

De druk van Nivolumab op resources van het Antoni van Leeuwenhoekziekenhuis

N.P.E. SCHUERMANS

University of Twente

27 juni 2016

Samenvatting

Voor de behandeling van Nivolumab moeten patiënten iedere twee weken naar het ziekenhuis. Iedere week moet de patiënt naar de longarts of verpleegkundig specialist, het priklab, de dagbehandeling en vijf keer per jaar naar de CT-scan. Hierdoor is dit behandelingsproces intensief. Het huidige behandelingsproces is een jaar, maar het is mogelijk dat dit een half jaar wordt. Het knelpunt van deze behandeling is de dagbehandeling. Het behandelingsproces wordt gemodelleerd met behulp van wachtrijtheorie en gesimuleerd met behulp van discrete event simulation. Wanneer het huidige behandelingsproces bekeken wordt, is de maximale aankomstintensiteit van nieuwe patiënten gemiddeld vijf bij een capaciteit van één server per station. Wanneer het behandelingsproces van maximaal een half jaar bekeken wordt, is de maximale aankomstintensiteit van nieuwe patiënten gemiddeld zeven bij een capaciteit van één server per station. Wanneer er dynamisch patiënten opgeroepen worden, is de afwijking van de benodigde capaciteit dan ook kleiner en is de maximale bezetting van de dagbehandeling in een week lager, dan bij een vaste aankomstintensiteit van patiënten.

Trefwoorden: Nivolumab, netwerk van wachtrijen, wachtrijtheorie, discrete event simulation

1. INLEIDING

Wanneer iemand ziek wordt, wil deze persoon waarschijnlijk zo snel mogelijk de behandeling met de beste vooruitzichten. Nivolumab is hier een voorbeeld van voor patiënten met tweedelijs niet-kleincellige longkanker. Tweedelijs longkanker betekent dat een patiënt voor de tweede keer longkanker heeft. Niet-kleincellige longkanker is de meest voorkomende variant van longkanker. Deze tumorcellen zijn niet kleiner dan normale longcellen, vandaar de naam. Ten opzichte van chemotherapie heeft een behandeling met Nivolumab een aantal voordelen, zoals het aantal patiënten met bijwerkingen. Dit aantal ligt bij een behandeling met Nivolumab veel lager dan bij chemotherapie. Bij chemotherapie heeft 1 op de 10 patiënten geen bijwerkingen en bij een behandeling met Nivolumab heeft 1 op de 14 patiënten last van bijwerkingen. Verder is de verwachte levens-

duur bij een behandeling met Nivolumab ook hoger dan bij chemotherapie. Het is dus wel duidelijk dat patiënten met tweedelijs niet-kleincellige longkanker behandeld willen worden met Nivolumab. Dit is alleen nog niet voor alle patiënten mogelijk, omdat het een intensief behandelingsproces is. De patiënt moet namelijk om de week naar het ziekenhuis. De patiënt heeft om en om controle bij de longarts en bij de verpleegkundig specialist, moet iedere keer naar de dagbehandeling voor het infuus en een lab analyse laten doen, zodat het effect van het medicijn onderzocht kan worden. Verder moet de patiënt ook nog een paar keer een CT-scan laten doen. Jaarlijks zijn er ongeveer 12.000 nieuwe potentiële patiënten, waarvan er uiteindelijk 2.000 tot 3.000 patiënten geschikt zijn voor de behandeling. In Nederland zijn er 13 centra die een behandeling van Nivolumab aanbieden, waarvan het Antoni van Leeuwenhoekziekenhuis er een is.

De data is verkregen in samenwerking met het Antoni van Leeuwenhoekziekenhuis. Zij denken dat door dit intensieve behandelingsproces er een grote druk op de resources komt te staan. Hierdoor worden er vijf nieuwe patiënten per week behandeld met Nivolumab, maar dat is dus niet genoeg om alle patiënten die dit eigenlijk wel willen te behandelen. Het doel van dit onderzoek is het modelleren van dit proces, zodat er meer inzicht verkregen wordt. Hierdoor kan er een betere inschatting gemaakt worden van de benodigde capaciteit per afdeling van het behandelingsproces. Zo kan er een betere inschatting gemaakt worden over de hoeveelheid patiënten die behandeld kunnen worden met Nivolumab. Verder ga ik nog kijken of er meer patiënten behandeld kunnen worden op het moment dat er dynamisch patiënten opgeroepen worden voor de behandeling. Dit betekent dat het aantal patiënten die opgeroepen worden per behandelingsweek afhangt van het aantal patiënten waarbij de behandeling stop is gezet.

