de s_1 en s_2 in de uitwerkingen te laten, voor eventueel gebruik van cluster 'processing units' met meer servers.

Ondanks dat μ_1 en μ_2 uitgebreider gedefinieerd zijn, verandert er niets aan de vergelijkingen eerder in dit hoofdstuk. En dus ook niet aan de uitwerkingen ervan, eveneens in hoofdstuk 3.

Daarnaast wordt er in dit verslag alleen gekeken naar dezelfde type 'processing units', waardoor $C_{eff,1} = C_{eff,2}$ en $\delta_1 = \delta_2$. Voor het gemak worden deze gelijk gesteld aan 1, zodat deze termen wegvallen. Ditzelfde geldt voor μ , zodat: $P_{d,k} = s_k V_{dd,k}^2 F_k$ en $\mu_k = s_k F_k$, $k = 1, 2$.

3.3.1 Gemiddeld energieverbruik bij de kortste gemiddelde behandelduur

Aangezien de voorgaande vergelijkingen niet veranderen, is de gemiddelde behandelduur bij zowel type 1, type 2 als type 3 aankomsten omschreven in Vergelijking (13), en is als volgt:

$$t(\alpha) = \frac{1}{\lambda_1 + \lambda_2 + \lambda_3} \left[\frac{\lambda_1 + \lambda_3 \alpha}{\mu_1 - [\lambda_1 + \lambda_3 \alpha]} + \frac{\lambda_2 + \lambda_3 (1 - \alpha)}{\mu_2 - [\lambda_2 + \lambda_3 (1 - \alpha)]} \right]$$

Met deze vergelijking kunnen de volgende parameters bepaald worden.

$$\begin{aligned} p_1 &= \frac{\lambda_1 + \lambda_3 \alpha}{\lambda_1 + \lambda_2 + \lambda_3} & t_1 &= \frac{1}{\mu_1 - [\lambda_1 + \lambda_3 \alpha]} \\ p_2 &= \frac{\lambda_2 + \lambda_3 (1 - \alpha)}{\lambda_1 + \lambda_2 + \lambda_3} & t_2 &= \frac{1}{\mu_2 - [\lambda_2 + \lambda_3 (1 - \alpha)]} \end{aligned}$$

Om het energieverbruik te kunnen bepalen is alleen nog het vermogen nodig van beide clusters.

$$P_{d,k} = \mu_k V_{dd,k}^2 \quad \text{met} \quad \mu_k = s_k F_k \quad k = 1, 2$$

Hierin hangt de waarde van $V_{dd,k}$ af van de gekozen waarde F_k , middels de waarden die in Tabel 1 staan. De waarde van $V_{dd,k}$ is dus niet vrij te kiezen. Na invullen wat nu bekend is, krijg je:

$$\tilde{E}^\alpha = \frac{1}{\lambda_1 + \lambda_2 + \lambda_3} \left(\frac{[\lambda_1 + \lambda_3 \alpha] P_{d,1}}{\mu_1 - [\lambda_1 + \lambda_3 \alpha]} + \frac{[\lambda_2 + \lambda_3 (1 - \alpha)] P_{d,2}}{\mu_2 - [\lambda_2 + \lambda_3 (1 - \alpha)]} \right)$$

De optimale waarde van α die bij dit proces hoort staat in Vergelijking (14).

Opmerking : Om het energieverbruik te bepalen bij alleen type 3 aankomsten, kan dezelfde vergelijking voor $\tilde{E}^\alpha$ gebruikt worden. In dat geval is $\lambda_1 = \lambda_2 = 0$. De optimale waarde van α die dan bij dit proces hoort staat in Vergelijking (10).

3.3.2 Optimalisatie energieverbruik

Om het energieverbruik te optimaliseren wordt de nieuwe kostenfunctie: $Cost(i, j) = \tilde{E}$

$$Cost(i, j) = \tilde{E} = p_1 \tilde{E}_1 + p_2 \tilde{E}_2 = p_1 P_{d,1} t_1 + p_2 P_{d,2} t_2$$

In deze formule is, voor alleen aankomsten van type 3, $p_1 t_1 = \gamma_1 i = \frac{i}{\lambda_3^*}$ en $p_2 t_2 = \gamma_2 j = \frac{j}{\lambda_3^*}$. De waarden voor γ_1 en γ_2 zijn immers al bepaald in Paragraaf 3.1.6. Verder geldt dat $P_{d,1} = \mu_1 V_{dd,1}^2$ en $P_{d,2} = \mu_2 V_{dd,2}^2$. Hierbij is opnieuw $\mu_1 = s_1 F_1$ en $\mu_2 = s_2 F_2$. Ook worden de waarden van $V_{dd,1}$ en $V_{dd,2}$ opnieuw vastgelegd door de keuze van respectievelijk F_1 en F_2 , middels de waarden in Tabel 1. Voor aankomsten van zowel type 1, type 2 als type 3, wijzigen alleen $p_1 t_1 = \frac{i}{\lambda_1^* + \lambda_2^* + \lambda_3^*}$ en $p_2 t_2 = \frac{j}{\lambda_1^* + \lambda_2^* + \lambda_3^*}$. Aangezien het vermogen alleen berekend wordt met gekozen waarden, en dus een constante binnen het systeem is, blijft deze deels staan in de uitwerkingen om het overzichtelijk te houden.

Voor aankomsten van zowel type 1, type 2 als type 3 ontstaat, na toepassing van de Bernoulli policy, de volgende vergelijking:

$$\mathbf{g}^\alpha + \mathbf{V}^\alpha(i, j) = \begin{cases} \frac{i P_{d,1} + j P_{d,2}}{\lambda_1^* + \lambda_2^* + \lambda_3^*} + [\lambda_1^* + \lambda_3^* \alpha] \mathbf{V}^\alpha(i+1, j) \\ + [\lambda_2^* + \lambda_3^* (1-\alpha)] \mathbf{V}^\alpha(i, j+1) \\ + \mu_1^* \mathbf{V}^\alpha(i-1, j) \mathbb{1}_{i>0} \\ + \mu_2^* \mathbf{V}^\alpha(i, j-1) \mathbb{1}_{j>0} \\ + [1 - (\lambda_1^* + \lambda_2^* + \lambda_3^* + \mu_1^* \mathbb{1}_{i>0} + \mu_2^* \mathbb{1}_{j>0})] \mathbf{V}^\alpha(i, j) \end{cases} \quad (15)$$

Opmerking : Deze vergelijking lijkt erg op Vergelijking (11), alleen zijn de kosten anders.

Corollarium 3.1. De volgende $\mathbf{g}^\alpha$ en $\mathbf{V}^\alpha(i, j)$ zijn een oplossing van Vergelijking (15).

$$\mathbf{V}^\alpha(i, j) = \frac{1}{2(\lambda_1^* + \lambda_2^* + \lambda_3^*)} \left(\frac{i(i+1) P_{d,1}}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{j(j+1) P_{d,2}}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} \right)$$

$$\mathbf{g}^\alpha = \frac{1}{\lambda_1^* + \lambda_2^* + \lambda_3^*} \left(\frac{[\lambda_1^* + \lambda_3^* \alpha] P_{d,1}}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{[\lambda_2^* + \lambda_3^* (1-\alpha)] P_{d,2}}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} \right)$$

Bewijs : Door in Vergelijking (11) in te vullen dat de waarde van $\gamma_1 = \frac{P_{d,1}}{\lambda_1^* + \lambda_2^* + \lambda_3^*}$ en $\gamma_2 = \frac{P_{d,2}}{\lambda_1^* + \lambda_2^* + \lambda_3^*}$, ontstaat precies dezelfde vergelijking als in Vergelijking (15). Van Vergelijking (11) is de oplossing bekend, en deze staat in Stelling 3.2. Dit is namelijk:

$$\mathbf{V}^\alpha(i, j) = \frac{1}{2} \left(\frac{\gamma_1 i(i+1)}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{\gamma_2 j(j+1)}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} \right)$$

$$\mathbf{g}^\alpha = \frac{\gamma_1 [\lambda_1^* + \lambda_3^* \alpha]}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{\gamma_2 [\lambda_2^* + \lambda_3^* (1-\alpha)]}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]}$$

Door hierin de waarden van γ_1 en γ_2 in te vullen, namelijk $\gamma_1 = \frac{P_{d,1}}{\lambda_1^* + \lambda_2^* + \lambda_3^*}$ en $\gamma_2 = \frac{P_{d,2}}{\lambda_1^* + \lambda_2^* + \lambda_3^*}$, krijg je precies de oplossing die in Corollarium 3.1 omschreven staat.

Q.E.D.

Opmerking : Als $P_{d,1} = P_{d,2} = 1$, dan is dit precies de gemiddelde behandelduur.

Door de $\mathbf{g}^\alpha (= \tilde{\mathbf{E}}(\alpha))$ te differentiëren, en deze afgeleide gelijk te stellen aan nul, kan een optimale waarde voor α gevonden worden. De uiteindelijk gevonden waarde ($\hat{\alpha}_E$) is dan de optimale waarde van α die het gemiddeld energieverbruik, onder de Bernoulli policy, minimaliseert.

$$\tilde{\mathbf{E}}'(\alpha) = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \left(\frac{\mu_1 P_{d,1}}{(\mu_1 - [\lambda_1 + \lambda_3 \alpha])^2} - \frac{\mu_2 P_{d,2}}{(\mu_2 - [\lambda_2 + \lambda_3 (1-\alpha)])^2} \right)$$

Met behulp van Mathematica is deze afgeleide gelijk gesteld aan nul, met als resultaat:

$$\alpha_{1,2} = \frac{P_{d,1} \lambda_3 \mu_1 [\lambda_2 + \lambda_3 - \mu_2] + P_{d,2} \lambda_3 \mu_2 [\lambda_1 - \mu_1] \pm \sqrt{P_{d,1} P_{d,2} (\lambda_3)^2 \mu_1 \mu_2 [\lambda_1 + \lambda_2 + \lambda_3 - \mu_1 - \mu_2]^2}}{(\lambda_3)^2 [P_{d,1} \mu_1 - P_{d,2} \mu_2]}$$

Deze gevonden α_1 en α_2 zijn, naast de grenswaarden $\alpha = 0$ en $\alpha = 1$, kandidaten voor $\hat{\alpha}_E$. Daarnaast moet gelden dat $0 \leq \hat{\alpha}_E \leq 1$. Dit samen maakt:

$$\hat{\alpha}_E = \left\{ x \in \{0, 1, \alpha_1 \mathbb{1}_{0 < \alpha_1 < 1}, \alpha_2 \mathbb{1}_{0 < \alpha_2 < 1}\} : \tilde{E}(x) \leq \tilde{E}(y) \text{ , } \forall y \in [0, 1] \right\}$$

Opmerking : Om het energieverbruik te optimaliseren bij alleen type 3 aankomsten, kan dezelfde vergelijking voor $\tilde{E}(\alpha)$ gebruikt worden. In dat geval is opnieuw $\lambda_1 = \lambda_2 = 0$.

3.4 One-step policy improvement

Nadat een optimale waarde van α (onder de Bernoulli policy) gevonden is, kan er nog 'one-step policy improvement' toegepast worden (met de optimale waarde voor α). Met deze methode worden opdrachten niet willekeurig naar een server gestuurd, maar hangt de keuze af van de huidige toestand waarin het systeem zich bevindt. Om een 'one-step policy improvement' op te stellen, wordt er gekeken naar de minimalisatieterm in het DTMDP-model. Voor de optimalisatie van de gemiddelde behandelduur bij alleen type 3 aankomsten staat deze in Vergelijking (5).

Opmerking : Om ruimte te besparen is ervoor gekozen om, bij de optimalisatie van het gemiddeld energieverbruik en bij de optimalisatie van de gemiddelde behandelduur voor zowel type 1, type 2 als type 3 aankomsten, geen DTMDP-model weer te geven. Dit is opzich geen probleem, want de minimalisatieterm in al deze gevallen is precies dezelfde als die in Vergelijking (5).

In deze minimalisatieterm wordt dan de $\mathbf{V}(i, j)$ vervangen door de gevonden $\mathbf{V}^\alpha(i, j)$.

$$\min \{ \mathbf{V}^\alpha(i+1, j) ; \mathbf{V}^\alpha(i, j+1) \}$$

Merk op dat de verbeterde politiek opdrachten naar de eerste server stuurt, als:

$$\begin{aligned} \mathbf{V}^\alpha(i+1, j) &\leq \mathbf{V}^\alpha(i, j+1) && \Longleftrightarrow \\ \frac{(i+2)(i+1)P_{d,1}}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{j(j+1)P_{d,2}}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} &\leq \frac{i(i+1)P_{d,1}}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{(j+2)(j+1)P_{d,2}}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} && \Longleftrightarrow \\ \frac{2(i+1)P_{d,1}}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} &\leq \frac{2(j+1)P_{d,2}}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} && \Longleftrightarrow \\ \frac{(i+1)P_{d,1}}{\mu_1 - [\lambda_1 + \lambda_3 \alpha]} &\leq \frac{(j+1)P_{d,2}}{\mu_2 - [\lambda_2 + \lambda_3 (1-\alpha)]} && \Longleftrightarrow \\ (\mu_2 - [\lambda_2 + \lambda_3 (1-\alpha)])(i+1)P_{d,1} &\leq (\mu_1 - [\lambda_1 + \lambda_3 \alpha])(j+1)P_{d,2} \end{aligned}$$

En als $\mathbf{V}^\alpha(i+1, j) > \mathbf{V}^\alpha(i, j+1)$, dan stuurt deze politiek opdrachten naar de tweede server.

Opmerking : Bij optimalisatie van de gemiddelde behandelduur, neem: $P_{d,1} = P_{d,2} = 1$.

Opmerking : Bij alleen aankomsten van type 3, neem: $\lambda_1 = \lambda_2 = 0$.

4 Wiskundige simulatiemodellen

De simulatiemodellen zijn gemodelleerd als een 'discrete event' simulatie en volgens de replicatiemethode. Dit betekent dat er per simulatie een vooraf ingesteld aantal deelsimulaties worden uitgevoerd. En per deelsimulatie wordt een vooraf ingesteld aantal opdrachten afgerond. Van elke deelsimulatie wordt dan de gemiddelde behandelduur en het gemiddeld energieverbruik opgeslagen. Nadat alle deelsimulaties afgerond zijn, wordt er van zowel de set gemiddelde behandelduren, als van de set gemiddeld energieverbruik, het steekproefgemiddelde en de steekproefvariantie bepaald. Samen met het aantal uitgevoerde deelsimulaties kan dan, voor beide stochasten, een 95% betrouwbaarheidsinterval opgesteld worden.

Om hierbij 'processor sharing' te simuleren wordt de nog te behandelen tijd bijgehouden. Deze wordt geschaald (vermenigvuldigd) met het aantal opdrachten aanwezig bij die desbetreffende server. Bij een nieuwe gebeurtenis, wordt de verstreken tijd als eerste verwerkt. Als deze nieuwe gebeurtenis een aankomst of vertrek is bij die desbetreffende server, dan worden alle opdrachten bij die server daarna eerst teruggeschaald, voordat de aankomst of vertrek zoals gebruikelijk verwerkt wordt. Daarna worden ze opnieuw geschaald om de nieuwe nog te behandelen tijd in deze nieuwe situatie te bepalen.

Voor het bepalen van de gemiddelde behandelduur, wordt voor elke opdracht afzonderlijk bijgehouden hoelang deze daadwerkelijk bij de server aanwezig is. Zodra een opdracht afgerond wordt, dan vertrekt deze bij de server en worden zijn gegevens opgeslagen in desbetreffende lijst. Beide servers hebben elk hun eigen lijst met afgeronde opdrachten. Door de behandeltijden in één lijst bij elkaar op te tellen en te delen door de lengte van de lijst (aantal afgeronde opdrachten bij die server), wordt de gemiddelde behandeltijd bij die desbetreffende server bepaald. En door de behandeltijden in beide lijsten bij elkaar op te tellen en te delen door het aantal afgeronde opdrachten, wordt de gemiddelde behandeltijd van het gehele systeem bepaald.

Voor het energieverbruik wordt voor beide servers de tijd bijgehouden dat deze server in gebruik is. Dus in het geval dat er minimaal één opdracht bij die server aanwezig is. Deze tijd wordt vermenigvuldigd met het vermogen van de desbetreffende server, waardoor per server het totale energieverbruik bepaald wordt. Deze kun je van beide servers bij elkaar optellen en delen door het totaal aantal afgeronde opdrachten, om het gemiddelde energieverbruik van het totale systeem te krijgen. Maar je kunt ze ook afzonderlijk delen door het aantal afgeronde opdrachten bij die server, om het gemiddelde energieverbruik per server te bepalen. Hierbij is de aanname gemaakt dat er bij een server geen energieverbruik is, als er geen opdracht aanwezig is bij die server.

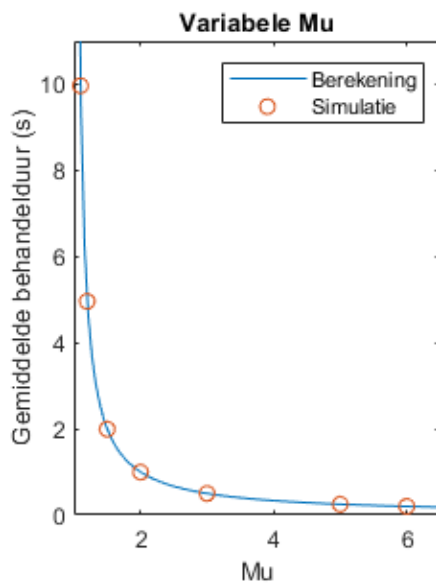
Deze manier om het gemiddeld energieverbruik te bepalen is opzich vrij logisch, maar wijkt het wel af van het model dat omschreven staat in Paragraaf 3.3. Hier is echter bewust voor gekozen. Door het energieverbruik namelijk op zichzelf te simuleren, en dus niet te bepalen aan de hand van de gemiddelde behandelduur, kan er meer gezegd worden over de betrouwbaarheid/werkbaarheid van het model in Paragraaf 3.3, en dus daarmee hoe toepasbaar het is.

Om dit te simuleren zijn er meerdere modellen gemaakt (zie Appendix: G voor een overzicht van de verschillende modellen). De meeste modellen zijn gebruikt om als tussenstap te dienen, en deze worden verder niet gebruikt in het verslag. De modellen die wel gebruikt worden, zijn: een simpele/enkele 'processor sharing' queue (ofwel alleen aankomsten van type 1), een model met alleen aankomsten van type 3 en een model met alle type aankomsten (type 1, type 2 en type 3). Voor deze laatste 2 modellen is zowel een Bernoulli policy als een 'one-step policy improvement' geprogrammeerd. In dit verslag zullen dan ook beide policy's worden gebruikt.

5 Resultaten

5.1 Een enkele 'processor sharing' queue, ofwel alleen aankomsten van type 1

In deze paragraaf wordt er gekeken naar een simpele/enkele 'processor sharing' queue (ofwel alleen aankomsten van type 1). Hierbij is allereerst gekeken naar de gemiddelde behandeldduur bij een variabele waarde van μ en waarbij de $\lambda (= 1)$ een constante is.

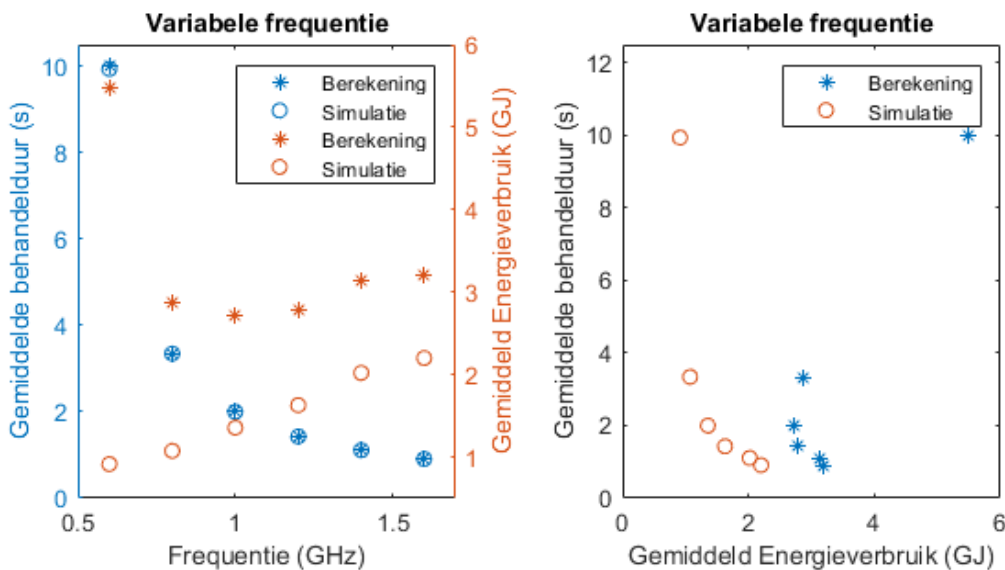


Figuur 2: Gemiddelde behandeldduur in seconden bij een variabele μ , waarbij $\lambda (= 1)$ een constante is.

μ	Gemiddelde behandeldduur (s)	
	Berekening	95% B.I. Simulatie
1.1	10	[9.9306 , 9.9936]
1.2	5	[4.9431 , 4.9736]
1.5	2	[1.9863 , 1.9973]
2	1	[0.9925 , 0.9975]
3	0.5	[0.4990 , 0.5013]
5	0.25	[0.2491 , 0.2501]
6	0.2	[0.1995 , 0.2003]

Tabel 4: Gemiddelde behandeldduur in seconden bij een variabele μ , waarbij $\lambda (= 1)$ een constante is.

De uitwerkingen hiervan staan in Tabel 4. Daarnaast is hiervan ook een grafiek geplott, deze staat in Figuur 2. Verder is er gekeken naar de gemiddelde behandeldduur en het gemiddeld energieverbruik bij verschillende frequenties. De uitwerkingen hiervan staan in Tabel 5 en de bijbehorende grafieken staan in Figuur 3. Hierbij is in de linker grafiek, de gemiddelde behandeldduur in het blauw weergegeven en het gemiddeld energieverbruik in het rood. En in de rechter grafiek is het energieverbruik uitgezet tegen de gemiddelde behandeldduur.



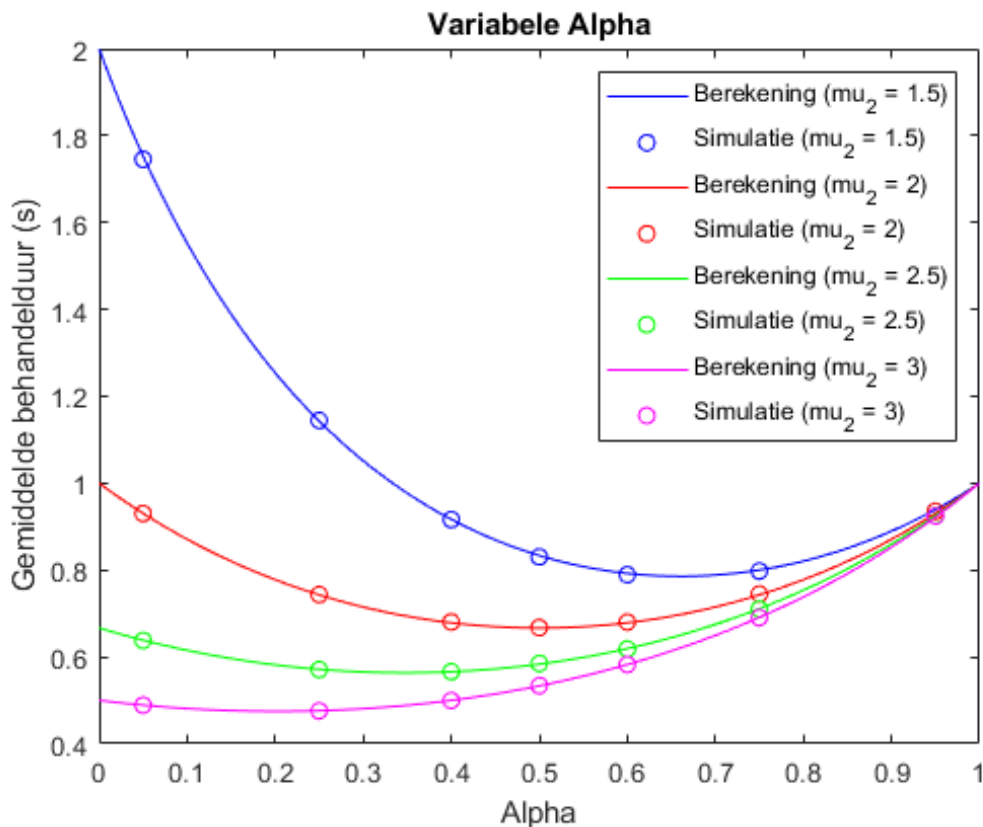
Figuur 3: De gemiddelde behandeldduur in seconden en het gemiddelde energieverbruik in GigaWatt bij een variabele frequentie. Hierbij is $\lambda (= 0.5)$ een constante.

Freq. (GHz)	Gemiddelde behandelduur (s)		Gemiddelde Energieverbruik (GJ)	
	Berekening	95% B.I. Simulatie	Berekening	95% B.I. Simulatie
0.6	10	[9.7356 , 10.1449]	5.4836	[0.9117 , 0.9154]
0.8	3.3333	[3.3128 , 3.3785]	2.8621	[1.0717 , 1.0760]
1.0	2	[1.9875 , 2.0137]	2.7098	[1.3533 , 1.3585]
1.2	1.4286	[1.4221 , 1.4343]	2.7912	[1.6253 , 1.6314]
1.4	1.1111	[1.1063 , 1.1158]	3.1366	[2.0156 , 2.0237]
1.6	0.9091	[0.9052 , 0.9118]	3.2033	[2.1959 , 2.2041]

Tabel 5: De gemiddelde behandelduur in seconden en het gemiddelde energieverbruik in GigaWatt bij een variabele frequentie. Hierbij is $\lambda (= 0.5)$ een constante.

5.2 Alleen aankomsten van type 3

Vanaf deze paragraaf worden de resultaten interessanter, aangezien de opdrachten nu daadwerkelijk verdeeld worden over de beschikbare 'processing units'. Voor deze resultaten wordt hoofdzakelijk de Bernoulli policy gebruikt. De 'one-step policy improvement' (OSPI) wordt daarentegen alleen gebruikt in de tabellen, en daarbij slechts alleen in de aangegeven regels. Daarnaast staat bij elke OSPI de gebruikte waarde voor α (tussen haakjes) daarbij aangegeven.



Figuur 4: Gemiddelde behandelduur in seconden bij een variabele α . De waarden van $\lambda_3 (= 1)$, $\mu_1 (= 2)$ en μ_2 zijn hierbij een constante. Dit is gedaan voor verschillende constante waarden van μ_2 .

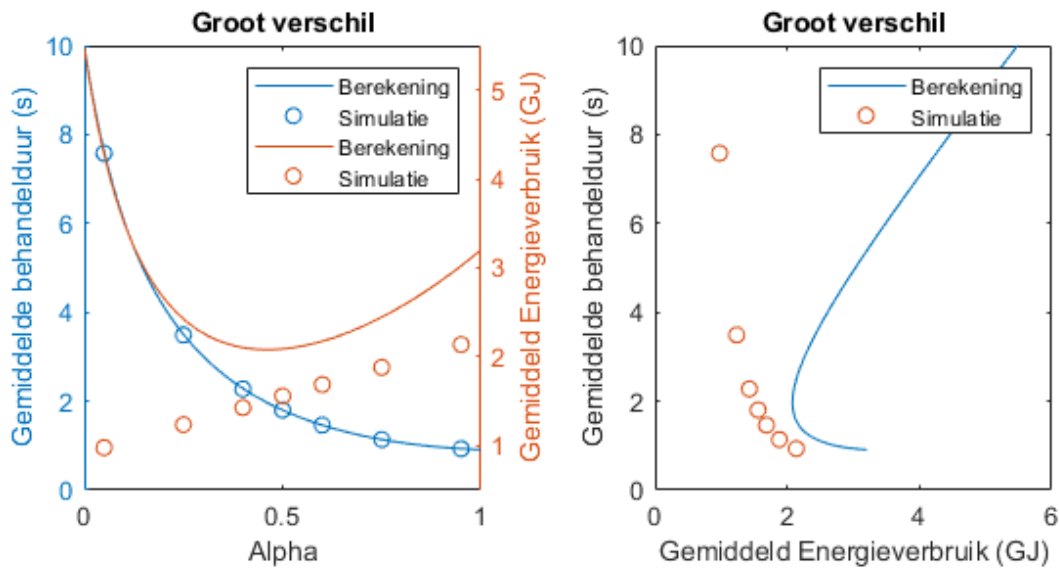
Allereerst is er gekeken naar de gemiddelde behandelduur bij een variabele α . Dit is gedaan met constante waarden voor λ_3 , μ_1 en μ_2 . Om dit bij verschillende waarden/verhoudingen van μ te kunnen onderzoeken, is ervoor gekozen om dit te doen voor verschillende constante waarden van μ_2 . De waarden van $\lambda_3 (= 1)$ en $\mu_1 (= 2)$ zullen hierbij (in elke situatie) hetzelfde blijven. De resultaten hiervan staan weergegeven in Figuur 4 en in Tabel 6.

		Gemiddelde behandelduur (s)	
	α	Berekening	95% B.I. Simulatie
$\mu_2 = 1.5$	0.05	1.7529	[1.7295 , 1.7615]
	0.25	1.1429	[1.1371 , 1.1514]
	0.4	0.9167	[0.9118 , 0.9204]
	0.5	0.8333	[0.8273 , 0.8334]
	0.6	0.7922	[0.7866 , 0.7922]
	0.75	0.8000	[0.7941 , 0.8012]
	0.95	0.9392	[0.9300 , 0.9406]
$\hat{\alpha}_t$	0.6603	0.7856	[0.7819 , 0.7875]
OSPI _t ($\hat{\alpha}_t$)		X	[0.6118 , 0.6154]
$\mu_2 = 2$	0.05	0.9304	[0.9250 , 0.9351]
	0.25	0.7429	[0.7400 , 0.7468]
	0.4	0.6786	[0.6781 , 0.6827]
	0.5	0.6667	[0.6653 , 0.6698]
	0.6	0.6786	[0.6778 , 0.6826]
	0.75	0.7429	[0.7419 , 0.7478]
	0.95	0.9304	[0.9304 , 0.9397]
$\hat{\alpha}_t$	0.5	0.6667	[0.6656 , 0.6701]
OSPI _t ($\hat{\alpha}_t$)		X	[0.5488 , 0.5521]

		Gemiddelde behandelduur (s)	
	α	Berekening	95% B.I. Simulatie
$\mu_2 = 2.5$	0.05	0.6385	[0.6343 , 0.6407]
	0.25	0.5714	[0.5691 , 0.5726]
	0.4	0.5658	[0.5643 , 0.5678]
	0.5	0.5833	[0.5832 , 0.5865]
	0.6	0.6190	[0.6159 , 0.6205]
	0.75	0.7111	[0.7072 , 0.7130]
	0.95	0.9252	[0.9203 , 0.9302]
$\hat{\alpha}_t$	0.3475	0.5635	[0.5623 , 0.5660]
OSPI _t ($\hat{\alpha}_t$)		X	[0.4553 , 0.4576]
$\mu_2 = 3$	0.05	0.4891	[0.4869 , 0.4907]
	0.25	0.4762	[0.4744 , 0.4773]
	0.4	0.5000	[0.4983 , 0.5010]
	0.5	0.5333	[0.5319 , 0.5357]
	0.6	0.5824	[0.5802 , 0.5845]
	0.75	0.6909	[0.6873 , 0.6936]
	0.95	0.9217	[0.9190 , 0.9298]
$\hat{\alpha}_t$	0.2020	0.4747	[0.4736 , 0.4765]
OSPI _t ($\hat{\alpha}_t$)		X	[0.3894 , 0.3912]

Tabel 6: Gemiddelde behandelduur in seconden bij een variabele α . De waarden van $\lambda_3(=1)$, $\mu_1(=2)$ en μ_2 zijn hierbij een constante. Dit is gedaan voor verschillende constante waarden van μ_2 .

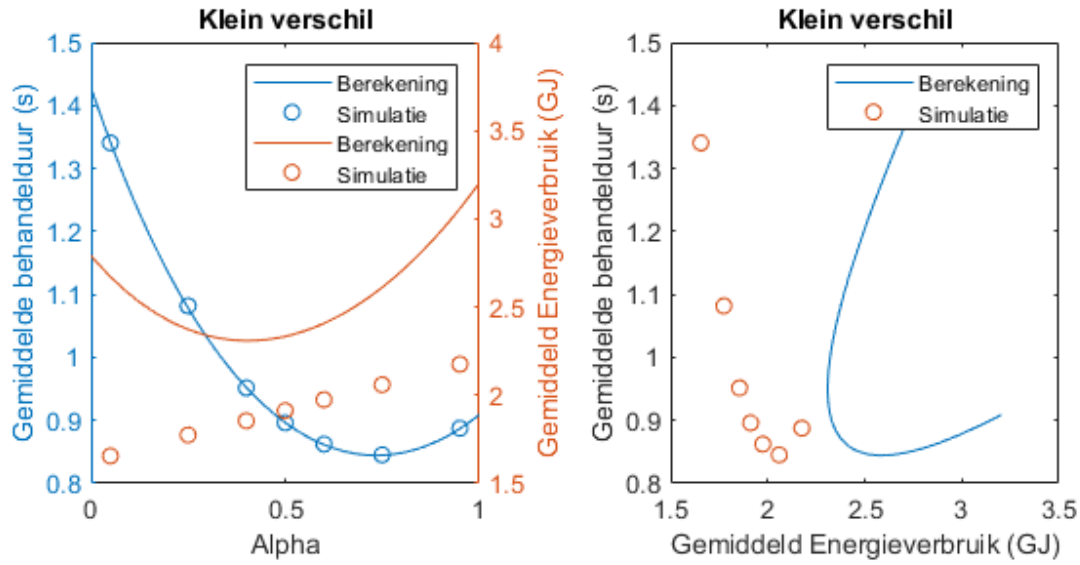
Om te kijken naar het energieverbruik, is er gekozen om beide servers op een verschillende snelheid te laten werken. Hiervoor zijn 2 situaties gecreëerd, één met een groot verschil in snelheid en de andere met een kleiner verschil. In beide situaties zijn de constanten $\lambda_3(=0.5)$ en $F_1(=1.6)$ hetzelfde. Bij het grotere verschil is de constante $F_2 = 0.6$. De resultaten hiervan staan in Figuur 5 en Tabel 7. En bij het kleinere verschil is de constante $F_2 = 1.2$. En staan de resultaten weergegeven in Figuur 6 en Tabel 8. In beide figuren is in het linker plaatje de gemiddelde behandelduur weergegeven in blauw en het gemiddeld energieverbruik in rood.



Figuur 5: De gemiddelde behandelduur in seconden en het gemiddelde energieverbruik in GigaWatt bij een variabele α en een groot verschil in frequentie. Hierbij zijn $\lambda_3(=0.5)$, $F_1(=1.6)$ en $F_2(=0.6)$ een constante.

		Gemiddelde behandelduur (s)		Gemiddelde Energieverbruik (GJ)	
	α	Berekening	95% B.I. Simulatie	Berekening	95% B.I. Simulatie
$F_1 = 1.6$ $F_2 = 0.6$	0.05	7.6317	[7.4523 , 7.7060]	4.2794	[0.9758 , 0.9796]
	0.25	3.5028	[3.4590 , 3.5222]	2.4251	[1.2338 , 1.2395]
	0.4	2.2857	[2.2604 , 2.2913]	2.1035	[1.4253 , 1.4316]
	0.5	1.7989	[1.7907 , 1.8128]	2.0884	[1.5550 , 1.5619]
	0.6	1.4615	[1.4553 , 1.4699]	2.1746	[1.6831 , 1.6910]
	0.75	1.1386	[1.1332 , 1.1421]	2.4459	[1.8763 , 1.8852]
	0.95	0.9314	[0.9276 , 0.9344]	3.0232	[2.1338 , 2.1420]
$\hat{\alpha}_t$	1.0	0.9091	[0.9073 , 0.9137]	3.2033	[2.1989 , 2.2066]
$\hat{\alpha}_E$	0.4615	1.9652	[1.9478 , 1.9722]	2.0804	[1.5050 , 1.5119]
OSPI _t ($\hat{\alpha}_t$)		X	[0.8741 , 0.8789]	X	[1.9613 , 1.9686]
OSPI _E ($\hat{\alpha}_E$)		X	[1.4525 , 1.4606]	X	[1.4269 , 1.4335]

Tabel 7: De gemiddelde behandelduur in seconden en het gemiddelde energieverbruik in GigaWatt bij een variabele α en een groot verschil in frequentie. Hierbij zijn $\lambda_3(= 0.5)$, $F_1(= 1.6)$ en $F_2(= 0.6)$ een constante.



Figuur 6: De gemiddelde behandelduur in seconden en het gemiddelde energieverbruik in GigaWatt bij een variabele α en een klein verschil in frequentie. Hierbij zijn $\lambda_3(= 0.5)$, $F_1(= 1.6)$ en $F_2(= 1.2)$ een constante.

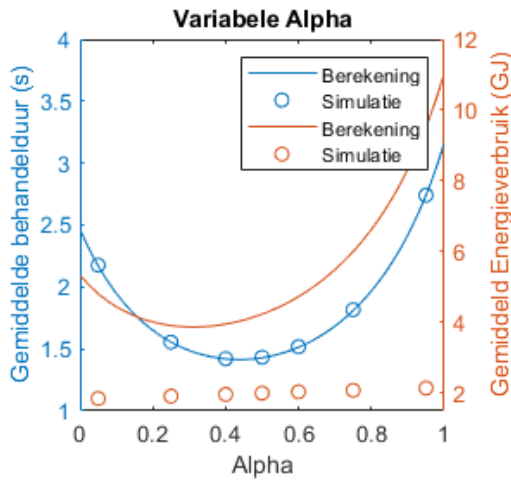
		Gemiddelde behandelduur (s)		Gemiddelde Energieverbruik (GJ)	
	α	Berekening	95% B.I. Simulatie	Berekening	95% B.I. Simulatie
$F_1 = 1.6$ $F_2 = 1.2$	0.05	1.3421	[1.3340 , 1.3471]	2.6720	[1.6522 , 1.6595]
	0.25	1.0786	[1.0778 , 1.0858]	2.3734	[1.7704 , 1.7783]
	0.4	0.9524	[0.9485 , 0.9545]	2.3093	[1.8512 , 1.8587]
	0.5	0.8967	[0.8932 , 0.8991]	2.3334	[1.9084 , 1.9163]
	0.6	0.8615	[0.8597 , 0.8649]	2.4078	[1.9707 , 1.9797]
	0.75	0.8448	[0.8426 , 0.8482]	2.6117	[2.0553 , 2.0642]
	0.95	0.8870	[0.8844 , 0.8908]	3.0586	[2.1715 , 2.1801]
$\hat{\alpha}_t$	0.7349	0.8446	[0.8420 , 0.8475]	2.5862	[2.0439 , 2.0527]
$\hat{\alpha}_E$	0.4034	0.9501	[0.9470 , 0.9529]	2.3093	[1.8576 , 1.8650]
OSPI _t ($\hat{\alpha}_t$)		X	[0.6989 , 0.7023]	X	[2.0775 , 2.0857]
OSPI _E ($\hat{\alpha}_E$)		X	[0.8247 , 0.8282]	X	[1.7776 , 1.7847]

Tabel 8: De gemiddelde behandelduur in seconden en het gemiddelde energieverbruik in GigaWatt bij een variabele α en een klein verschil in frequentie. Hierbij zijn $\lambda_3(= 0.5)$, $F_1(= 1.6)$ en $F_2(= 1.2)$ een constante.