In hoofdstuk 2 wordt eerst gekeken naar de wiskunde in de gezondheidszorg. In hoofdstuk 3 wordt het behandelingsproces in het Antoni van Leeuwenhoekziekenhuis beschreven. Vervolgens wordt er in hoofdstuk 4 een wiskundig model opgesteld met behulp van wachtrijtheorie en een netwerk van wachtrijen. Waarbij patiënten in het netwerk komen op het moment dat ze behandeld worden met Nivolumab en het netwerk verlaten op het moment dat ze stoppen met de behandeling. Vervolgens is er in een verfijnd model opgesteld met behulp van discrete event simulation.

In hoofdstuk 5 worden de resultaten weergegeven. Tenslotte zijn in hoofdstuk 6 en hoofdstuk 7 respectievelijk de discussie en de conclusie.

2. LITERATUURONDERZOEK

In dit hoofdstuk wordt er een literatuuronderzoek gedaan naar de wiskunde in de gezondheidszorg. Uit [1] volgt dat Operations Research een toepassing heeft in de gezondheidszorg en dat hierbij vaak gebruik gemaakt wordt van wachtrijtheorie. Dit komt omdat wachtrij-

modellen relatief weinig data nodig hebben en ze zijn eenvoudig en snel te gebruiken. Wachtrijtheorie wordt op verschillende systemen in de gezondheidszorg toegepast, zoals in ziekenhuizen, de ambulances en de farmaceutische industrie. Dit onderzoek gaat over het behandelingsproces van Nivolumab en dit is een proces in een ziekenhuis. Hierdoor wordt er voor gekozen om meer onderzoek te doen naar de wachtrijtheorie toegepast op ziekenhuizen. In [6] wordt er een model beschreven die de situatie van de patiënten in de wachtrij meeneemt. In dit artikel wordt er namelijk een model gemaakt van een wachtrij van patiënten die wachten op een medische behandeling om de ziekte te behandelen of de omstandigheden te verlichten. Hierbij wordt rekening gehouden met de condities van de patiënt. Dit heeft namelijk invloed op de te krijgen behandeling. De servicetijd kan hierdoor beïnvloed worden, maar de behandeling kan ook niet meer nodig zijn. Hierdoor ontstaat er een wachtrijmodel met prioriteiten. Dit deel van het behandelingsproces wordt in dit onderzoek niet meegenomen. Er wordt niet gekeken naar welke patiënten er wel en welke patiënten er niet toegelaten worden tot het behandelingsproces.

In [3] wordt er een wachtrijmodel opgesteld voor de eerste hulp afdeling. In dit artikel wordt aangetoond dat een personeelsschema bij de eerste hulp effectiever is op het moment dat deze opgesteld wordt met behulp van wachtrijtheorie. Dit wordt gedaan met behulp van een M/M/s model voor de eerste hulp, waarbij aangenomen wordt dat er een oneindige wachtruimte. Er wordt vervolgens gebruik gemaakt van SIPP (stationaire onafhankelijke periode na periode) benadering voor de aankomstprocessen, aangezien deze niet constant zijn gedurende de dag. In dit onderzoek wordt geen planning voor het personeel gemaakt, maar gekeken naar de druk op de resources. Verder wordt er ook uitgegaan van een stationaire aankomstintensiteit, aangezien de patiënten ingepland kunnen worden.

In artikel [5] wordt een model gemaakt met behulp van een netwerk van wachtrijen.

Het doel is om de wachttijd van patiënten met een afspraak te verminderen. Om dit te doen is er de software tool QNA gebruikt voor het analyseren van het netwerk van wachtrijen. Voor ieder onderdeel van het netwerk wordt een kans opgesteld dat een patiënt na dat onderdeel naar een volgend onderdeel van het netwerk gaat. Zo kunnen de aankomstintensiteiten bepaald worden. In dit onderzoek is er ook een netwerk met verschillende onderdelen. Er zou dus gebruik gemaakt kunnen worden van een netwerk van wachtrijen.

In artikel [4] wordt een netwerk van wachtrijen opgesteld voor de geestelijke gezondheidszorg. Door sluiting van verschillende zorginstellingen worden patiënten geplaatst in zorginstelling waar er soms meer faciliteiten zijn dan nodig. Omdat patiënten in deze zorginstelling wachten op een andere zorginstelling, blokkeren ze deze instelling voor patiënten die dit wel nodig hebben. In dit onderzoek is er geen sprake van blokkering, dus kan dit niet gebruikt worden.