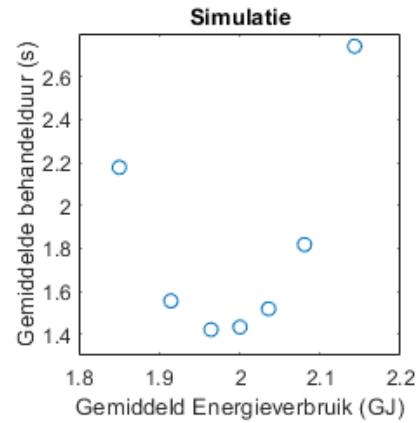
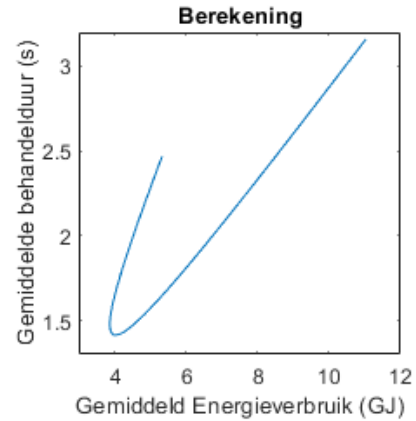
5.3 Aankomsten van zowel type 1, type 2 als type 3

De Bernoulli policy en de 'one-step policy improvement' worden op dezelfde manier gebruikt als in de vorige paragraaf. Aangezien de gemiddelde behandelduur bij een variabele α voor deze situatie erg lijkt op die uit de vorige paragraaf, is ervoor gekozen om deze resultaten weer te geven in Appendix H.

Voor de situatie waarin $\lambda_1(= 0.5)$, $\lambda_2(= 0.1)$, $\lambda_3(= 0.8)$, $F_1(= 1.6)$ en $F_2(= 1.2)$ een constante zijn, worden de resultaten weergegeven in Figuur 7, Figuur 8 en in Tabel 9.



Figuur 7: Gemiddelde behandelduur in seconden en gemiddeld energieverbruik in GigaWatt bij een variabele α . Hierbij zijn de waarden van $\lambda_1(= 0.5)$, $\lambda_2(= 0.1)$, $\lambda_3(= 0.8)$, $F_1(= 1.6)$ en $F_2(= 1.2)$ een constante.



Figuur 8: Gemiddeld energieverbruik in GigaWatt uitgezet tegen de gemiddelde behandelduur in seconden. De waarden van $\lambda_1(= 0.5)$, $\lambda_2(= 0.1)$, $\lambda_3(= 0.8)$, $F_1(= 1.6)$ en $F_2(= 1.2)$ zijn hierbij een constante.

		Gemiddelde behandelduur (s)		Gemiddelde Energieverbruik (GJ)	
α		Berekening	95% B.I. Simulatie	Berekening	95% B.I. Simulatie
$F_1 = 1.6$ $F_2 = 1.2$	0.05	2.1706	[2.1474 , 2.2055]	4.8122	[1.8468 , 1.8543]
	0.25	1.5556	[1.5443 , 1.5647]	3.9114	[1.9111 , 1.9181]
	0.4	1.4191	[1.4110 , 1.4306]	3.9515	[1.9601 , 1.9689]
	0.5	1.4286	[1.4230 , 1.4425]	4.2328	[1.9966 , 2.0043]
	0.6	1.5136	[1.5069 , 1.5288]	4.7297	[2.0320 , 2.0399]
	0.75	1.8095	[1.7951 , 1.8384]	6.0023	[2.0762 , 2.0851]
	0.95	2.7414	[2.6939 , 2.7885]	9.5115	[2.1385 , 2.1473]
	$\hat{\alpha}_t$	0.4372	[1.4045 , 1.4218]	4.0326	[1.9759 , 1.9847]
$\hat{\alpha}_E$	0.3111	1.4758	[1.4661 , 1.4873]	3.8701	[1.9355 , 1.9429]
	OSPI _t ($\hat{\alpha}_t$)	X	[1.0963 , 1.1075]	X	[2.0127 , 2.0205]
	OSPI _E ($\hat{\alpha}_E$)	X	[1.1110 , 1.1215]	X	[1.9332 , 1.9411]

Tabel 9: Gemiddelde behandelduur in seconden en gemiddeld energieverbruik in GigaWatt bij een variabele α . Hierbij zijn de waarden van $\lambda_1(= 0.5)$, $\lambda_2(= 0.1)$, $\lambda_3(= 0.8)$, $F_1(= 1.6)$ en $F_2(= 1.2)$ een constante.

6 Discussie

Bij het bestuderen van de resultaten, valt allereerst op dat de berekeningen voor het gemiddeld energieverbruik niet overeenkomen met de simulaties. De berekening voor de gemiddelde behandelduur komt daarentegen sterk overeen met de simulaties. Zo sterk zelfs dat de berekeningen, in de meeste situaties, binnen het 95% - betrouwbaarheidsinterval (van de simulaties) vallen, of daarbij in de buurt liggen. Dit maakt dat zowel de berekening als het simulatiemodel, wat betreft de gemiddelde behandelduur, zeer betrouwbaar lijken te zijn.

Dat de resultaten van het gemiddeld energieverbruik niet met elkaar in overeenstemming zijn, is op zich vrij logisch te verklaren. Voor het energieverbruik wordt namelijk de volgende formule gebruikt: $E = p_d t$, en hierbij wordt de variabele t verschillend gedefinieerd. In het geval van de berekening, is deze t gelijk aan de gemiddelde behandelduur. Bij het simulatiemodel is bewust gekozen om hiervan af te wijken, en deze t gelijk te stellen aan de gemiddelde tijd dat een 'processing unit' in bedrijf is. Bij de berekening wordt er namelijk geen rekening gehouden met het 'processor sharing' aspect van beide servers. Hierdoor wordt het energieverbruik van een server meerdere keren geteld, als er meerdere opdrachten bij die server aanwezig zijn. Maar de server gaat in zo'n geval niet harder werken, waardoor het verbruik dus ongeveer hetzelfde is als bij de behandeling van 1 opdracht. Het zou daarom interessant kunnen zijn om ook hiervoor een wiskundig model op te stellen. Dit zou bijv. in een ander onderzoek gedaan kunnen worden.

Om de motivatie voor het simulatiemodel nog meer te onderbouwen wordt Tabel 5 bekeken. Als hierin de waarden van het energieverbruik bij 1.6 GHz gedeeld worden door de waarden van het energieverbruik bij 0.8 GHz, voor zowel de berekening als het energieverbruik, dan levert dit respectievelijk een factor 1.12 en een factor 2.05 op. En als ditzelfde gedaan wordt door de waarden van het energieverbruik bij 1.2 GHz te delen door de waarden van het energieverbruik bij 0.6 GHz, dan is dit respectievelijk een factor 0.51 en een factor 1.78. Hierbij komt het simulatiemodel overeen met de gevonden factoren in Paragraaf 2.1, en voelt dit logischer.

Het bepalen van de optimale waarde voor $\hat{\alpha}_t$ lijkt, voor zowel de berekeningen als de simulaties, zeer goed te werken. In alle tabellen was dit namelijk, onder de Bernoulli policy, de meest optimale waarde voor α (als er gekeken wordt naar de gemiddelde behandelduur). Voor de optimale waarde $\hat{\alpha}_E$ geldt dit alleen bij de berekeningen. Bij de simulaties wordt het gemiddeld energieverbruik steeds lager, naarmate er meer opdrachten naar de server worden gestuurd die een lagere frequentie heeft. Maar dit heeft wel van negatieve invloed op de behandelduur.

Voor de 'one-step policy improvement' zijn de resultaten zeer positief. Dit was zeer duidelijk te zien bij de $OSPI_t$, maar ook de $OSPI_E$ verbeterde het energieverbruik voor alle gebruikte α 's, en dat terwijl de gebruikte α 's verre van optimaal zijn voor het energieverbruik bij het simulatiemodel.

Om straks bij de conclusie een antwoord te kunnen geven op de onderzoeksvraag, worden Figuur 5, Figuur 6, Figuur 7 en Figuur 8 nog eerst beter bekeken. In Paragraaf 2.2 werd er namelijk gesproken over de 'performance per Watt', en dat minimalisatie van het energieverbruik hiertoe zou leiden. Bij de berekeningen in Paragraaf 3.3.2 is de kostenfunctie: $Cost(i, j) = \tilde{E}$. Door deze kosten te minimaliseren, wat gedaan is in deze berekeningen, zou je dus iets krijgen als de 'performance per Watt'. Het zou hierdoor dus goed mogelijk kunnen zijn dat de berekening van het energieverbruik, niet het energieverbruik weergeeft, maar de 'performance per Watt'. Als met deze nieuwe inzichten wordt gekeken naar de grafieken, zou dit best wel eens kunnen kloppen. Daarnaast verbetert de $OSPI_E$ in alle situaties ook de gemiddelde behandelduur bij $\hat{\alpha}_E$, waardoor het dus aannemelijk is dat deze optimalisatie ook verband houdt met de verbetering van de gemiddelde behandelduur. Of dit ook daadwerkelijk de 'performance per Watt' is zou nog verder onderzocht moeten worden. Maar voor nu lijkt dit een goede maatgever voor optimalisatie.

7 Conclusie

In dit verslag is gezocht naar een optimale verdeling/toewijzing van opdrachten over de beschikbare 'processing units', voor zowel de gemiddelde behandelduur als het gemiddeld energieverbruik. Dit is gedaan, omdat de onderzoeksvraag deze twee factoren combineert. Daarbij wordt er namelijk gezocht naar een zo optimaal mogelijk energieverbruik, waarbij de gemiddelde behandelduur nog acceptabel is. Uit de discussie blijkt dat de gebruikte berekeningen voor optimalisatie van het gemiddeld energieverbruik, hiervoor een goede maatgever is.

Met deze maatgever is er dus een passende oplossing gevonden voor de onderzoeksvraag, en deze ziet er als volgt uit. Stuur opdrachten naar de eerste server, als:

$$(\mu_2 - [\lambda_2 + \lambda_3 (1 - \hat{\alpha}_E)]) (i + 1) P_{d,1} \leq (\mu_1 - [\lambda_1 + \lambda_3 \hat{\alpha}_E]) (j + 1) P_{d,2}$$

In alle andere gevallen (dus bij $>$) stuur je een opdracht naar de tweede server.

Referenties

- [1] Sandjai Bhulai and Ger Koole. On the structure of value functions for threshold policies in queueing models. *Journal of Applied Probability*, pages 613–622, 2003.
- [2] Jacek Błażewicz, Safia Kedad-Sidhoum, Florence Monna, Grégory Mounié, and Denis Trystram. A study of scheduling problems with preemptions on multi-core computers with gpu accelerators. *Discrete Applied Mathematics*, 196:72–82, 2015.
- [3] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.
- [4] Chandan Hks and Sonika P. Reddy. Power efficient distribution and execution of tasks upon hardware fault with multiple processors, April 4 2017. US Patent 9,612,907.
- [5] Intel. Enhanced intel® speedstep® technology for the intel® pentium® m processor. 2004.
- [6] Ettikan K. Karuppiah, Yong Keh Kok, and Keeratpal Singh. A middleware framework for programmable multi-gpu-based big data applications. In *GPU Computing and Applications*, pages 187–206. Springer, 2015.
- [7] Mihaela Mitici, Martijn Onderwater, Maurits de Graaf, Jan-Kees van Ommeren, Nico van Dijk, Jasper Goseling, and Richard J. Boucherie. Optimal query assignment for wireless sensor networks. *Int. Journal of Electron. and Commun. (AEÜ)*, 69(8):1102–1112, 2015.
- [8] Cristóbal A. Navarro, Nancy Hitschfeld-Kahler, and Luis Mateu. A survey on parallel computing and its applications in data-parallel problems using gpu architectures. *Communications in Computational Physics*, 15(2):285–329, 2014.
- [9] Martijn Onderwater, Sandjai Bhulai, and Rob van der Mei. On the control of a queueing system with aging state information. *Stochastic Models*, 31(4):588–617, 2015.
- [10] Martin L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.
- [11] Koichi Shirahata, Hitoshi Sato, and Satoshi Matsuoka. Hybrid map task scheduling for gpu-based heterogeneous clusters. In *2010 IEEE Second International Conference on Cloud Computing Technology and Science*, pages 733–740. IEEE, 2010.
- [12] Baoxue Zhao, Jianlong Zhong, Bingsheng He, Qiong Luo, Wenbin Fang, and Naga K. Govindaraju. Gpu-accelerated cloud computing for data-intensive applications. In *Cloud Computing for Data-Intensive Applications*, pages 105–129. Springer, 2014.

A Bepalen van $\mathbf{V}^\alpha(i, j)$ en $\mathbf{g}^\alpha$

Om een oplossing te vinden voor Vergelijking (6), die niet recursief is, moet er een aantal stappen doorlopen worden. Aangezien de uitwerking van deze stappen nogal uitgebreid zijn, staan ze in deze appendix. In het verslag zelf wordt slechts alleen gebruik gemaakt van de einduitkomsten uit deze appendix, welke nog wel geverifieerd worden.

A.1 Het oplossen van de verschilvergelijking $\Delta_1^2 \mathbf{V}^\alpha(i, j)$

Aangezien verschilvergelijkingen aan de randen ($i = 0$ en $j = 0$) afwijken, wordt er eerst gekeken naar het gebied zonder deze randen ($i, j > 0$). Definieer: $\Delta_1 \mathbf{V}^\alpha(i, j) = \mathbf{V}^\alpha(i+1, j) - \mathbf{V}^\alpha(i, j)$, dan geldt voor $i > 0$ en $j > 0$:

$$\Delta_1 \mathbf{V}^\alpha(i, j) = \begin{cases} \gamma_1 + \lambda_3^* [\alpha \Delta_1 \mathbf{V}^\alpha(i+1, j) + (1-\alpha) \Delta_1 \mathbf{V}^\alpha(i, j+1)] \\ + \mu_1^* \Delta_1 \mathbf{V}^\alpha(i-1, j) \\ + \mu_2^* \Delta_1 \mathbf{V}^\alpha(i, j-1) \\ + [1 - (\lambda_3^* + \mu_1^* + \mu_2^*)] \Delta_1 \mathbf{V}^\alpha(i, j) \end{cases} \quad (16)$$

Vervolgens wordt het volgende gedefinieerd: $\Delta_1^2 \mathbf{V}^\alpha(i, j) = \Delta_1 \mathbf{V}^\alpha(i+1, j) - \Delta_1 \mathbf{V}^\alpha(i, j)$. Met deze definitie geldt dan het volgende voor de situatie $i > 0$ en $j > 0$:

$$\Delta_1^2 \mathbf{V}^\alpha(i, j) = \begin{cases} \lambda_3^* [\alpha \Delta_1^2 \mathbf{V}^\alpha(i+1, j) + (1-\alpha) \Delta_1^2 \mathbf{V}^\alpha(i, j+1)] \\ + \mu_1^* \Delta_1^2 \mathbf{V}^\alpha(i-1, j) \\ + \mu_2^* \Delta_1^2 \mathbf{V}^\alpha(i, j-1) \\ + [1 - (\lambda_3^* + \mu_1^* + \mu_2^*)] \Delta_1^2 \mathbf{V}^\alpha(i, j) \end{cases}$$

Herschrijven levert het volgende op:

$$\begin{aligned} (\lambda_3^* + \mu_1^* + \mu_2^*) \Delta_1^2 \mathbf{V}^\alpha(i, j) &= \begin{cases} \lambda_3^* [\alpha \Delta_1^2 \mathbf{V}^\alpha(i+1, j) + (1-\alpha) \Delta_1^2 \mathbf{V}^\alpha(i, j+1)] \\ + \mu_1^* \Delta_1^2 \mathbf{V}^\alpha(i-1, j) + \mu_2^* \Delta_1^2 \mathbf{V}^\alpha(i, j-1) \end{cases} \\ [\lambda_3^* \alpha + \mu_1^*] \Delta_1^2 \mathbf{V}^\alpha(i, j) + [\lambda_3^* (1-\alpha) + \mu_2^*] \Delta_1^2 \mathbf{V}^\alpha(i, j) &= \begin{cases} \lambda_3^* \alpha \Delta_1^2 \mathbf{V}^\alpha(i+1, j) \\ + \lambda_3^* (1-\alpha) \Delta_1^2 \mathbf{V}^\alpha(i, j+1) \\ + \mu_1^* \Delta_1^2 \mathbf{V}^\alpha(i-1, j) \\ + \mu_2^* \Delta_1^2 \mathbf{V}^\alpha(i, j-1) \end{cases} \end{aligned} \quad (17)$$

Deze notatie wekt de suggestie dat de oplossing van deze vergelijking kan worden opgesplitst in een deel dat alleen afhankelijk is van i , en een deel dat alleen afhankelijk is van j . Zodat de oplossing kan worden gegeven in de vorm: $\Delta_1^2 \mathbf{V}^\alpha(i, j) = \mathbf{V}_1^\alpha(i) + \mathbf{V}_2^\alpha(j)$ en dat daarbij $\mathbf{V}_1^\alpha(i)$ en $\mathbf{V}_2^\alpha(j)$ voldoen aan:

$$\begin{cases} [\lambda_3^* \alpha + \mu_1^*] \mathbf{V}_1^\alpha(i) &= \lambda_3^* \alpha \mathbf{V}_1^\alpha(i+1) + \mu_1^* \mathbf{V}_1^\alpha(i-1) \\ [\lambda_3^* (1-\alpha) + \mu_2^*] \mathbf{V}_2^\alpha(j) &= \lambda_3^* (1-\alpha) \mathbf{V}_2^\alpha(j+1) + \mu_2^* \mathbf{V}_2^\alpha(j-1) \end{cases} \quad (18)$$

Deze vergelijkingen zijn simpele homogene verschilvergelijkingen, waarvan de oplossingen gegeven wordt door:

$$\begin{cases} \mathbf{V}_1^\alpha(i) = \frac{\mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)] \left(\frac{\mu_1^*}{\lambda_3^* \alpha}\right)^i}{\mu_1^* - \lambda_3^* \alpha} \\ \mathbf{V}_2^\alpha(j) = \frac{\mu_2^* \mathbf{V}_2^\alpha(0) - \lambda_3^* (1-\alpha) \mathbf{V}_2^\alpha(1)}{\mu_2^* - \lambda_3^* (1-\alpha)} + \frac{\lambda_3^* (1-\alpha) [\mathbf{V}_2^\alpha(1) - \mathbf{V}_2^\alpha(0)] \left(\frac{\mu_2^*}{\lambda_3^* (1-\alpha)}\right)^j}{\mu_2^* - \lambda_3^* (1-\alpha)} \end{cases} \quad (19)$$

Lemma A.1. De gevonden $\mathbf{V}_1^\alpha(i)$ en $\mathbf{V}_2^\alpha(j)$ in Vergelijking (19) zijn een oplossing voor de vergelijkingen in (18), en met $\Delta_1^2 \mathbf{V}^\alpha(i, j) = \mathbf{V}_1^\alpha(i) + \mathbf{V}_2^\alpha(j)$, is het ook een oplossing voor Vergelijking (17).

Bewijs : Zie Appendix B.

Q.E.D.

De waarden van $\mathbf{V}_1^\alpha(0)$, $\mathbf{V}_1^\alpha(1)$, $\mathbf{V}_2^\alpha(0)$ en $\mathbf{V}_2^\alpha(1)$ moeten hierbij nog onderzocht worden, om de oplossing kloppend te maken voor de grenswaarden. Met dit doel wordt er nu eerst gekeken naar de grenswaarden van $j = 0$. In dat geval geldt voor $i > 0$

$$\begin{aligned} \Delta_1 \mathbf{V}^\alpha(i, 0) &= \begin{cases} \gamma_1 + \lambda_3^* [\alpha \Delta_1 \mathbf{V}^\alpha(i+1, 0) + (1-\alpha) \Delta_1 \mathbf{V}^\alpha(i, 1)] \\ + \mu_1^* \Delta_1 \mathbf{V}^\alpha(i-1, 0) \\ + [1 - (\lambda_3^* + \mu_1^*)] \Delta_1 \mathbf{V}^\alpha(i, 0) \end{cases} \\ \Delta_1^2 \mathbf{V}^\alpha(i, 0) &= \begin{cases} \lambda_3^* [\alpha \Delta_1^2 \mathbf{V}^\alpha(i+1, 0) + (1-\alpha) \Delta_1^2 \mathbf{V}^\alpha(i, 1)] \\ + \mu_1^* \Delta_1^2 \mathbf{V}^\alpha(i-1, 0) \\ + [1 - (\lambda_3^* + \mu_1^*)] \Delta_1^2 \mathbf{V}^\alpha(i, 0) \end{cases} \end{aligned}$$

Herschrijven levert:

$$(\lambda_3^* + \mu_1^*) \Delta_1^2 \mathbf{V}^\alpha(i, 0) = \begin{cases} \lambda_3^* [\alpha \Delta_1^2 \mathbf{V}^\alpha(i+1, 0) + (1-\alpha) \Delta_1^2 \mathbf{V}^\alpha(i, 1)] \\ + \mu_1^* \Delta_1^2 \mathbf{V}^\alpha(i-1, 0) \end{cases}$$

Wat leidt tot het volgende systeem aan vergelijkingen:

$$\begin{cases} [\lambda_3^* \alpha + \mu_1^*] \mathbf{V}_1^\alpha(i) = \lambda_3^* \alpha \mathbf{V}_1^\alpha(i+1) + \mu_1^* \mathbf{V}_1^\alpha(i-1) \\ \lambda_3^* (1-\alpha) \mathbf{V}_2^\alpha(0) = \lambda_3^* (1-\alpha) \mathbf{V}_2^\alpha(1) \end{cases}$$

In deze vergelijkingen is te zien dat, het Markov Beslissingsprobleem op de grens van $j = 0$, zich precies hetzelfde gedraagt als zijnde niet op de grens. Verder is te zien dat $\mathbf{V}_2^\alpha(0) = \mathbf{V}_2^\alpha(1)$, en dat dus de $\mathbf{V}_2^\alpha(j)$ in Vergelijking (19) een constante ($\mathbf{V}_2^\alpha(j) = c_2$) is. Zonder verlies van algemeenheid kan deze constante zo gekozen worden dat $\Delta_1^2 \mathbf{V}^\alpha(i, j)$ volledig verklaard wordt door alleen $\mathbf{V}_1^\alpha(i)$, namelijk door te kiezen voor $c_2 = 0$. Hierdoor wordt $\Delta_1^2 \mathbf{V}^\alpha(i, j)$

$$\Delta_1^2 \mathbf{V}^\alpha(i, j) = \frac{\mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)] \left(\frac{\mu_1^*}{\lambda_3^* \alpha}\right)^i}{\mu_1^* - \lambda_3^* \alpha}$$

A.2 Analyse van $\Delta_1 \mathbf{V}^\alpha(i, j+1) - \Delta_1 \mathbf{V}^\alpha(i, j)$

In Vergelijking (16) was $\Delta_1 \mathbf{V}^\alpha(i, j)$, voor $i > 0$ en $j > 0$, als volgt:

$$\Delta_1 \mathbf{V}^\alpha(i, j) = \begin{cases} \gamma_1 + \lambda_3^* [\alpha \Delta_1 \mathbf{V}^\alpha(i+1, j) + (1-\alpha) \Delta_1 \mathbf{V}^\alpha(i, j+1)] \\ + \mu_1^* \Delta_1 \mathbf{V}^\alpha(i-1, j) \\ + \mu_2^* \Delta_1 \mathbf{V}^\alpha(i, j-1) \\ + [1 - (\lambda_3^* + \mu_1^* + \mu_2^*)] \Delta_1 \mathbf{V}^\alpha(i, j) \end{cases}$$

Definieer nu: $\Delta_2 \Delta_1 \mathbf{V}^\alpha(i, j) := \Delta_1 \mathbf{V}^\alpha(i, j+1) - \Delta_1 \mathbf{V}^\alpha(i, j)$. Hieruit volgt dat voor $i > 0$ en $j > 0$:

$$\Delta_2 \Delta_1 \mathbf{V}^\alpha(i, j) = \begin{cases} \lambda_3^* [\alpha \Delta_2 \Delta_1 \mathbf{V}^\alpha(i+1, j) + (1-\alpha) \Delta_2 \Delta_1 \mathbf{V}^\alpha(i, j+1)] \\ + \mu_1^* \Delta_2 \Delta_1 \mathbf{V}^\alpha(i-1, j) \\ + \mu_2^* \Delta_2 \Delta_1 \mathbf{V}^\alpha(i, j-1) \\ + [1 - (\lambda_3^* + \mu_1^* + \mu_2^*)] \Delta_2 \Delta_1 \mathbf{V}^\alpha(i, j) \end{cases}$$

Wegens dezelfde reden en op dezelfde wijze als hiervoor, wordt ook $\Delta_2 \Delta_1 \mathbf{V}^\alpha(i, j)$ opgesplitst in: $\Delta_2 \Delta_1 \mathbf{V}^\alpha(i, j) = \tilde{\mathbf{V}}_1^\alpha(i) + \tilde{\mathbf{V}}_2^\alpha(j)$, waarbij

$$\begin{cases} \tilde{\mathbf{V}}_1^\alpha(i) = \frac{\mu_1^* \tilde{\mathbf{V}}_1^\alpha(0) - \lambda_3^* \alpha \tilde{\mathbf{V}}_1^\alpha(1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\lambda_3^* \alpha [\tilde{\mathbf{V}}_1^\alpha(1) - \tilde{\mathbf{V}}_1^\alpha(0)] \left(\frac{\mu_1^*}{\lambda_3^* \alpha}\right)^i}{\mu_1^* - \lambda_3^* \alpha} \\ \tilde{\mathbf{V}}_2^\alpha(j) = \frac{\mu_2^* \tilde{\mathbf{V}}_2^\alpha(0) - \lambda_3^* (1-\alpha) \tilde{\mathbf{V}}_2^\alpha(1)}{\mu_2^* - \lambda_3^* (1-\alpha)} + \frac{\lambda_3^* (1-\alpha) [\tilde{\mathbf{V}}_2^\alpha(1) - \tilde{\mathbf{V}}_2^\alpha(0)] \left(\frac{\mu_2^*}{\lambda_3^* (1-\alpha)}\right)^j}{\mu_2^* - \lambda_3^* (1-\alpha)} \end{cases}$$

Ook hier moeten de waarden $\tilde{\mathbf{V}}_1^\alpha(0)$, $\tilde{\mathbf{V}}_1^\alpha(1)$, $\tilde{\mathbf{V}}_2^\alpha(0)$ en $\tilde{\mathbf{V}}_2^\alpha(1)$ nog onderzocht worden, om het aan de randen kloppend te maken. Er wordt hierbij eerst gekeken naar: $\Delta_2 \Delta_1 \mathbf{V}^\alpha(i, 0)$.

$$\begin{aligned} \Delta_1 \mathbf{V}^\alpha(i, 1) &= \begin{cases} \gamma_1 + \lambda_3^* [\alpha \Delta_1 \mathbf{V}^\alpha(i+1, 1) + (1-\alpha) \Delta_1 \mathbf{V}^\alpha(i, 2)] \\ + \mu_1^* \Delta_1 \mathbf{V}^\alpha(i-1, 1) \\ + \mu_2^* \Delta_1 \mathbf{V}^\alpha(i, 0) \\ + [1 - (\lambda_3^* + \mu_1^* + \mu_2^*)] \Delta_1 \mathbf{V}^\alpha(i, 1) \end{cases} \\ \Delta_1 \mathbf{V}^\alpha(i, 0) &= \begin{cases} \gamma_1 + \lambda_3^* [\alpha \Delta_1 \mathbf{V}^\alpha(i+1, 0) + (1-\alpha) \Delta_1 \mathbf{V}^\alpha(i, 1)] \\ + \mu_1^* \Delta_1 \mathbf{V}^\alpha(i-1, 0) \\ + [1 - (\lambda_3^* + \mu_1^*)] \Delta_1 \mathbf{V}^\alpha(i, 0) \end{cases} \end{aligned}$$

Daaruit volgt:

$$\Delta_2 \Delta_1 \mathbf{V}^\alpha(i, 0) = \begin{cases} \lambda_3^* [\alpha \Delta_2 \Delta_1 \mathbf{V}^\alpha(i+1, 0) + (1-\alpha) \Delta_2 \Delta_1 \mathbf{V}^\alpha(i, 1)] \\ + \mu_1^* \Delta_2 \Delta_1 \mathbf{V}^\alpha(i-1, 0) - \mu_2^* \Delta_2 \Delta_1 \mathbf{V}^\alpha(i, 0) \\ + [1 - (\lambda_3^* + \mu_1^*)] \Delta_2 \Delta_1 \mathbf{V}^\alpha(i, 0) \end{cases}$$

$$[\lambda_3^* \alpha + \mu_1^*] \Delta_2 \Delta_1 \mathbf{V}^\alpha(i, 0) + [\lambda_3^* (1-\alpha) + \mu_2^*] \Delta_2 \Delta_1 \mathbf{V}^\alpha(i, 0) = \begin{cases} \lambda_3^* \alpha \Delta_2 \Delta_1 \mathbf{V}^\alpha(i+1, 0) \\ + \mu_1^* \Delta_2 \Delta_1 \mathbf{V}^\alpha(i-1, 0) \\ + \lambda_3^* (1-\alpha) \Delta_2 \Delta_1 \mathbf{V}^\alpha(i, 1) \end{cases}$$

Door opsplitsen ontstaan de volgende vergelijkingen:

$$\begin{cases} [\lambda_3^* \alpha + \mu_1^*] \Delta_2 \Delta_1 \mathbf{V}^\alpha(i, 0) &= \lambda_3^* \alpha \Delta_2 \Delta_1 \mathbf{V}^\alpha(i+1, 0) + \mu_1^* \Delta_2 \Delta_1 \mathbf{V}^\alpha(i-1, 0) \\ [\lambda_3^* (1-\alpha) + \mu_2^*] \Delta_2 \Delta_1 \mathbf{V}^\alpha(i, 0) &= \lambda_3^* (1-\alpha) \Delta_2 \Delta_1 \mathbf{V}^\alpha(i, 1) \end{cases}$$

Door substitutie van $\Delta_2 \Delta_1 \mathbf{V}^\alpha(i, j) = \tilde{\mathbf{V}}_1^\alpha(i) + \tilde{\mathbf{V}}_2^\alpha(j)$ in de laatste vergelijking ontstaat:

$$[\lambda_3^* (1-\alpha) + \mu_2^*] [\tilde{\mathbf{V}}_1^\alpha(i) + \tilde{\mathbf{V}}_2^\alpha(0)] = \lambda_3^* (1-\alpha) [\tilde{\mathbf{V}}_1^\alpha(i) + \tilde{\mathbf{V}}_2^\alpha(1)]$$

Herschrijven levert:

$$\mu_2^* \tilde{\mathbf{V}}_1^\alpha(i) = \lambda_3^* (1-\alpha) \tilde{\mathbf{V}}_2^\alpha(1) - [\lambda_3^* (1-\alpha) + \mu_2^*] \tilde{\mathbf{V}}_2^\alpha(0)$$

Hieruit volgt dat $\tilde{\mathbf{V}}_1^\alpha(i)$ een constante ($\tilde{c}_1$) is voor $i > 0$. Vervolgens wordt op dezelfde manier $\Delta_2 \Delta_1 \mathbf{V}^\alpha(0, j)$ onderzocht (zie Appendix: C voor de uitwerking) voor $j > 0$, en daaruit volgt dat $\tilde{\mathbf{V}}_2^\alpha(j)$ ook een constante ($\tilde{c}_2$) is. Dit samen reduceert de vorige vergelijking tot:

$$\mu_2^* \tilde{c}_1 = \lambda_3^* (1-\alpha) \tilde{c}_2 - [\lambda_3^* (1-\alpha) + \mu_2^*] \tilde{c}_2 = -\mu_2^* \tilde{c}_2$$

Hieruit volgt dat $\tilde{c}_1 = -\tilde{c}_2$, wat maakt dat: $\Delta_2 \Delta_1 \mathbf{V}^\alpha(i, j) = 0$.

A.3 Verschilvergelijking $\Delta_1 \mathbf{V}^\alpha(i, j)$ oplossen

Tot dusver is de volgende vergelijking gevonden voor $\Delta_1^2 \mathbf{V}^\alpha(i, j)$:

$$\Delta_1^2 \mathbf{V}^\alpha(i, j) = \frac{\mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)] \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^i}{\mu_1^* - \lambda_3^* \alpha} \quad (20)$$

Verder is vastgesteld dat $\Delta_2 \Delta_1 \mathbf{V}^\alpha(i, j) = 0$, zodat $\Delta_1 \mathbf{V}^\alpha(i, j)$ onafhankelijk is van j voor alle mogelijke waarden van i . Een andere expressie voor $\Delta_1 \mathbf{V}^\alpha(i, j)$ is:

$$\Delta_1 \mathbf{V}^\alpha(i, j) = \Delta_1 \mathbf{V}^\alpha(0, j) + \sum_{k=0}^{i-1} \Delta_1^2 \mathbf{V}^\alpha(k, j) \quad (21)$$

Hierbij is gebruik gemaakt van het feit dat $\Delta_1^2 \mathbf{V}^\alpha(i, j) = \Delta_1 \mathbf{V}^\alpha(i+1, j) - \Delta_1 \mathbf{V}^\alpha(i, j)$. Door het invullen van $\Delta_1^2 \mathbf{V}^\alpha(i, j)$ en het herschrijven van de somformule, ontstaat de volgende vergelijking:

$$\Delta_1 \mathbf{V}^\alpha(i, j) = \begin{cases} \Delta_1 \mathbf{V}^\alpha(0, j) + \left(\frac{\mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1)}{\mu_1^* - \lambda_3^* \alpha} \right) i \\ + \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)]}{\mu_1^* - \lambda_3^* \alpha} \frac{1 - \left[\frac{\mu_1^*}{\lambda_3^* \alpha} \right]^i}{1 - \frac{\mu_1^*}{\lambda_3^* \alpha}} \end{cases}$$

Aangezien $\Delta_1 \mathbf{V}^\alpha(i, j)$ onafhankelijk is van j voor alle mogelijke waarden van i , geldt dat $\Delta_1 \mathbf{V}^\alpha(0, j)$ een constante is. De waarde van deze constante wordt even verderop bepaald. Nu wordt eerst de zojuist gevonden vergelijking voor $\Delta_1 \mathbf{V}^\alpha(i, j)$ gesubstitueerd in Vergelijking (16).

$$\Delta_1 \mathbf{V}^\alpha(0, j) + \left(\frac{\mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1)}{\mu_1^* - \lambda_3^* \alpha} \right) i + \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)]}{\mu_1^* - \lambda_3^* \alpha} \frac{1 - \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^i}{1 - \frac{\mu_1^*}{\lambda_3^* \alpha}} =$$

$$\left\{ \begin{aligned} & \gamma_1 + \lambda_3^* \alpha \left[\Delta_1 \mathbf{V}^\alpha(0, j) + \left(\frac{\mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1)}{\mu_1^* - \lambda_3^* \alpha} \right) (i+1) + \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)]}{\mu_1^* - \lambda_3^* \alpha} \frac{1 - \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^{i+1}}{1 - \frac{\mu_1^*}{\lambda_3^* \alpha}} \right] \\ & + \mu_1^* \left[\Delta_1 \mathbf{V}^\alpha(0, j) + \left(\frac{\mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1)}{\mu_1^* - \lambda_3^* \alpha} \right) (i-1) + \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)]}{\mu_1^* - \lambda_3^* \alpha} \frac{1 - \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^{i-1}}{1 - \frac{\mu_1^*}{\lambda_3^* \alpha}} \right] \\ & + \lambda_3^* \alpha (1 - \alpha) \left[\Delta_1 \mathbf{V}^\alpha(0, j+1) + \left(\frac{\mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1)}{\mu_1^* - \lambda_3^* \alpha} \right) i + \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)]}{\mu_1^* - \lambda_3^* \alpha} \frac{1 - \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^i}{1 - \frac{\mu_1^*}{\lambda_3^* \alpha}} \right] \\ & + \mu_2^* \left[\Delta_1 \mathbf{V}^\alpha(0, j-1) + \left(\frac{\mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1)}{\mu_1^* - \lambda_3^* \alpha} \right) i + \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)]}{\mu_1^* - \lambda_3^* \alpha} \frac{1 - \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^i}{1 - \frac{\mu_1^*}{\lambda_3^* \alpha}} \right] \\ & + [1 - (\lambda_3^* + \mu_1^* + \mu_2^*)] \left[\Delta_1 \mathbf{V}^\alpha(0, j) + \left(\frac{\mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1)}{\mu_1^* - \lambda_3^* \alpha} \right) i + \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)]}{\mu_1^* - \lambda_3^* \alpha} \frac{1 - \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^i}{1 - \frac{\mu_1^*}{\lambda_3^* \alpha}} \right] \end{aligned} \right.$$

Herschrijven levert:

$$0 = \left\{ \begin{aligned} & \gamma_1 + \lambda_3^* \alpha \left[\left(\frac{\mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1)}{\mu_1^* - \lambda_3^* \alpha} \right) + \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)]}{\mu_1^* - \lambda_3^* \alpha} \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^i \right] \\ & - \mu_1^* \left[\left(\frac{\mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1)}{\mu_1^* - \lambda_3^* \alpha} \right) + \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)]}{\mu_1^* - \lambda_3^* \alpha} \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^{i-1} \right] \\ & + \lambda_3^* \alpha (1 - \alpha) [\Delta_1 \mathbf{V}^\alpha(0, j+1) - \Delta_1 \mathbf{V}^\alpha(0, j)] \\ & + \mu_2^* [\Delta_1 \mathbf{V}^\alpha(0, j-1) - \Delta_1 \mathbf{V}^\alpha(0, j)] \end{aligned} \right.$$

Aangezien $\Delta_2 \Delta_1 \mathbf{V}^\alpha(i, j) = 0$, kan het voorgaande stelsel van vergelijkingen verder vereenvoudigd worden tot:

$$\gamma_1 = \left\{ \begin{aligned} & \mu_1^* \left[\left(\frac{\mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1)}{\mu_1^* - \lambda_3^* \alpha} \right) + \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)]}{\mu_1^* - \lambda_3^* \alpha} \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^{i-1} \right] \\ & - \lambda_3^* \alpha \left[\left(\frac{\mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1)}{\mu_1^* - \lambda_3^* \alpha} \right) + \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)]}{\mu_1^* - \lambda_3^* \alpha} \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^i \right] \end{aligned} \right.$$

$$\gamma_1 = \left\{ \begin{aligned} & \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)]}{\mu_1^* - \lambda_3^* \alpha} \left[\mu_1^* \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^{i-1} - \lambda_3^* \alpha \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^i \right] \\ & + (\mu_1^* - \lambda_3^* \alpha) \left[\frac{\mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1)}{\mu_1^* - \lambda_3^* \alpha} \right] \end{aligned} \right.$$

$$\gamma_1 = \begin{cases} \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)]}{\mu_1^* - \lambda_3^* \alpha} \left[\mu_1^* \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^{i-1} - \lambda_3^* \alpha \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^i \right] \\ + \mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1) \end{cases}$$