3. PROBLEEMOMSCHRIJVING

In dit hoofdstuk wordt het behandelingsproces in het Antoni van Leeuwenhoekziekenhuis beschreven. Wanneer een patiënt wordt behandeld met Nivolumab moet deze patiënt iedere twee weken naar het ziekenhuis. Er zijn dus 26 behandelingsweken per jaar voor een patiënt die met Nivolumab behandeld wordt. De ene keer komt de patiënt op controle bij de longarts en de andere keer bij de verpleegkundig specialist. Dus iedere vier weken komt deze patiënt bij de longarts en de andere keren bij de verpleegkundig specialist. Iedere behandelweek begint met een controle, vervolgens gaat de patiënt naar het priklab, zodat de toestand van de patiënt bekeken kan worden. Daarna gaat een patiënt naar de dagbehandeling voor het infuus. Het kan zijn dat de controle bij de longarts niet doorgaat. Dit kan ook vervangen worden door een belafpraak of kan helemaal vervallen. Dit is zodat er geen onnodige consulten zijn. Vijf keer per jaar gaat de patiënt

ook nog naar de CT-scan, zodat er gekeken kan worden of de behandeling werkt. Wanneer een patiënt nieuw is, gaat deze patiënt naar de longarts voor een intake van gemiddeld 45 minuten. Een gewone controle bij de longarts of verpleegkundig specialist duurt gemiddeld 15 minuten. Het priklab duurt gemiddeld 20 minuten, bij de dagbehandeling is er gemiddeld 45 minuten een bed nodig en de CT-scan 30 minuten. De strategie van het AVL om patiënten toe te laten voor de behandeling is als volgt, iedere week worden er vijf nieuwe patiënten toegelaten. Dus in 2016 worden er 260 patiënten toegelaten voor de behandeling met Nivolumab. Het knelpunt in deze behandeling is volgens het Antoni van Leeuwenhoekziekenhuis de afspraken bij de dagbehandeling. Het is wenselijk voor de patiënt om de afspraak bij de longarts of verpleegkundig specialist, de afspraken van het priklab, de afspraak op de dagbehandeling en eventueel de CT-scan op dezelfde dag te hebben, helaas lukt dit maar in 80% van de gevallen. Maar de gehele behandeling vind wel plaats in dezelfde week. Er wordt alleen met de behandeling gestopt op het moment dat er een CT-scan geweest is. De patiënt stopt dan met de behandeling nadat deze patiënt bij de longarts is geweest. Nu worden de patiënten nog maximaal een jaar behandeld, maar dit wordt waarschijnlijk korter. Een patiënt kan dan waarschijnlijk nog maximaal een half jaar behandeld worden.

De openingstijden verschillen per station. Deze zijn hieronder weergegeven.

Longarts: 8:30-14:30 uur

Priklab: 7:30-17:30 uur

Dagbehandeling: 8:00-16:00 uur

Verpleegkundig specialist: 8:30-14:30 uur

CT-scan: 8:00-16:00 uur

Verder zijn alle stations vijf dagen per week geopend, dus gewoon maandag tot en met vrijdag, behalve de CT-scan. De CT-scan is namelijk ook nog op zaterdag geopend.

De week begint op maandag en alle patiënten beginnen bij de longarts of bij de verpleegkundig specialist.

Dus alle stations die maandag eerder open zijn krijgen geen patiënten van de behandeling van Nivolumab. Dus deze capaciteit wordt niet meegenomen. Hieruit volgt dat de week begint op maandag om 8:30 uur. Verder geldt dat de opening van het priklab van vrijdag 16:00-17:30 uur ook niet bezocht wordt door de patiënten die behandeld worden met Nivolumab. De patiënten doen namelijk niet langer dan een week over de behandelingsweek en na het priklab moet de patiënt altijd nog naar de dagbehandeling. Hieruit kan bepaald worden hoe lang ieder station geopend is voor de behandeling van Nivolumab per week.

longarts: 30 uur per week

Priklab: 46.5 uur per week

Dagbehandeling: 40 uur per week

Verpleegkundig specialist: 30 uur per week

CT-scan: 48 uur per week

Nu kunnen de gemiddelde bedieningsduren zoals hierboven in minuten gegeven zijn uitgedrukt worden per week

Intake longarts: 0.025 werkweek

Controle longarts: 0.008 werkweek

Priklab: 0.007 werkweek

Dagbehandeling: 0.019 werkweek

Verpleegkundig specialist: 0.008 werkweek

CT-scan: 0.01 werkweek

Er zijn natuurlijk ook meerdere servers per afdeling, zoals hieronder te zien is.

longarts: 8 servers

Priklab: 4 servers

Dagbehandeling: 48 servers

Verpleegkundig specialist: 3 servers

CT-scan: 2 servers

De hierboven beschreven capaciteit is de capaciteit van het Antoni van Leeuwenhoekziekenhuis, maar deze capaciteit is natuurlijk niet geheel beschikbaar voor de behandeling van Nivolumab. Er zijn natuurlijk nog veel meer behandelingen die ook gebruikt maken van deze capaciteit. De onderzoeksvraag van dit verslag: Hoe groot is de druk van de behandeling van Nivolumab op de resources? Hiervoor zijn een aantal deelvragen die beantwoord

worden.