Hieruit volgt dat:

$$\mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1) = \gamma_1$$

Deze vergelijking wordt gebruikt om $\mathbf{V}_1^\alpha(1)$ te substitueren in Vergelijking (20).

$$\Delta_1^2 \mathbf{V}^\alpha(i, j) = \frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} + \left[\mathbf{V}_1^\alpha(0) - \frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^i$$

Hierdoor wordt $\Delta_1 \mathbf{V}^\alpha(i, j)$ met behulp van Vergelijking (21):

$$\Delta_1 \mathbf{V}^\alpha(i, j) = \begin{cases} \Delta_1 \mathbf{V}^\alpha(0, j) + \left(\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right) i \\ + \left[\mathbf{V}_1^\alpha(0) - \frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] \frac{1 - \left[\frac{\mu_1^*}{\lambda_3^* \alpha} \right]^i}{1 - \frac{\mu_1^*}{\lambda_3^* \alpha}} \end{cases} \quad (22)$$

A.4 Waarde bepalen van constante $\Delta_1 \mathbf{V}^\alpha(0, j)$

Nu wordt de constante $\Delta_1 \mathbf{V}^\alpha(0, j)$ onderzocht, door naar de bijbehorende verschilvergelijking te kijken.

$$\Delta_1 \mathbf{V}^\alpha(0, j) = \begin{cases} \gamma_1 + \lambda_3^* [\alpha \Delta_1 \mathbf{V}^\alpha(1, j) + (1 - \alpha) \Delta_1 \mathbf{V}^\alpha(0, j + 1)] \\ + \mu_2^* \Delta_1 \mathbf{V}^\alpha(0, j - 1) \\ + [1 - (\lambda_3^* + \mu_1^* + \mu_2^*)] \Delta_1 \mathbf{V}^\alpha(0, j) \end{cases}$$

Dit omschrijven levert de volgende vergelijking op:

$$0 = \begin{cases} \gamma_1 + \lambda_3^* \alpha [\Delta_1 \mathbf{V}^\alpha(1, j) - \Delta_1 \mathbf{V}^\alpha(0, j)] \\ + \lambda_3^* (1 - \alpha) [\Delta_1 \mathbf{V}^\alpha(0, j + 1) - \Delta_1 \mathbf{V}^\alpha(0, j)] \\ - \mu_1^* \Delta_1 \mathbf{V}^\alpha(0, j) \\ + \mu_2^* [\Delta_1 \mathbf{V}^\alpha(0, j - 1) - \Delta_1 \mathbf{V}^\alpha(0, j)] \end{cases}$$

Aangezien $\Delta_2 \Delta_1 \mathbf{V}^\alpha(i, j) = 0$, kun je dit herschrijven tot:

$$0 = \gamma_1 + \lambda_3^* \alpha [\Delta_1 \mathbf{V}^\alpha(1, j) - \Delta_1 \mathbf{V}^\alpha(0, j)] - \mu_1^* \Delta_1 \mathbf{V}^\alpha(0, j)$$

Met Vergelijking (22) wordt verkregen dat: $\Delta_1 \mathbf{V}^\alpha(1, j) = \Delta_1 \mathbf{V}^\alpha(0, j) + \mathbf{V}_1^\alpha(0)$. Invullen geeft:

$$0 = \gamma_1 + \lambda_3^* \alpha \mathbf{V}_1^\alpha(0) - \mu_1^* \Delta_1 \mathbf{V}^\alpha(0, j)$$

Ofwel:

$$\Delta_1 \mathbf{V}^\alpha(0, j) = \frac{\lambda_3^* \alpha}{\mu_1^*} \mathbf{V}_1^\alpha(0) + \frac{\gamma_1}{\mu_1^*}$$

Deze uitkomst voor $\Delta_1 \mathbf{V}^\alpha(0, j)$ gesubstitueerd in Vergelijking (22) levert:

$$\Delta_1 \mathbf{V}^\alpha(i, j) = \begin{cases} \frac{\lambda_3^* \alpha}{\mu_1^*} \mathbf{V}_1^\alpha(0) + \frac{\gamma_1}{\mu_1^*} + \left(\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right) i \\ + \left[\mathbf{V}_1^\alpha(0) - \frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] \frac{1 - \left[\frac{\mu_1^*}{\lambda_3^* \alpha} \right]^i}{1 - \frac{\mu_1^*}{\lambda_3^* \alpha}} \end{cases} \quad (23)$$

A.5 Het afleiden van $\mathbf{V}^\alpha(i, j)$

Aangezien $\Delta_1 \mathbf{V}^\alpha(i, j) = \mathbf{V}^\alpha(i+1, j) - \mathbf{V}^\alpha(i, j)$ kan de volgende vergelijking opgesteld worden.

$$\mathbf{V}^\alpha(i, j) = \mathbf{V}^\alpha(0, j) + \sum_{k=0}^{i-1} \Delta_1 \mathbf{V}^\alpha(k, j)$$

Door het herschrijven van de somformule en gebruik te maken van Vergelijking (23) ontstaat:

$$\mathbf{V}^\alpha(i, j) = \begin{cases} \mathbf{V}^\alpha(0, j) + \left(\frac{\lambda_3^* \alpha}{\mu_1^*} \mathbf{V}_1^\alpha(0) + \frac{\gamma_1}{\mu_1^*} \right) i + \frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \frac{i(i-1)}{2} \\ + \frac{1}{1 - \frac{\mu_1^*}{\lambda_3^* \alpha}} \left[\mathbf{V}_1^\alpha(0) - \frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] \left[i - \frac{1 - \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^i}{1 - \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)} \right] \end{cases} \quad (24)$$

Voor een uniek antwoord is het belangrijk dat $\mathbf{V}^\alpha(i, j)$ niet exponentieel groeit. Daarom wordt de laatste constante $\mathbf{V}_1^\alpha(0)$ in Vergelijking (24) zo gekozen, dat de term $\left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^i$ verdwijnt.

$$\mathbf{V}_1^\alpha(0) = \frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha}$$

Dit gesubstitueerd in Vergelijking (24) geeft:

$$\mathbf{V}^\alpha(i, j) = \mathbf{V}^\alpha(0, j) + \left(\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right) i + \frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \frac{i(i-1)}{2}$$

Ofwel:

$$\mathbf{V}^\alpha(i, j) = \mathbf{V}^\alpha(0, j) + \frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \frac{i(i+1)}{2}$$

Door de stappen in appendix A nogmaals te doorlopen, en te kijken naar de verschilvergelijking voor j in plaats van voor i , dus $\Delta_2 \mathbf{V}^\alpha(i, j) = \mathbf{V}^\alpha(i, j+1) - \mathbf{V}^\alpha(i, j)$, wordt de volgende vergelijking voor $\mathbf{V}^\alpha(i, j)$ gevonden.

$$\mathbf{V}^\alpha(i, j) = \mathbf{V}^\alpha(i, 0) + \frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \frac{j(j+1)}{2}$$

Voor de uitwerking van de tussenstappen, zie Appendix: D. Deze uitkomsten samen maakt:

$$\mathbf{V}^\alpha(i, j) = \frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \frac{i(i+1)}{2} + \frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \frac{j(j+1)}{2} \quad (25)$$

A.6 Bepalen van constante $\mathbf{g}^\alpha$

Invullen in Vergelijking (6) geeft:

$$\mathbf{g}^\alpha + \frac{1}{2} \left[\frac{\gamma_1 i(i+1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\gamma_2 j(j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] = \begin{cases} \gamma_1 i + \gamma_2 j + \lambda_3^* \alpha \left(\frac{1}{2} \left[\frac{\gamma_1 (i+2)(i+1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\gamma_2 j(j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \right) \\ + \lambda_3^* (1 - \alpha) \left(\frac{1}{2} \left[\frac{\gamma_1 i(i+1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\gamma_2 (j+2)(j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \right) \\ + \mu_1^* \left(\frac{1}{2} \left[\frac{\gamma_1 i(i-1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\gamma_2 j(j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \right) \mathbb{1}_{i>0} \\ + \mu_2^* \left(\frac{1}{2} \left[\frac{\gamma_1 i(i+1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\gamma_2 j(j-1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \right) \mathbb{1}_{j>0} \\ + [1 - \lambda_3^*] \left(\frac{1}{2} \left[\frac{\gamma_1 i(i+1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\gamma_2 j(j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \right) \\ - \mu_1^* \mathbb{1}_{i>0} \left(\frac{1}{2} \left[\frac{\gamma_1 i(i+1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\gamma_2 j(j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \right) \\ - \mu_2^* \mathbb{1}_{j>0} \left(\frac{1}{2} \left[\frac{\gamma_1 i(i+1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\gamma_2 j(j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \right) \end{cases}$$

$$\mathbf{g}^\alpha = \begin{cases} \gamma_1 i + \gamma_2 j + \lambda_3^* \alpha \left(\frac{1}{2} \left[\frac{(i+2)(i+1)}{\mu_1^* - \lambda_3^* \alpha} - \frac{i(i+1)}{\mu_1^* - \lambda_3^* \alpha} \right] \right) \\ + \lambda_3^* (1 - \alpha) \left(\frac{\gamma_2}{2} \left[\frac{(j+2)(j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} - \frac{j(j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \right) \\ + \mu_1^* \left(\frac{\gamma_1}{2} \left[\frac{i(i-1)}{\mu_1^* - \lambda_3^* \alpha} - \frac{i(i+1)}{\mu_1^* - \lambda_3^* \alpha} \right] \right) \mathbb{1}_{i>0} \\ + \mu_2^* \left(\frac{\gamma_2}{2} \left[\frac{j(j-1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} - \frac{j(j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \right) \mathbb{1}_{j>0} \end{cases}$$

$$\mathbf{g}^\alpha = \begin{cases} \gamma_1 i + \gamma_2 j + \lambda_3^* \alpha \left(\frac{\gamma_1 (i+1)}{\mu_1^* - \lambda_3^* \alpha} \right) + \lambda_3^* (1 - \alpha) \left(\frac{\gamma_2 (j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right) \\ - \mu_1^* \left(\frac{\gamma_1 i}{\mu_1^* - \lambda_3^* \alpha} \right) \mathbb{1}_{i>0} - \mu_2^* \left(\frac{\gamma_2 j}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right) \mathbb{1}_{j>0} \end{cases}$$

Lemma A.2. Voor $i \geq 0$ en $j \geq 0$ geldt dat:

$$\mathbf{g}^\alpha = \lambda_3^* \alpha \left[\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] + \lambda_3^* (1 - \alpha) \left[\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right]$$

Bewijs : Zie Appendix E.

Q.E.D.

B Een oplossing voor de vergelijkingen.

De vergelijkingen in (19) zouden een oplossing zijn en zien er als volgt uit.

$$\begin{cases} \mathbf{V}_1^\alpha(i) = \frac{\mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)] \left(\frac{\mu_1^*}{\lambda_3^* \alpha}\right)^i}{\mu_1^* - \lambda_3^* \alpha} \\ \mathbf{V}_2^\alpha(j) = \frac{\mu_2^* \mathbf{V}_2^\alpha(0) - \lambda_3^* (1-\alpha) \mathbf{V}_2^\alpha(1)}{\mu_2^* - \lambda_3^* (1-\alpha)} + \frac{\lambda_3^* (1-\alpha) [\mathbf{V}_2^\alpha(1) - \mathbf{V}_2^\alpha(0)] \left(\frac{\mu_2^*}{\lambda_3^* (1-\alpha)}\right)^j}{\mu_2^* - \lambda_3^* (1-\alpha)} \end{cases}$$

B.1 Een oplossing voor de deelvergelijkingen.

Lemma B.1 (Lemma A.1a). *De gevonden $\mathbf{V}_1^\alpha(i)$ en $\mathbf{V}_2^\alpha(j)$ in Vergelijking (19) zijn een oplossing voor de volgende vergelijkingen. Deze vergelijkingen staan tevens in Vergelijking (18).*

$$\begin{cases} [\lambda_3^* \alpha + \mu_1^*] \mathbf{V}_1^\alpha(i) &= \lambda_3^* \alpha \mathbf{V}_1^\alpha(i+1) + \mu_1^* \mathbf{V}_1^\alpha(i-1) \\ [\lambda_3^* (1-\alpha) + \mu_2^*] \mathbf{V}_2^\alpha(j) &= \lambda_3^* (1-\alpha) \mathbf{V}_2^\alpha(j+1) + \mu_2^* \mathbf{V}_2^\alpha(j-1) \end{cases}$$

Bewijs : Om te bewijzen dat dit inderdaad een oplossing is, wordt er substitutie toegepast. Door substitutie toe te passen van $\mathbf{V}_1^\alpha(i)$ in de eerste deelvergelijking ontstaat de volgende vergelijking.

$$\begin{aligned} [\lambda_3^* \alpha + \mu_1^*] \left(\frac{\mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)] \left(\frac{\mu_1^*}{\lambda_3^* \alpha}\right)^i}{\mu_1^* - \lambda_3^* \alpha} \right) = \\ \left\{ \begin{aligned} &\lambda_3^* \alpha \left(\frac{\mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)] \left(\frac{\mu_1^*}{\lambda_3^* \alpha}\right)^{i+1}}{\mu_1^* - \lambda_3^* \alpha} \right) \\ &+ \mu_1^* \left(\frac{\mu_1^* \mathbf{V}_1^\alpha(0) - \lambda_3^* \alpha \mathbf{V}_1^\alpha(1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)] \left(\frac{\mu_1^*}{\lambda_3^* \alpha}\right)^{i-1}}{\mu_1^* - \lambda_3^* \alpha} \right) \end{aligned} \right\} \end{aligned}$$

Herschrijven levert het volgende op.

$$\begin{aligned} [\lambda_3^* \alpha + \mu_1^*] \left(\frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)] \left(\frac{\mu_1^*}{\lambda_3^* \alpha}\right)^i}{\mu_1^* - \lambda_3^* \alpha} \right) &= \begin{cases} \lambda_3^* \alpha \left(\frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)] \left(\frac{\mu_1^*}{\lambda_3^* \alpha}\right)^{i+1}}{\mu_1^* - \lambda_3^* \alpha} \right) \\ + \mu_1^* \left(\frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)] \left(\frac{\mu_1^*}{\lambda_3^* \alpha}\right)^{i-1}}{\mu_1^* - \lambda_3^* \alpha} \right) \end{cases} \\ [\lambda_3^* \alpha + \mu_1^*] \left(\frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)] \left(\frac{\mu_1^*}{\lambda_3^* \alpha}\right)^i}{\mu_1^* - \lambda_3^* \alpha} \right) &= \begin{cases} \lambda_3^* \alpha \left(\left(\frac{\mu_1^*}{\lambda_3^* \alpha}\right) \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)] \left(\frac{\mu_1^*}{\lambda_3^* \alpha}\right)^i}{\mu_1^* - \lambda_3^* \alpha} \right) \\ + \mu_1^* \left(\left(\frac{\lambda_3^* \alpha}{\mu_1^*}\right) \frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)] \left(\frac{\mu_1^*}{\lambda_3^* \alpha}\right)^i}{\mu_1^* - \lambda_3^* \alpha} \right) \end{cases} \end{aligned}$$

$$[\lambda_3^* \alpha + \mu_1^*] \left(\frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)] \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^i}{\mu_1^* - \lambda_3^* \alpha} \right) = \begin{cases} \mu_1^* \left(\frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)] \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^i}{\mu_1^* - \lambda_3^* \alpha} \right) \\ + \lambda_3^* \alpha \left(\frac{\lambda_3^* \alpha [\mathbf{V}_1^\alpha(1) - \mathbf{V}_1^\alpha(0)] \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^i}{\mu_1^* - \lambda_3^* \alpha} \right) \end{cases}$$

Dit is dus inderdaad een oplossing voor de eerste deelvergelijking. Voor de tweede deelvergelijking wordt $\mathbf{V}_2^\alpha(j)$ gebruikt in plaats van $\mathbf{V}_1^\alpha(i)$. Deze is op dezelfde manier opgebouwd, alleen wordt er $(1 - \alpha)$ gebruikt in plaats van α , μ_2^* in plaats van μ_1^* en j in plaats van i . Dit geldt niet alleen voor de oplossing van $\mathbf{V}_2^\alpha(j)$ gegeven in (19), maar ook voor de tweede deelvergelijking in (18). De tweede deelvergelijking kan hierdoor op precies dezelfde wijze worden opgelost als de eerste, en zal daarbij ook tot hetzelfde resultaat komen. Hieruit volgt dus dat de $\mathbf{V}_1^\alpha(i)$ en de $\mathbf{V}_2^\alpha(j)$ in (19) inderdaad een oplossing is voor de deelvergelijkingen in (18).

Q.E.D.

B.2 Een oplossing voor de oorspronkelijke vergelijking

Lemma B.2 (Lemma A.1b). *De gevonden $\mathbf{V}_1^\alpha(i)$ en $\mathbf{V}_2^\alpha(j)$ in Vergelijking (19) zijn een oplossing voor Vergelijking (17), waarbij geldt dat $\Delta_1^2 \mathbf{V}^\alpha(i, j) = \mathbf{V}_1^\alpha(i) + \mathbf{V}_2^\alpha(j)$. Vergelijking (17) ziet er als volgt uit.*

$$[\lambda_3^* \alpha + \mu_1^*] \Delta_1^2 \mathbf{V}^\alpha(i, j) + [\lambda_3^* (1 - \alpha) + \mu_2^*] \Delta_1^2 \mathbf{V}^\alpha(i, j) = \begin{cases} \lambda_3^* \alpha \Delta_1^2 \mathbf{V}^\alpha(i + 1, j) \\ + \lambda_3^* (1 - \alpha) \Delta_1^2 \mathbf{V}^\alpha(i, j + 1) \\ + \mu_1^* \Delta_1^2 \mathbf{V}^\alpha(i - 1, j) \\ + \mu_2^* \Delta_1^2 \mathbf{V}^\alpha(i, j - 1) \end{cases}$$

Bewijs : Om dit te kunnen bewijzen, is allereerst nodig dat: $\Delta_1^2 \mathbf{V}^\alpha(i, j) = \mathbf{V}_1^\alpha(i) + \mathbf{V}_2^\alpha(j)$. Als dit ingevuld wordt in Vergelijking (17), dan krijg je:

$$[\lambda_3^* \alpha + \mu_1^*] (\mathbf{V}_1^\alpha(i) + \mathbf{V}_2^\alpha(j)) + [\lambda_3^* (1 - \alpha) + \mu_2^*] (\mathbf{V}_1^\alpha(i) + \mathbf{V}_2^\alpha(j)) = \begin{cases} \lambda_3^* \alpha [\mathbf{V}_1^\alpha(i + 1) + \mathbf{V}_2^\alpha(j)] \\ + \lambda_3^* (1 - \alpha) [\mathbf{V}_1^\alpha(i) + \mathbf{V}_2^\alpha(j + 1)] \\ + \mu_1^* [\mathbf{V}_1^\alpha(i - 1) + \mathbf{V}_2^\alpha(j)] \\ + \mu_2^* [\mathbf{V}_1^\alpha(i) + \mathbf{V}_2^\alpha(j - 1)] \end{cases}$$

Herschrijven levert:

$$[\lambda_3^* \alpha + \mu_1^*] \mathbf{V}_1^\alpha(i) + [\lambda_3^* (1 - \alpha) + \mu_2^*] \mathbf{V}_2^\alpha(j) = \begin{cases} \lambda_3^* \alpha \mathbf{V}_1^\alpha(i + 1) + \mu_1^* \mathbf{V}_1^\alpha(i - 1) \\ + \lambda_3^* (1 - \alpha) \mathbf{V}_2^\alpha(j + 1) + \mu_2^* \mathbf{V}_2^\alpha(j - 1) \end{cases}$$

Dit zijn precies de Vergelijkingen in (18). En aangezien (19) een oplossing daarvan is (zie Appendix: B.1), is het ook een oplossing van Vergelijking (17).

Q.E.D.

C Berekening voor $\Delta_2 \Delta_1 \mathbf{V}^\alpha(0, j)$

$$\Delta_1 \mathbf{V}^\alpha(0, j+1) = \begin{cases} \gamma_1 + \lambda_3^* [\alpha \Delta_1 \mathbf{V}^\alpha(1, j+1) + (1-\alpha) \Delta_1 \mathbf{V}^\alpha(0, j+2)] \\ + \mu_2^* \Delta_1 \mathbf{V}^\alpha(0, j) \\ + [1 - (\lambda_3^* + \mu_1^* + \mu_2^*)] \Delta_1 \mathbf{V}^\alpha(0, j+1) \end{cases}$$

$$\Delta_1 \mathbf{V}^\alpha(0, j) = \begin{cases} \gamma_1 + \lambda_3^* [\alpha \Delta_1 \mathbf{V}^\alpha(1, j) + (1-\alpha) \Delta_1 \mathbf{V}^\alpha(0, j+1)] \\ + \mu_2^* \Delta_1 \mathbf{V}^\alpha(0, j-1) \\ + [1 - (\lambda_3^* + \mu_1^* + \mu_2^*)] \Delta_1 \mathbf{V}^\alpha(0, j) \end{cases}$$

Daaruit volgt dat:

$$\Delta_2 \Delta_1 \mathbf{V}^\alpha(0, j) = \begin{cases} \lambda_3^* [\alpha \Delta_2 \Delta_1 \mathbf{V}^\alpha(1, j) + (1-\alpha) \Delta_2 \Delta_1 \mathbf{V}^\alpha(0, j+1)] \\ + \mu_2^* \Delta_2 \Delta_1 \mathbf{V}^\alpha(0, j-1) \\ + [1 - (\lambda_3^* + \mu_1^* + \mu_2^*)] \Delta_2 \Delta_1 \mathbf{V}^\alpha(0, j) \end{cases}$$

Ofwel:

$$[\lambda_3^* \alpha + \mu_1^*] \Delta_2 \Delta_1 \mathbf{V}^\alpha(0, j) + [\lambda_3^* (1-\alpha) + \mu_2^*] \Delta_2 \Delta_1 \mathbf{V}^\alpha(0, j) = \begin{cases} \lambda_3^* \alpha \Delta_2 \Delta_1 \mathbf{V}^\alpha(1, j) \\ + \lambda_3^* (1-\alpha) \Delta_2 \Delta_1 \mathbf{V}^\alpha(0, j+1) \\ + \mu_2^* \Delta_2 \Delta_1 \mathbf{V}^\alpha(0, j-1) \end{cases}$$

Door opsplitsen ontstaan de volgende deelvergelijkingen.

$$\begin{cases} [\lambda_3^* \alpha + \mu_1^*] \Delta_2 \Delta_1 \mathbf{V}^\alpha(0, j) & = \lambda_3^* \alpha \Delta_2 \Delta_1 \mathbf{V}^\alpha(1, j) \\ [\lambda_3^* (1-\alpha) + \mu_2^*] \Delta_2 \Delta_1 \mathbf{V}^\alpha(0, j) & = \lambda_3^* (1-\alpha) \Delta_2 \Delta_1 \mathbf{V}^\alpha(0, j+1) + \mu_2^* \Delta_2 \Delta_1 \mathbf{V}^\alpha(0, j-1) \end{cases}$$

Door substitutie van $\Delta_2 \Delta_1 \mathbf{V}^\alpha(i, j) = \tilde{\mathbf{V}}_1^\alpha(i) + \tilde{\mathbf{V}}_2^\alpha(j)$ in de eerste vergelijking ontstaat:

$$[\lambda_3^* \alpha + \mu_1^*] [\tilde{\mathbf{V}}_1^\alpha(0) + \tilde{\mathbf{V}}_2^\alpha(j)] = \lambda_3^* \alpha [\tilde{\mathbf{V}}_1^\alpha(1) + \tilde{\mathbf{V}}_2^\alpha(j)]$$

Herschrijven levert:

$$\mu_1^* \tilde{\mathbf{V}}_2^\alpha(j) = \lambda_3^* \alpha \tilde{\mathbf{V}}_1^\alpha(1) - [\lambda_3^* \alpha + \mu_1^*] \tilde{\mathbf{V}}_1^\alpha(0)$$

Hieruit volgt dat $\tilde{\mathbf{V}}_2^\alpha(j)$ een constante ($\tilde{c}_2$) is voor $j > 0$.

D Het oplossen van de verschilvergelijking $\Delta_2 \mathbf{V}^\alpha(i, j)$

D.1 Het oplossen van de verschilvergelijking $\Delta_2^1 \mathbf{V}^\alpha(i, j)$

Definieer: $\Delta_2 \mathbf{V}^\alpha(i, j) = \mathbf{V}^\alpha(i, j+1) - \mathbf{V}^\alpha(i, j)$, dan geldt voor $i > 0$ en $j > 0$:

$$\Delta_2 \mathbf{V}^\alpha(i, j) = \begin{cases} \gamma_2 + \lambda_3^* [\alpha \Delta_2 \mathbf{V}^\alpha(i+1, j) + (1-\alpha) \Delta_2 \mathbf{V}^\alpha(i, j+1)] \\ + \mu_1^* \Delta_2 \mathbf{V}^\alpha(i-1, j) \\ + \mu_2^* \Delta_2 \mathbf{V}^\alpha(i, j-1) \\ + [1 - (\lambda_3^* + \mu_1^* + \mu_2^*)] \Delta_2 \mathbf{V}^\alpha(i, j) \end{cases} \quad (26)$$

Vervolgens wordt gedefinieerd: $\Delta_2^2 \mathbf{V}^\alpha(i, j) = \Delta_2 \mathbf{V}^\alpha(i, j+1) - \Delta_2 \mathbf{V}^\alpha(i, j)$. Met deze definitie geldt voor $i > 0$ en $j > 0$ dat:

$$\Delta_2^2 \mathbf{V}^\alpha(i, j) = \begin{cases} \lambda_3^* [\alpha \Delta_2^2 \mathbf{V}^\alpha(i+1, j) + (1-\alpha) \Delta_2^2 \mathbf{V}^\alpha(i, j+1)] \\ + \mu_1^* \Delta_2^2 \mathbf{V}^\alpha(i-1, j) \\ + \mu_2^* \Delta_2^2 \mathbf{V}^\alpha(i, j-1) \\ + [1 - (\lambda_3^* + \mu_1^* + \mu_2^*)] \Delta_2^2 \mathbf{V}^\alpha(i, j) \end{cases}$$

Herschrijven levert het volgende op:

$$[\lambda_3^* \alpha + \mu_1^*] \Delta_2^2 \mathbf{V}^\alpha(i, j) + [\lambda_3^* (1-\alpha) + \mu_2^*] \Delta_2^2 \mathbf{V}^\alpha(i, j) = \begin{cases} \lambda_3^* \alpha \Delta_2^2 \mathbf{V}^\alpha(i+1, j) \\ + \lambda_3^* (1-\alpha) \Delta_2^2 \mathbf{V}^\alpha(i, j+1) \\ + \mu_1^* \Delta_2^2 \mathbf{V}^\alpha(i-1, j) \\ + \mu_2^* \Delta_2^2 \mathbf{V}^\alpha(i, j-1) \end{cases}$$

Deze notatie wekt de suggestie dat de oplossing van deze vergelijking kan worden opgesplitst in een deel dat alleen afhankelijk is van i , en een deel dat alleen afhankelijk is van j . Zodat de oplossing kan worden gegeven in de vorm: $\Delta_2^2 \mathbf{V}^\alpha(i, j) = \mathbf{V}_1(i) + \mathbf{V}_2(j)$. En dat daarbij $\mathbf{V}_1(i)$ en $\mathbf{V}_2(j)$ zouden voldoen aan:

$$\begin{cases} [\lambda_3^* \alpha + \mu_1^*] \mathbf{V}_1(i) & = \lambda_3^* \alpha \mathbf{V}_1(i+1) + \mu_1^* \mathbf{V}_1(i-1) \\ [\lambda_3^* (1-\alpha) + \mu_2^*] \mathbf{V}_2(j) & = \lambda_3^* (1-\alpha) \mathbf{V}_2(j+1) + \mu_2^* \mathbf{V}_2(j-1) \end{cases}$$

Deze simpele homogene verschilvergelijkingen hebben als oplossing:

$$\begin{cases} \mathbf{V}_1(i) = \frac{\mu_1^* \mathbf{V}_1(0) - \lambda_3^* \alpha \mathbf{V}_1(1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\lambda_3^* \alpha [\mathbf{V}_1(1) - \mathbf{V}_1(0)] \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^i}{\mu_1^* - \lambda_3^* \alpha} \\ \mathbf{V}_2(j) = \frac{\mu_2^* \mathbf{V}_2(0) - \lambda_3^* (1-\alpha) \mathbf{V}_2(1)}{\mu_2^* - \lambda_3^* (1-\alpha)} + \frac{\lambda_3^* (1-\alpha) [\mathbf{V}_2(1) - \mathbf{V}_2(0)] \left(\frac{\mu_2^*}{\lambda_3^* (1-\alpha)} \right)^j}{\mu_2^* - \lambda_3^* (1-\alpha)} \end{cases} \quad (27)$$

De waarden van $\mathbf{V}_1(0)$, $\mathbf{V}_1(1)$, $\mathbf{V}_2(0)$ en $\mathbf{V}_2(1)$ moeten hierbij nog onderzocht worden, om de oplossing kloppend te maken voor waarden aan de rand. Met dit doel wordt er nu eerst gekeken naar de grenswaarden van $i = 0$. In dat geval geldt voor $j > 0$:

$$\Delta_2 \mathbf{V}^\alpha(0, j) = \begin{cases} \gamma_2 + \lambda_3^* [\alpha \Delta_2 \mathbf{V}^\alpha(1, j) + (1-\alpha) \Delta_2 \mathbf{V}^\alpha(0, j+1)] \\ + \mu_2^* \Delta_2 \mathbf{V}^\alpha(0, j-1) \\ + [1 - (\lambda_3^* + \mu_2^*)] \Delta_2 \mathbf{V}^\alpha(0, j) \end{cases}$$

$$\Delta_2^2 \mathbf{V}^\alpha(0, j) = \begin{cases} \lambda_3^* [\alpha \Delta_2^2 \mathbf{V}^\alpha(1, j) + (1-\alpha) \Delta_2^2 \mathbf{V}^\alpha(0, j+1)] \\ + \mu_2^* \Delta_2^2 \mathbf{V}^\alpha(0, j-1) \\ + [1 - (\lambda_3^* + \mu_2^*)] \Delta_2^2 \mathbf{V}^\alpha(0, j) \end{cases}$$

Herschrijven levert:

$$(\lambda_3^* + \mu_2^*) \Delta_2^2 \mathbf{V}^\alpha(0, j) = \begin{cases} \lambda_3^* [\alpha \Delta_2^2 \mathbf{V}^\alpha(1, j) + (1 - \alpha) \Delta_2^2 \mathbf{V}^\alpha(0, j + 1)] \\ + \mu_2^* \Delta_2^2 \mathbf{V}^\alpha(0, j - 1) \end{cases}$$

Wat leidt tot het volgende systeem aan deelvergelijkingen.

$$\begin{cases} \lambda_3^* \alpha \mathbf{V}_1(0) & = \lambda_3^* \alpha \mathbf{V}_1(1) \\ [\lambda_3^* (1 - \alpha) + \mu_2^*] \mathbf{V}_2(j) & = \lambda_3^* (1 - \alpha) \mathbf{V}_2(j + 1) + \mu_2^* \mathbf{V}_2(j - 1) \end{cases}$$

In deze vergelijkingen is te zien dat, het Markov Beslissingsprobleem op de grens van $i = 0$, zich precies hetzelfde gedraagt als zijnde niet op de grens. Verder is te zien dat $\mathbf{V}_1(0) = \mathbf{V}_1(1)$, en dat dus de $\mathbf{V}_1(i)$ in Vergelijking (27) een constante ($\mathbf{V}_1(i) = d_1$) is. Zonder verlies van algemeenheid kan deze constante zo gekozen worden dat $\Delta_2^2 \mathbf{V}^\alpha(i, j)$ volledig verklaart wordt door alleen $\mathbf{V}_2(j)$, namelijk door te kiezen voor $d_1 = 0$. Hierdoor wordt $\Delta_2^2 \mathbf{V}^\alpha(i, j)$:

$$\Delta_2^2 \mathbf{V}^\alpha(i, j) = \frac{\mu_2^* \mathbf{V}_2(0) - \lambda_3^* (1 - \alpha) \mathbf{V}_2(1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} + \frac{\lambda_3^* (1 - \alpha) [\mathbf{V}_2(1) - \mathbf{V}_2(0)] \left(\frac{\mu_2^*}{\lambda_3^* (1 - \alpha)} \right)^j}{\mu_2^* - \lambda_3^* (1 - \alpha)}$$

D.2 Analyse van $\Delta_2 \mathbf{V}^\alpha(i + 1, j) - \Delta_2 \mathbf{V}^\alpha(i, j)$

In Vergelijking (26) was $\Delta_2 \mathbf{V}^\alpha(i, j)$, voor $i > 0$ en $j > 0$, als volgt.

$$\Delta_2 \mathbf{V}^\alpha(i, j) = \begin{cases} \gamma_2 + \lambda_3^* [\alpha \Delta_2 \mathbf{V}^\alpha(i + 1, j) + (1 - \alpha) \Delta_2 \mathbf{V}^\alpha(i, j + 1)] \\ + \mu_1^* \Delta_2 \mathbf{V}^\alpha(i - 1, j) \\ + \mu_2^* \Delta_2 \mathbf{V}^\alpha(i, j - 1) \\ + [1 - (\lambda_3^* + \mu_1^* + \mu_2^*)] \Delta_2 \mathbf{V}^\alpha(i, j) \end{cases}$$

Definieer nu: $\Delta_1 \Delta_2 \mathbf{V}^\alpha(i, j) := \Delta_2 \mathbf{V}^\alpha(i + 1, j) - \Delta_2 \mathbf{V}^\alpha(i, j)$. Hieruit volgt dat:

$$\Delta_1 \Delta_2 \mathbf{V}^\alpha(i, j) = \begin{cases} \lambda_3^* [\alpha \Delta_1 \Delta_2 \mathbf{V}^\alpha(i + 1, j) + (1 - \alpha) \Delta_1 \Delta_2 \mathbf{V}^\alpha(i, j + 1)] \\ + \mu_1^* \Delta_1 \Delta_2 \mathbf{V}^\alpha(i - 1, j) \\ + \mu_2^* \Delta_1 \Delta_2 \mathbf{V}^\alpha(i, j - 1) \\ + [1 - (\lambda_3^* + \mu_1^* + \mu_2^*)] \Delta_1 \Delta_2 \mathbf{V}^\alpha(i, j) \end{cases}$$

Wegens dezelfde reden en op dezelfde manier als eerst, wordt ook $\Delta_1 \Delta_2 \mathbf{V}^\alpha(i, j)$ opgesplitst in: $\Delta_1 \Delta_2 \mathbf{V}^\alpha(i, j) = \tilde{\mathbf{V}}_1(i) + \tilde{\mathbf{V}}_2(j)$, waarbij

$$\begin{cases} \tilde{\mathbf{V}}_1(i) = \frac{\mu_1^* \tilde{\mathbf{V}}_1(0) - \lambda_3^* \alpha \tilde{\mathbf{V}}_1(1)}{\mu_1^* - \lambda_3^* \alpha} + \frac{\lambda_3^* \alpha [\tilde{\mathbf{V}}_1(1) - \tilde{\mathbf{V}}_1(0)] \left(\frac{\mu_1^*}{\lambda_3^* \alpha} \right)^i}{\mu_1^* - \lambda_3^* \alpha} \\ \tilde{\mathbf{V}}_2(j) = \frac{\mu_2^* \tilde{\mathbf{V}}_2(0) - \lambda_3^* (1 - \alpha) \tilde{\mathbf{V}}_2(1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} + \frac{\lambda_3^* (1 - \alpha) [\tilde{\mathbf{V}}_2(1) - \tilde{\mathbf{V}}_2(0)] \left(\frac{\mu_2^*}{\lambda_3^* (1 - \alpha)} \right)^j}{\mu_2^* - \lambda_3^* (1 - \alpha)} \end{cases}$$

Ook hier moeten de waarden $\tilde{\mathbf{V}}_1(0)$, $\tilde{\mathbf{V}}_1(1)$, $\tilde{\mathbf{V}}_2(0)$ en $\tilde{\mathbf{V}}_2(1)$ nog onderzocht worden, om het aan de randen kloppend te maken. Er wordt hierbij eerst gekeken naar: $\Delta_1 \Delta_2 \mathbf{V}^\alpha(0, j)$, dan voor $j > 0$.

$$\begin{aligned}\Delta_2 \mathbf{V}^\alpha(1, j) &= \begin{cases} \gamma_2 + \lambda_3^* [\alpha \Delta_2 \mathbf{V}^\alpha(2, j) + (1 - \alpha) \Delta_2 \mathbf{V}^\alpha(1, j + 1)] \\ + \mu_1^* \Delta_2 \mathbf{V}^\alpha(0, j) \\ + \mu_2^* \Delta_2 \mathbf{V}^\alpha(1, j - 1) \\ + [1 - (\lambda_3^* + \mu_1^* + \mu_2^*)] \Delta_2 \mathbf{V}^\alpha(1, j) \end{cases} \\ \Delta_2 \mathbf{V}^\alpha(0, j) &= \begin{cases} \gamma_2 + \lambda_3^* [\alpha \Delta_2 \mathbf{V}^\alpha(1, j) + (1 - \alpha) \Delta_2 \mathbf{V}^\alpha(0, j + 1)] \\ + \mu_2^* \Delta_2 \mathbf{V}^\alpha(0, j - 1) \\ + [1 - (\lambda_3^* + \mu_2^*)] \Delta_2 \mathbf{V}^\alpha(0, j) \end{cases}\end{aligned}$$

Daaruit volgt dat:

$$\Delta_1 \Delta_2 \mathbf{V}^\alpha(0, j) = \begin{cases} \lambda_3^* [\alpha \Delta_1 \Delta_2 \mathbf{V}^\alpha(1, j) + (1 - \alpha) \Delta_1 \Delta_2 \mathbf{V}^\alpha(0, j + 1)] \\ + \mu_2^* \Delta_1 \Delta_2 \mathbf{V}^\alpha(0, j - 1) - \mu_1^* \Delta_1 \Delta_2 \mathbf{V}^\alpha(0, j) \\ + [1 - (\lambda_3^* + \mu_2^*)] \Delta_1 \Delta_2 \mathbf{V}^\alpha(0, j) \end{cases}$$

Ofwel:

$$[\lambda_3^* \alpha + \mu_1^*] \Delta_1 \Delta_2 \mathbf{V}^\alpha(0, j) + [\lambda_3^* (1 - \alpha) + \mu_2^*] \Delta_1 \Delta_2 \mathbf{V}^\alpha(0, j) = \begin{cases} \lambda_3^* \alpha \Delta_1 \Delta_2 \mathbf{V}^\alpha(1, j) \\ + \lambda_3^* (1 - \alpha) \Delta_1 \Delta_2 \mathbf{V}^\alpha(0, j + 1) \\ + \mu_2^* \Delta_1 \Delta_2 \mathbf{V}^\alpha(0, j - 1) \end{cases}$$