Hoeveel servers zijn er per station nodig om vijf nieuwe patiënten per week op te roepen bij het huidige behandelingsproces?

Kunnen er meerdere patiënten per week opgeroepen worden bij deze capaciteit?

Hoeveel servers zijn er per station nodig om vijf nieuwe patiënten per week op te roepen bij een behandelingsproces van maximaal een half jaar?

Kunnen er meerdere patiënten per week opgeroepen worden bij deze capaciteit?

Wat is het effect van dynamisch patiënten oproepen op de capaciteit?

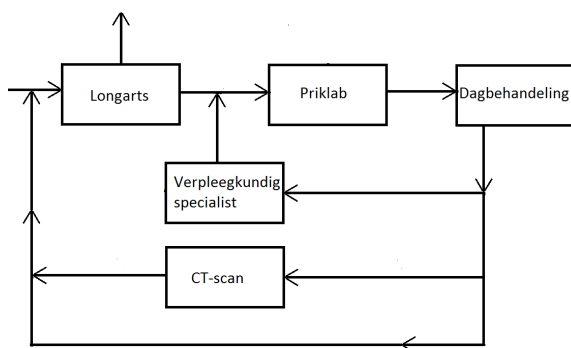
4. MODEL

In dit hoofdstuk wordt een wiskundig model gemaakt voor de situatie zoals beschreven in hoofdstuk 3. In lid 4.1 wordt er een analytisch model beschreven waarbij er gebruik gemaakt wordt van een netwerk van wachtrijen. Hierbij heeft iedere afdeling van het behandelingsproces een wachtrij en zijn er verschillende mogelijkheden om door dit systeem te gaan. Vervolgens wordt er in lid 4.2 een simulatie gemaakt met behulp van discrete event simulation. Daarna wordt er in lid 4.3 een methode beschreven zodat er bepaald kan worden hoeveel patiënten er maximaal per week opgeroepen kunnen worden. Tenslotte wordt er in lid 4.4 een strategie besproken om patiënten dynamisch op te roepen. In Tabel 11 in Appendix A is een lijst te zien van de gebruikte symbolen te zien.

4.1. ANALYTISCH MODEL

In Figuur 1 is een weergave van de situatie zoals beschreven in hoofdstuk 3. Patiënten kunnen met verschillende paden door dit systeem lopen. Hierdoor ontstaan er verschillende type patiënten. Wanneer het behandelingsproces maximaal een jaar duurt zijn er natuurlijk meer verschillende type patiënten dan wanneer het behandelingsproces maximaal een half jaar duurt. Ieder vakje in Figuur 1 is een wachtrij.

De longarts wordt in het vervolg station 1 genoemd, het priklab station 2, de dagbehandeling station 3, de verpleegkundig specialist station 4 en de CT-scan station 5.



Figuur 1: Een weergave van het behandelingsproces.

Om een analytisch model op te stellen zijn de volgende aannames gemaakt.

- Patiënten worden het gehele behandelingsproces niet bij dezelfde arts/ verpleegkundig specialist behandeld.
- Een nieuwe patiënt krijgt in plaats van een intake een gewone controle, hierdoor is de gemiddelde bedieningsduur bij de longarts voor iedere patiënt gelijk.
- De kans om het systeem te verlaten is geometrisch verdeeld.
- De bedieningsduren zijn exponentieel verdeeld.
- De bedieningsduren zijn onderling onafhankelijk en identiek verdeeld.
- De patiënten zijn altijd beschikbaar voor de behandeling.
- De bedieningsvolgorde is FIFO.
- De looptijden tussen de verschillende stations is nul.
- Het externe aankomstproces bij de longarts is Poisson.
- Ieder station beschikt over een oneindige wachtruimte.
- Patiënten gaan iedere vier weken naar de longarts.