Door opsplitsen ontstaan de volgende vergelijkingen:

$$\begin{cases} [\lambda_3^* \alpha + \mu_1^*] \Delta_1 \Delta_2 \mathbf{V}^\alpha(0, j) &= \lambda_3^* \alpha \Delta_1 \Delta_2 \mathbf{V}^\alpha(1, j) \\ [\lambda_3^* (1 - \alpha) + \mu_2^*] \Delta_1 \Delta_2 \mathbf{V}^\alpha(0, j) &= \lambda_3^* (1 - \alpha) \Delta_1 \Delta_2 \mathbf{V}^\alpha(0, j + 1) + \mu_2^* \Delta_1 \Delta_2 \mathbf{V}^\alpha(0, j - 1) \end{cases}$$

Door substitutie van $\Delta_1 \Delta_2 \mathbf{V}^\alpha(i, j) = \tilde{\mathbf{V}}_1(i) + \tilde{\mathbf{V}}_2(j)$ in de eerste vergelijking ontstaat:

$$[\lambda_3^* \alpha + \mu_1^*] [\tilde{\mathbf{V}}_1(0) + \tilde{\mathbf{V}}_2(j)] = \lambda_3^* \alpha [\tilde{\mathbf{V}}_1(1) + \tilde{\mathbf{V}}_2(j)]$$

Herschrijven levert:

$$\mu_1^* \tilde{\mathbf{V}}_2(j) = \lambda_3^* \alpha \tilde{\mathbf{V}}_1(1) - [\lambda_3^* \alpha + \mu_1^*] \tilde{\mathbf{V}}_1(0) \quad (28)$$

Hieruit volgt dat $\tilde{\mathbf{V}}_2(j)$ een constante ($\tilde{d}_2$) is voor $j > 0$. Vervolgens wordt op dezelfde manier $\Delta_1 \Delta_2 \mathbf{V}^\alpha(i, 0)$ onderzocht voor $i > 0$.

$$\begin{aligned}\Delta_2 \mathbf{V}^\alpha(i + 1, 0) &= \begin{cases} \gamma_2 + \lambda_3^* [\alpha \Delta_2 \mathbf{V}^\alpha(i + 2, 0) + (1 - \alpha) \Delta_2 \mathbf{V}^\alpha(i + 1, 1)] \\ + \mu_1^* \Delta_2 \mathbf{V}^\alpha(i, 0) \\ + [1 - (\lambda_3^* + \mu_1^* + \mu_2^*)] \Delta_2 \mathbf{V}^\alpha(i + 1, 0) \end{cases} \\ \Delta_2 \mathbf{V}^\alpha(i, 0) &= \begin{cases} \gamma_2 + \lambda_3^* [\alpha \Delta_2 \mathbf{V}^\alpha(i + 1, 0) + (1 - \alpha) \Delta_2 \mathbf{V}^\alpha(i, 1)] \\ + \mu_1^* \Delta_2 \mathbf{V}^\alpha(i - 1, 0) \\ + [1 - (\lambda_3^* + \mu_1^* + \mu_2^*)] \Delta_2 \mathbf{V}^\alpha(i, 0) \end{cases}\end{aligned}$$

Daaruit volgt dat:

$$\Delta_1 \Delta_2 \mathbf{V}^\alpha(i, 0) = \begin{cases} \lambda_3^* [\alpha \Delta_1 \Delta_2 \mathbf{V}^\alpha(i+1, 0) + (1-\alpha) \Delta_1 \Delta_2 \mathbf{V}^\alpha(i, 1)] \\ + \mu_1^* \Delta_1 \Delta_2 \mathbf{V}^\alpha(i-1, 0) \\ + [1 - (\lambda_3^* + \mu_1^* + \mu_2^*)] \Delta_1 \Delta_2 \mathbf{V}^\alpha(i, 0) \end{cases}$$

Ofwel:

$$[\lambda_3^* \alpha + \mu_1^*] \Delta_1 \Delta_2 \mathbf{V}^\alpha(i, 0) + [\lambda_3^* (1-\alpha) + \mu_2^*] \Delta_1 \Delta_2 \mathbf{V}^\alpha(i, 0) = \begin{cases} \lambda_3^* \alpha \Delta_1 \Delta_2 \mathbf{V}^\alpha(i+1, 0) \\ + \mu_1^* \Delta_1 \Delta_2 \mathbf{V}^\alpha(i-1, 0) \\ + \lambda_3^* (1-\alpha) \Delta_1 \Delta_2 \mathbf{V}^\alpha(i, 1) \end{cases}$$

Door opsplitsen ontstaan de volgende vergelijkingen.

$$\begin{cases} [\lambda_3^* \alpha + \mu_1^*] \Delta_1 \Delta_2 \mathbf{V}^\alpha(i, 0) & = \lambda_3^* \alpha \Delta_1 \Delta_2 \mathbf{V}^\alpha(i+1, 0) + \mu_1^* \Delta_1 \Delta_2 \mathbf{V}^\alpha(i-1, 0) \\ [\lambda_3^* (1-\alpha) + \mu_2^*] \Delta_1 \Delta_2 \mathbf{V}^\alpha(i, 0) & = \lambda_3^* (1-\alpha) \Delta_1 \Delta_2 \mathbf{V}^\alpha(i, 1) \end{cases}$$

Door substitutie van $\Delta_1 \Delta_2 \mathbf{V}^\alpha(i, j) = \tilde{\mathbf{V}}_1(i) + \tilde{\mathbf{V}}_2(j)$ in de laatste vergelijking ontstaat:

$$[\lambda_3^* (1-\alpha) + \mu_2^*] [\tilde{\mathbf{V}}_1(i) + \tilde{\mathbf{V}}_2(0)] = \lambda_3^* (1-\alpha) [\tilde{\mathbf{V}}_1(i) + \tilde{\mathbf{V}}_2(1)]$$

Herschrijven levert:

$$\mu_2^* \tilde{\mathbf{V}}_1(i) = \lambda_3^* (1-\alpha) \tilde{\mathbf{V}}_2(1) - [\lambda_3^* (1-\alpha) + \mu_2^*] \tilde{\mathbf{V}}_2(0)$$

Hieruit volgt dat $\tilde{\mathbf{V}}_1(i)$ een constante ($\tilde{d}_1$) is voor $i > 0$. Dit samen reduceert Vergelijking (28) tot:

$$\mu_1^* \tilde{d}_2 = \lambda_3^* \alpha \tilde{d}_1 - [\lambda_3^* \alpha + \mu_1^*] \tilde{d}_1 = -\mu_1^* \tilde{d}_1$$

Hieruit volgt dat $\tilde{d}_1 = -\tilde{d}_2$, en dit maakt dat: $\Delta_1 \Delta_2 \mathbf{V}^\alpha(i, j) = 0$.

D.3 Verschilvergelijking $\Delta_2 \mathbf{V}^\alpha(i, j)$ oplossen

Tot dusver is de volgende vergelijking gevonden voor $\Delta_2^2 \mathbf{V}^\alpha(i, j)$

$$\Delta_2^2 \mathbf{V}^\alpha(i, j) = \frac{\mu_2^* \mathbf{V}_2(0) - \lambda_3^* (1-\alpha) \mathbf{V}_2(1)}{\mu_2^* - \lambda_3^* (1-\alpha)} + \frac{\lambda_3^* (1-\alpha) [\mathbf{V}_2(1) - \mathbf{V}_2(0)] \left(\frac{\mu_2^*}{\lambda_3^* (1-\alpha)} \right)^j}{\mu_2^* - \lambda_3^* (1-\alpha)} \quad (29)$$

Verder is vastgesteld dat $\Delta_1 \Delta_2 \mathbf{V}^\alpha(i, j) = 0$, zodat $\Delta_2 \mathbf{V}^\alpha(i, j)$ onafhankelijk is van i voor alle mogelijke waarden van j . Een andere expressie voor $\Delta_2 \mathbf{V}^\alpha(i, j)$ is:

$$\Delta_2 \mathbf{V}^\alpha(i, j) = \Delta_2 \mathbf{V}^\alpha(i, 0) + \sum_{k=0}^{j-1} \Delta_2^2 \mathbf{V}^\alpha(i, k) \quad (30)$$

Hierbij is gebruik gemaakt van het feit dat $\Delta_2^2 \mathbf{V}^\alpha(i, j) = \Delta_2 \mathbf{V}^\alpha(i, j+1) - \Delta_2 \mathbf{V}^\alpha(i, j)$. Door het invullen van $\Delta_2^2 \mathbf{V}^\alpha(i, j)$ en het herschrijven van de somformule, ontstaat de volgende vergelijking.

$$\Delta_2 \mathbf{V}^\alpha(i, j) = \begin{cases} \Delta_2 \mathbf{V}^\alpha(i, 0) + \left(\frac{\mu_2^* \mathbf{V}_2(0) - \lambda_3^* (1-\alpha) \mathbf{V}_2(1)}{\mu_2^* - \lambda_3^* (1-\alpha)} \right) j \\ + \frac{\lambda_3^* (1-\alpha) [\mathbf{V}_2(1) - \mathbf{V}_2(0)]}{\mu_2^* - \lambda_3^* (1-\alpha)} \frac{1 - \left[\frac{\mu_2^*}{\lambda_3^* (1-\alpha)} \right]^j}{1 - \frac{\mu_2^*}{\lambda_3^* (1-\alpha)}} \end{cases}$$

Aangezien $\Delta_2 \mathbf{V}^\alpha(i, j)$ onafhankelijk is van i voor alle mogelijke waarden van j , geldt dat $\Delta_2 \mathbf{V}^\alpha(i, 0)$ een constante is. De waarde van deze constante wordt even verderop bepaald. Nu wordt eerst de zojuist gevonden vergelijking voor $\Delta_2 \mathbf{V}^\alpha(i, j)$ gesubstitueerd in Vergelijking (26).

$$\begin{aligned} & \Delta_2 \mathbf{V}^\alpha(i, 0) + \left(\frac{\mu_2^* \mathbf{V}_2(0) - \lambda_3^* (1-\alpha) \mathbf{V}_2(1)}{\mu_2^* - \lambda_3^* (1-\alpha)} \right) j + \frac{\lambda_3^* (1-\alpha) [\mathbf{V}_2(1) - \mathbf{V}_2(0)]}{\mu_2^* - \lambda_3^* (1-\alpha)} \frac{1 - \left(\frac{\mu_2^*}{\lambda_3^* (1-\alpha)} \right)^j}{1 - \frac{\mu_2^*}{\lambda_3^* (1-\alpha)}} = \\ & \left\{ \begin{aligned} & \gamma_2 + \lambda_3^* \alpha \left[\Delta_2 \mathbf{V}^\alpha(i+1, 0) + \left(\frac{\mu_2^* \mathbf{V}_2(0) - \lambda_3^* (1-\alpha) \mathbf{V}_2(1)}{\mu_2^* - \lambda_3^* (1-\alpha)} \right) j + \frac{\lambda_3^* (1-\alpha) [\mathbf{V}_2(1) - \mathbf{V}_2(0)]}{\mu_2^* - \lambda_3^* (1-\alpha)} \frac{1 - \left(\frac{\mu_2^*}{\lambda_3^* (1-\alpha)} \right)^j}{1 - \frac{\mu_2^*}{\lambda_3^* (1-\alpha)}} \right] \\ & + \mu_1^* \left[\Delta_2 \mathbf{V}^\alpha(i-1, 0) + \left(\frac{\mu_2^* \mathbf{V}_2(0) - \lambda_3^* (1-\alpha) \mathbf{V}_2(1)}{\mu_2^* - \lambda_3^* (1-\alpha)} \right) j + \frac{\lambda_3^* (1-\alpha) [\mathbf{V}_2(1) - \mathbf{V}_2(0)]}{\mu_2^* - \lambda_3^* (1-\alpha)} \frac{1 - \left(\frac{\mu_2^*}{\lambda_3^* (1-\alpha)} \right)^j}{1 - \frac{\mu_2^*}{\lambda_3^* (1-\alpha)}} \right] \\ & + \lambda_3^* \alpha (1-\alpha) \left[\Delta_2 \mathbf{V}^\alpha(i, 0) + \left(\frac{\mu_2^* \mathbf{V}_2(0) - \lambda_3^* (1-\alpha) \mathbf{V}_2(1)}{\mu_2^* - \lambda_3^* (1-\alpha)} \right) (j+1) + \frac{\lambda_3^* (1-\alpha) [\mathbf{V}_2(1) - \mathbf{V}_2(0)]}{\mu_2^* - \lambda_3^* (1-\alpha)} \frac{1 - \left(\frac{\mu_2^*}{\lambda_3^* (1-\alpha)} \right)^{j+1}}{1 - \frac{\mu_2^*}{\lambda_3^* (1-\alpha)}} \right] \\ & + \mu_2^* \left[\Delta_2 \mathbf{V}^\alpha(i, 0) + \left(\frac{\mu_2^* \mathbf{V}_2(0) - \lambda_3^* (1-\alpha) \mathbf{V}_2(1)}{\mu_2^* - \lambda_3^* (1-\alpha)} \right) (j-1) + \frac{\lambda_3^* (1-\alpha) [\mathbf{V}_2(1) - \mathbf{V}_2(0)]}{\mu_2^* - \lambda_3^* (1-\alpha)} \frac{1 - \left(\frac{\mu_2^*}{\lambda_3^* (1-\alpha)} \right)^{j-1}}{1 - \frac{\mu_2^*}{\lambda_3^* (1-\alpha)}} \right] \\ & + [1 - (\lambda_3^* + \mu_1^* + \mu_2^*)] \left[\Delta_2 \mathbf{V}^\alpha(i, 0) + \left(\frac{\mu_2^* \mathbf{V}_2(0) - \lambda_3^* (1-\alpha) \mathbf{V}_2(1)}{\mu_2^* - \lambda_3^* (1-\alpha)} \right) j + \frac{\lambda_3^* (1-\alpha) [\mathbf{V}_2(1) - \mathbf{V}_2(0)]}{\mu_2^* - \lambda_3^* (1-\alpha)} \frac{1 - \left(\frac{\mu_2^*}{\lambda_3^* (1-\alpha)} \right)^j}{1 - \frac{\mu_2^*}{\lambda_3^* (1-\alpha)}} \right] \end{aligned} \right\} \end{aligned}$$

Herschrijven levert:

$$0 = \begin{cases} \gamma_2 + \lambda_3^* \alpha [\Delta_2 \mathbf{V}^\alpha(i+1, 0) - \Delta_2 \mathbf{V}^\alpha(i, 0)] \\ + \mu_1^* [\Delta_2 \mathbf{V}^\alpha(i-1, 0) - \Delta_2 \mathbf{V}^\alpha(i, 0)] \\ + \lambda_3^* \alpha (1-\alpha) \left[\left(\frac{\mu_2^* \mathbf{V}_2(0) - \lambda_3^* (1-\alpha) \mathbf{V}_2(1)}{\mu_2^* - \lambda_3^* (1-\alpha)} \right) + \frac{\lambda_3^* (1-\alpha) [\mathbf{V}_2(1) - \mathbf{V}_2(0)]}{\mu_2^* - \lambda_3^* (1-\alpha)} \left(\frac{\mu_2^*}{\lambda_3^* (1-\alpha)} \right)^j \right] \\ - \mu_2^* \left[\left(\frac{\mu_2^* \mathbf{V}_2(0) - \lambda_3^* (1-\alpha) \mathbf{V}_2(1)}{\mu_2^* - \lambda_3^* (1-\alpha)} \right) + \frac{\lambda_3^* (1-\alpha) [\mathbf{V}_2(1) - \mathbf{V}_2(0)]}{\mu_2^* - \lambda_3^* (1-\alpha)} \left(\frac{\mu_2^*}{\lambda_3^* (1-\alpha)} \right)^{j-1} \right] \end{cases}$$

Aangezien $\Delta_1 \Delta_2 \mathbf{V}^\alpha(i, j) = 0$, kan de voorgaande vergelijking verder vereenvoudigd worden tot:

$$\gamma_2 = \begin{cases} \mu_2^* \left[\left(\frac{\mu_2^* \mathbf{V}_2(0) - \lambda_3^* (1-\alpha) \mathbf{V}_2(1)}{\mu_2^* - \lambda_3^* (1-\alpha)} \right) + \frac{\lambda_3^* (1-\alpha) [\mathbf{V}_2(1) - \mathbf{V}_2(0)]}{\mu_2^* - \lambda_3^* (1-\alpha)} \left(\frac{\mu_2^*}{\lambda_3^* (1-\alpha)} \right)^{j-1} \right] \\ - \lambda_3^* (1-\alpha) \left[\left(\frac{\mu_2^* \mathbf{V}_2(0) - \lambda_3^* (1-\alpha) \mathbf{V}_2(1)}{\mu_2^* - \lambda_3^* (1-\alpha)} \right) + \frac{\lambda_3^* (1-\alpha) [\mathbf{V}_2(1) - \mathbf{V}_2(0)]}{\mu_2^* - \lambda_3^* (1-\alpha)} \left(\frac{\mu_2^*}{\lambda_3^* (1-\alpha)} \right)^j \right] \end{cases}$$

$$\gamma_2 = \begin{cases} \frac{\lambda_3^* (1-\alpha) [\mathbf{V}_2(1) - \mathbf{V}_2(0)]}{\mu_2^* - \lambda_3^* (1-\alpha)} \left[\mu_2^* \left(\frac{\mu_2^*}{\lambda_3^* (1-\alpha)} \right)^{j-1} - \lambda_3^* (1-\alpha) \left(\frac{\mu_2^*}{\lambda_3^* (1-\alpha)} \right)^j \right] \\ + (\mu_2^* - \lambda_3^* (1-\alpha)) \left[\frac{\mu_2^* \mathbf{V}_2(0) - \lambda_3^* (1-\alpha) \mathbf{V}_2(1)}{\mu_2^* - \lambda_3^* (1-\alpha)} \right] \end{cases}$$

$$\gamma_2 = \begin{cases} \frac{\lambda_3^* (1-\alpha) [\mathbf{V}_2(1) - \mathbf{V}_2(0)]}{\mu_2^* - \lambda_3^* (1-\alpha)} \left[\mu_2^* \left(\frac{\mu_2^*}{\lambda_3^* (1-\alpha)} \right)^{j-1} - \lambda_3^* (1-\alpha) \left(\frac{\mu_2^*}{\lambda_3^* (1-\alpha)} \right)^j \right] \\ + \mu_2^* \mathbf{V}_2(0) - \lambda_3^* (1-\alpha) \mathbf{V}_2(1) \end{cases}$$

Hieruit volgt dat:

$$\mu_2^* \mathbf{V}_2(0) - \lambda_3^* (1-\alpha) \mathbf{V}_2(1) = \gamma_2$$

Deze vergelijking wordt gebruikt om $\mathbf{V}_2(1)$ te substitueren in Vergelijking (29).

$$\Delta_2^2 \mathbf{V}^\alpha(i, j) = \frac{\gamma_2}{\mu_2^* - \lambda_3^* (1-\alpha)} + \left[\mathbf{V}_2(0) - \frac{\gamma_2}{\mu_2^* - \lambda_3^* (1-\alpha)} \right] \left(\frac{\mu_2^*}{\lambda_3^* (1-\alpha)} \right)^j$$

Hierdoor wordt $\Delta_2 \mathbf{V}^\alpha(i, j)$ met behulp van vergelijking (30):

$$\Delta_2 \mathbf{V}^\alpha(i, j) = \begin{cases} \Delta_2 \mathbf{V}^\alpha(i, 0) + \left(\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1-\alpha)} \right) j \\ + \left[\mathbf{V}_2(0) - \frac{\gamma_2}{\mu_2^* - \lambda_3^* (1-\alpha)} \right] \frac{1 - \left[\frac{\mu_2^*}{\lambda_3^* (1-\alpha)} \right]^j}{1 - \frac{\mu_2^*}{\lambda_3^* (1-\alpha)}} \end{cases} \quad (31)$$

D.4 Waarde bepalen van constante $\Delta_2 \mathbf{V}^\alpha(i, 0)$

Nu wordt de constante $\Delta_2 \mathbf{V}^\alpha(i, 0)$ onderzocht, door naar de bijbehorende verschilvergelijking te kijken.

$$\Delta_2 \mathbf{V}^\alpha(i, 0) = \begin{cases} \gamma_2 + \lambda_3^* [\alpha \Delta_2 \mathbf{V}^\alpha(i+1, 0) + (1-\alpha) \Delta_2 \mathbf{V}^\alpha(i, 1)] \\ + \mu_1^* \Delta_2 \mathbf{V}^\alpha(i-1, 0) \\ + [1 - (\lambda_3^* + \mu_1^* + \mu_2^*)] \Delta_2 \mathbf{V}^\alpha(i, 0) \end{cases}$$

Dit omschrijven levert de volgende vergelijking op.

$$0 = \begin{cases} \gamma_2 + \lambda_3^* \alpha [\Delta_2 \mathbf{V}^\alpha(i+1, 0) - \Delta_2 \mathbf{V}^\alpha(i, 0)] \\ + \lambda_3^* (1-\alpha) [\Delta_2 \mathbf{V}^\alpha(i, 1) - \Delta_2 \mathbf{V}^\alpha(i, 0)] \\ + \mu_1^* [\Delta_2 \mathbf{V}^\alpha(i-1, 0) - \Delta_2 \mathbf{V}^\alpha(i, 0)] \\ - \mu_2^* \Delta_2 \mathbf{V}^\alpha(i, 0) \end{cases}$$

Aangezien $\Delta_1 \Delta_2 \mathbf{V}^\alpha(i, j) = 0$, kun je dit herschrijven tot:

$$0 = \gamma_2 + \lambda_3^* (1-\alpha) [\Delta_2 \mathbf{V}^\alpha(i, 1) - \Delta_2 \mathbf{V}^\alpha(i, 0)] - \mu_2^* \Delta_2 \mathbf{V}^\alpha(i, 0)$$

Met Vergelijking (31) wordt verkregen dat: $\Delta_2 \mathbf{V}^\alpha(i, 1) = \Delta_2 \mathbf{V}^\alpha(i, 0) + \mathbf{V}_2(0)$. Invullen geeft:

$$0 = \gamma_2 + \lambda_3^* (1 - \alpha) \mathbf{V}_2(0) - \mu_2^* \Delta_2 \mathbf{V}^\alpha(i, 0)$$

Ofwel:

$$\Delta_2 \mathbf{V}^\alpha(i, 0) = \frac{\lambda_3^* (1 - \alpha)}{\mu_2^*} \mathbf{V}_2(0) + \frac{\gamma_2}{\mu_2^*}$$

Deze uitkomst voor $\Delta_2 \mathbf{V}^\alpha(i, 0)$ gesubstitueerd in Vergelijking (31) levert:

$$\Delta_2 \mathbf{V}^\alpha(i, j) = \begin{cases} \frac{\lambda_3^* (1 - \alpha)}{\mu_2^*} \mathbf{V}_2(0) + \frac{\gamma_2}{\mu_2^*} + \left(\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right) j \\ + \left[\mathbf{V}_2(0) - \frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \frac{1 - \left[\frac{\mu_2^*}{\lambda_3^* (1 - \alpha)} \right]^j}{1 - \frac{\mu_2^*}{\lambda_3^* (1 - \alpha)}} \end{cases} \quad (32)$$

D.5 Het afleiden van $\mathbf{V}^\alpha(i, j)$

Aangezien $\Delta_2 \mathbf{V}^\alpha(i, j) = \mathbf{V}^\alpha(i, j + 1) - \mathbf{V}^\alpha(i, j)$ kan de volgende vergelijking opgesteld worden.

$$\mathbf{V}^\alpha(i, j) = \mathbf{V}^\alpha(i, 0) + \sum_{k=0}^{j-1} \Delta_2 \mathbf{V}^\alpha(i, k)$$

Door het herschrijven van de somformule en gebruik te maken van vergelijking (32) ontstaat:

$$\mathbf{V}^\alpha(i, j) = \begin{cases} \mathbf{V}^\alpha(i, 0) + \left(\frac{\lambda_3^* (1 - \alpha)}{\mu_2^*} \mathbf{V}_2(0) + \frac{\gamma_2}{\mu_2^*} \right) j + \frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \frac{j(j-1)}{2} \\ + \frac{1}{1 - \frac{\mu_2^*}{\lambda_3^* (1 - \alpha)}} \left[\mathbf{V}_2(0) - \frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \left[j - \frac{1 - \left(\frac{\mu_2^*}{\lambda_3^* (1 - \alpha)} \right)^j}{1 - \left(\frac{\mu_2^*}{\lambda_3^* (1 - \alpha)} \right)} \right] \end{cases} \quad (33)$$

Voor een uniek antwoord is het belangrijk dat $\mathbf{V}^\alpha(i, j)$ niet exponentieel groeit. Daarom wordt de laatste constante $\mathbf{V}_2(0)$ in Vergelijking (33) zo gekozen, dat de term $\left(\frac{\mu_2^*}{\lambda_3^* (1 - \alpha)} \right)^j$ verdwijnt.

$$\mathbf{V}_2(0) = \frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)}$$

Dit gesubstitueerd in Vergelijking (33), geeft:

$$\mathbf{V}^\alpha(i, j) = \mathbf{V}^\alpha(i, 0) + \left(\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right) j + \frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \frac{j(j-1)}{2}$$

Ofwel:

$$\mathbf{V}^\alpha(i, j) = \mathbf{V}^\alpha(i, 0) + \frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \frac{j(j+1)}{2}$$

E Uitwerking van $\mathbf{g}^\alpha$ voor alle situaties

Lemma E.1 (Lemma A.2). Voor $i \geq 0$ en $j \geq 0$ geldt dat:

$$\mathbf{g}^\alpha = \lambda_3^* \alpha \left[\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] + \lambda_3^* (1 - \alpha) \left[\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right]$$

Bewijs : Tot dusver is de volgende vergelijking voor $\mathbf{g}^\alpha$ gevonden.

$$\mathbf{g}^\alpha = \begin{cases} \gamma_1 i + \gamma_2 j + \lambda_3^* \alpha \left(\frac{\gamma_1 (i+1)}{\mu_1^* - \lambda_3^* \alpha} \right) + \lambda_3^* (1 - \alpha) \left(\frac{\gamma_2 (j+1)}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right) \\ - \mu_1^* \left(\frac{\gamma_1 i}{\mu_1^* - \lambda_3^* \alpha} \right) \mathbb{1}_{i>0} - \mu_2^* \left(\frac{\gamma_2 j}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right) \mathbb{1}_{j>0} \end{cases}$$

Vanwege de stapfunctie's in deze vergelijking wordt elke situatie afzonderlijk bekeken, door te beginnen met $(i, j) = (0, 0)$, dan krijg je:

$$\mathbf{g}^\alpha = \lambda_3^* \alpha \left[\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] + \lambda_3^* (1 - \alpha) \left[\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right]$$

Als $(i, j) = (i, 0)$ met $i > 0$, dan wordt dit:

$$\begin{aligned} \mathbf{g}^\alpha &= \begin{cases} \gamma_1 i + \lambda_3^* \alpha \left[\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] + \lambda_3^* (1 - \alpha) \left[\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \\ + [\lambda_3^* \alpha - \mu_1^*] \left(\frac{\gamma_1 i}{\mu_1^* - \lambda_3^* \alpha} \right) \end{cases} \\ &= \lambda_3^* \alpha \left[\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] + \lambda_3^* (1 - \alpha) \left[\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \end{aligned}$$

Bij $(i, j) = (0, j)$ met $j > 0$, dan is dit:

$$\begin{aligned} \mathbf{g}^\alpha &= \begin{cases} \gamma_2 j + \lambda_3^* \alpha \left[\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] + \lambda_3^* (1 - \alpha) \left[\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \\ + [\lambda_3^* (1 - \alpha) - \mu_2^*] \left(\frac{\gamma_2 j}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right) \end{cases} \\ &= \lambda_3^* \alpha \left[\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] + \lambda_3^* (1 - \alpha) \left[\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \end{aligned}$$

En in de situatie dat $i, j > 0$, dan krijg je:

$$\begin{aligned} \mathbf{g}^\alpha &= \begin{cases} \gamma_1 i + \gamma_2 j + \lambda_3^* \alpha \left[\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] + \lambda_3^* (1 - \alpha) \left[\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \\ + [\lambda_3^* \alpha - \mu_1^*] \left(\frac{\gamma_1 i}{\mu_1^* - \lambda_3^* \alpha} \right) + [\lambda_3^* (1 - \alpha) - \mu_2^*] \left(\frac{\gamma_2 j}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right) \end{cases} \\ &= \lambda_3^* \alpha \left[\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] + \lambda_3^* (1 - \alpha) \left[\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right] \end{aligned}$$

Voor alle vier de situaties is de constante $\mathbf{g}^\alpha$ hetzelfde, en is dus $\mathbf{g}^\alpha$ voor elke situatie:

$$\mathbf{g}^\alpha = \lambda_3^* \alpha \left[\frac{\gamma_1}{\mu_1^* - \lambda_3^* \alpha} \right] + \lambda_3^* (1 - \alpha) \left[\frac{\gamma_2}{\mu_2^* - \lambda_3^* (1 - \alpha)} \right]$$

Q.E.D.

F De gevonden $V^\alpha(i, j)$ en g^α zijn een oplossing

Stelling F.1 (Stelling 3.2). De gevonden $V^\alpha(i, j)$ en g^α in Paragraaf 3.2 zouden een oplossing zijn voor de Vergelijking (11), welke eveneens in Paragraaf 3.2 staat.

Bewijs : Door deze mogelijke oplossing in te vullen in Vergelijking (11), krijg je:

$$2 \left[\frac{\gamma_1 [\lambda_1^* + \lambda_3^* \alpha]}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{\gamma_2 [\lambda_2^* + \lambda_3^* (1-\alpha)]}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} \right] + \frac{\gamma_1 i (i+1)}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{\gamma_2 j (j+1)}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} =$$

$$\left\{ \begin{aligned} & 2 (\gamma_1 i + \gamma_2 j) + [\lambda_1^* + \lambda_3^* \alpha] \left[\frac{\gamma_1 (i+2) (i+1)}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{\gamma_2 j (j+1)}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} \right] \\ & + [\lambda_2^* + \lambda_3^* (1-\alpha)] \left[\frac{\gamma_1 i (i+1)}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{\gamma_2 (j+2) (j+1)}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} \right] \\ & + \mu_1^* \left[\frac{\gamma_1 i (i-1)}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{\gamma_2 j (j+1)}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} \right] \mathbb{1}_{i>0} \\ & + \mu_2^* \left[\frac{\gamma_1 i (i+1)}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{\gamma_2 j (j-1)}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} \right] \mathbb{1}_{j>0} \\ & + [1 - (\lambda_1^* + \lambda_2^* + \lambda_3^* + \mu_1^* \mathbb{1}_{i>0} + \mu_2^* \mathbb{1}_{j>0})] \left[\frac{\gamma_1 i (i+1)}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{\gamma_2 j (j+1)}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} \right] \end{aligned} \right.$$

Dit kun je herschrijven tot:

$$2 \left[\frac{\gamma_1 [\lambda_1^* + \lambda_3^* \alpha]}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{\gamma_2 [\lambda_2^* + \lambda_3^* (1-\alpha)]}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} \right] =$$

$$\left\{ \begin{aligned} & 2 (\gamma_1 i + \gamma_2 j) + [\lambda_1^* + \lambda_3^* \alpha] \left[\frac{\gamma_1 (i+2) (i+1)}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} - \frac{\gamma_1 i (i+1)}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} \right] \\ & + [\lambda_2^* + \lambda_3^* (1-\alpha)] \left[\frac{\gamma_2 (j+2) (j+1)}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} - \frac{\gamma_2 j (j+1)}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} \right] \\ & + \mu_1^* \left[\frac{\gamma_1 i (i-1)}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} - \frac{\gamma_1 i (i+1)}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} \right] \mathbb{1}_{i>0} \\ & + \mu_2^* \left[\frac{\gamma_2 j (j-1)}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} - \frac{\gamma_2 j (j+1)}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} \right] \mathbb{1}_{j>0} \end{aligned} \right.$$

Ofwel:

$$2 \left[\frac{\gamma_1 [\lambda_1^* + \lambda_3^* \alpha]}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{\gamma_2 [\lambda_2^* + \lambda_3^* (1-\alpha)]}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} \right] =$$

$$\left\{ \begin{aligned} & 2 (\gamma_1 i + \gamma_2 j) + [\lambda_1^* + \lambda_3^* \alpha] \left[\frac{2 \gamma_1 (i+1)}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} \right] + \mu_1^* \left[\frac{-2 \gamma_1 i}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} \right] \mathbb{1}_{i>0} \\ & + [\lambda_2^* + \lambda_3^* (1-\alpha)] \left[\frac{2 \gamma_2 (j+1)}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} \right] + \mu_2^* \left[\frac{-2 \gamma_2 j}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} \right] \mathbb{1}_{j>0} \end{aligned} \right.$$

Als $(i, j) = (0, 0)$, dan krijg je:

$$2 \left[\frac{\gamma_1 [\lambda_1^* + \lambda_3^* \alpha]}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{\gamma_2 [\lambda_2^* + \lambda_3^* (1-\alpha)]}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} \right] = \left\{ \begin{aligned} & [\lambda_1^* + \lambda_3^* \alpha] \left[\frac{2 \gamma_1}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} \right] \\ & + [\lambda_2^* + \lambda_3^* (1-\alpha)] \left[\frac{2 \gamma_2}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1-\alpha)]} \right] \end{aligned} \right.$$

Als $(i, j) = (i, 0)$ met $i > 0$, dan is dit:

$$2 \left[\frac{\gamma_1 [\lambda_1^* + \lambda_3^* \alpha]}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{\gamma_2 [\lambda_2^* + \lambda_3^* (1 - \alpha)]}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1 - \alpha)]} \right] = \begin{cases} 2 \gamma_1 i + [\lambda_1^* + \lambda_3^* \alpha] \left[\frac{2 \gamma_1}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} \right] \\ - (\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]) \left[\frac{2 \gamma_1 i}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} \right] \\ + [\lambda_2^* + \lambda_3^* (1 - \alpha)] \left[\frac{2 \gamma_2}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1 - \alpha)]} \right] \end{cases}$$

Als $(i, j) = (0, j)$ met $j > 0$, dan wordt dit:

$$2 \left[\frac{\gamma_1 [\lambda_1^* + \lambda_3^* \alpha]}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{\gamma_2 [\lambda_2^* + \lambda_3^* (1 - \alpha)]}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1 - \alpha)]} \right] = \begin{cases} 2 \gamma_2 j + [\lambda_1^* + \lambda_3^* \alpha] \left[\frac{2 \gamma_1}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} \right] \\ + [\lambda_2^* + \lambda_3^* (1 - \alpha)] \left[\frac{2 \gamma_2}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1 - \alpha)]} \right] \\ - (\mu_2^* - [\lambda_2^* + \lambda_3^* (1 - \alpha)]) \left[\frac{2 \gamma_2 j}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1 - \alpha)]} \right] \end{cases}$$

En als $(i, j) = (i, j)$ met $i > 0$ en $j > 0$, dan krijg je:

$$2 \left[\frac{\gamma_1 [\lambda_1^* + \lambda_3^* \alpha]}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} + \frac{\gamma_2 [\lambda_2^* + \lambda_3^* (1 - \alpha)]}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1 - \alpha)]} \right] = \begin{cases} 2 (\gamma_1 i + \gamma_2 j) + [\lambda_1^* + \lambda_3^* \alpha] \left[\frac{2 \gamma_1}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} \right] \\ - (\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]) \left[\frac{2 \gamma_1 i}{\mu_1^* - [\lambda_1^* + \lambda_3^* \alpha]} \right] \\ + [\lambda_2^* + \lambda_3^* (1 - \alpha)] \left[\frac{2 \gamma_2}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1 - \alpha)]} \right] \\ - (\mu_2^* - [\lambda_2^* + \lambda_3^* (1 - \alpha)]) \left[\frac{2 \gamma_2 j}{\mu_2^* - [\lambda_2^* + \lambda_3^* (1 - \alpha)]} \right] \end{cases}$$

De gevonden oplossing is dus inderdaad een oplossing voor alle situaties.

Q.E.D.

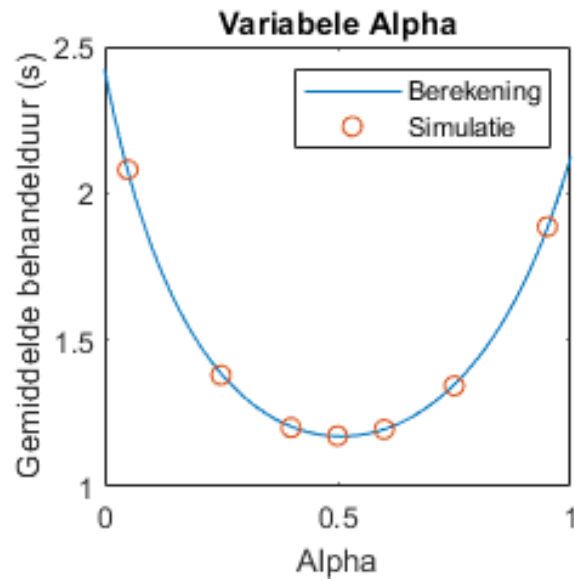
G Simulatiemodellen

Voor het opstellen van een simulatiemodel, zijn de volgende modellen gemaakt:

- Model 1: Alleen aankomsten van type 1, zonder energieverbruik (Bernoulli policy)
- Model 2: Alleen aankomsten van type 2, zonder energieverbruik (Bernoulli policy)
- Model 3: Model 1 en Model 2 gecombineerd (Bernoulli policy)
- Model 4: Alleen aankomsten van type 3, zonder energieverbruik (Bernoulli policy)
- Model 5: Model 3 en Model 4 gecombineerd (Bernoulli policy)
- Model 6: Model 1 inclusief energieverbruik (Bernoulli policy)
- Model 7: Model 4 inclusief energieverbruik (Bernoulli policy)
- Model 8: Model 5 inclusief energieverbruik (Bernoulli policy)
- Model 9: Model 7 aangepast voor 'one-step policy improvement'
- Model 10: Model 8 aangepast voor 'one-step policy improvement'

Aangezien de eerste modellen niet interessant genoeg zijn om uitgebreid te behandelen, wordt er in dit verslag alleen gewerkt met Model 6, Model 7, Model 8, Model 9, en Model 10.

H Extra resultaten voor type 1, type 2 en type 3 aankomsten



Figuur 9: Gemiddelde behandelduur in seconden bij een variabele α . Hierbij zijn de waarden van $\lambda_1 (= 0.8)$, $\lambda_2 (= 0.5)$, $\lambda_3 (= 1)$, $\mu_1 (= 2.2)$ en $\mu_2 (= 1.8)$ een constante.

		Gemiddelde behandelduur (s)	
α		Berekening	95% B.I. Simulatie
0.05		2.0750	[2.0385 , 2.1229]
0.25		1.3851	[1.3641 , 1.3962]
0.4		1.2050	[1.1913 , 1.2106]
0.5		1.1715	[1.1641 , 1.1815]
0.6		1.1957	[1.1853 , 1.2043]
0.75		1.3473	[1.3268 , 1.3602]
0.95		1.8821	[1.8523 , 1.9207]
$\hat{\alpha}_t$	0.5074	1.1713	[1.1632 , 1.1817]
OSPI _t ($\hat{\alpha}_t$)		X	[0.9100 , 0.9213]

Tabel 10: Gemiddelde behandelduur in seconden bij een variabele α . Hierbij zijn de waarden van $\lambda_1 (= 0.8)$, $\lambda_2 (= 0.5)$, $\lambda_3 (= 1)$, $\mu_1 (= 2.2)$ en $\mu_2 (= 1.8)$ een constante.