De eerste aanname is niet een grote vereenvoudiging, aangezien dit in het Antoni van Leeuwenhoekziekenhuis nu ook niet het geval is. In het Antoni van Leeuwenhoek zieken-

huis wordt natuurlijk wel geprobeerd om de patiënten zo veel mogelijk bij dezelfde arts/ verpleegkundig specialist terug te laten komen, maar dat gebeurt niet in dit model. Aangezien er naar de algemene capaciteit gekeken wordt en niet naar de capaciteit per server heeft dit geen invloed. De tweede aanname dat een patiënt een controle in plaats van een intake krijgt, is wel een aanname die invloed heeft op de capaciteit. Dit verschil is namelijk gemiddeld 30 minuten per bediening. Deze aanname is gemaakt, omdat bij een netwerk van wachtrijen met daarin verschillende types iedere patiënt op ieder moment dezelfde gemiddelde bedieningsduur moet hebben bij een station. Er zou gekozen kunnen worden voor een netwerk met daarin twee stations voor de longarts, een met de controle en een met de intake. In dit geval zijn er echter twee verschillende wachtrijen voor dezelfde afdeling. Er is gekozen voor een exponentiële verdeling voor de bedieningsduur en een Poisson aankomstproces, omdat dit de berekeningen een stuk eenvoudiger maakt. Later in de simulatie zullen deze aannames niet meer gemaakt worden. De bedieningsvolgorde is FIFO, aangezien de patiënt die vooraan in de wachtrij staat als eerste behandeld wordt. Er zijn natuurlijk ook looptijden tussen de verschillende stations of tijden die tussen de verschillende stations moeten zitten, bijvoorbeeld voordat de uitslag van het priklab bekend is. Dit wordt verwaarloosd met deze aanname. De aanname dat ieder station een oneindige wachtruimte heeft, is gemaakt om te voorkomen dat een patiënt naar huis wordt gestuurd omdat de wachtruimte vol is. Er wordt hiermee dus aangenomen dat een patiënt altijd in de wachtruimte terecht kan. Tenslotte wordt er aangenomen dat de patiënt iedere vier weken naar de longarts gaat, dus deze afspraak kan niet vervangen worden door een belafpraak of geheel vervallen. Hierdoor is de benodigde capaciteit bij de longarts wat hoger dan de werkelijke benodigde capaciteit. Er is een tijdschaal van een week.

Iedere patiënt moet namelijk iedere twee weken behandeld worden, dus de patiënten die in week één behandeld worden zullen de patiënten die in week twee behandeld worden nooit tegenkomen in het behandelproces. Daarom kan er een tijdschaal van een week genomen worden, waarbij de week tussen de behandelingsweken genegeerd wordt. Dit proces verloopt twee keer tegelijk onafhankelijk van elkaar, namelijk de patiënten van oneven weken en patiënten van de even weken.

Verspreid over de 26 behandelingsweken worden de vijf CT-scans gepland, zoals hieronder te zien is. Een patiënt kan het systeem alleen verlaten na een controle bij de longarts met voorafgaand een CT-scan. Hierdoor wordt het behandelingsproces opgesplitst in de volgende zes delen.

1. L-V-L-V+CT-L
2. V-L-V+CT-L
3. V-L-V+CT-L
4. V-L-V+CT-L
5. V-L-V+CT-L
6. V-L-V-L-V-longarts

Hierbij staat de V voor een behandelweek waarbij de controle bij de verpleegkundige specialist is, de L voor een behandelweek waarbij de controle bij de longarts is en V+CT een behandelweek waarbij de controle bij de verpleegkundig specialist is en de patiënt in deze week ook een CT-scan krijgt. De L na een CT-scan is alleen de longarts en de rest van de behandelingen in die week vallen onder het volgende deel, aangezien patiënten na de longarts het systeem kunnen verlaten. De longarts aan het einde van deel 6 wil zeggen dat daar alleen nog maar een controle bij de longarts is.

Zoals te zien is, is hierbij aangenomen dat deel 1 en 6 wat langer zijn dan de delen 2, 3, 4 en 5. Dit komt omdat bij deel 1 het pad begint bij de longarts en het ook moet eindigen bij de longarts. De delen 2, 3, 4 en 5 beginnen bij de verpleegkundig specialist en eindigen bij de longarts. Verder zijn 26 behandelweken en hieruit volgt de lengte van deel 6. De behandeling kan alleen beëindigd worden op het moment dat de patiënt bij de CT-scan is geweest. Dus

er zijn de volgende type patiënten, namelijk de patiënten die de volgende paden door het systeem afleggen.

Maar op het moment dat een patiënt al voor de eerste CT-scan heel slecht op Nivolumab reageert, wordt er natuurlijk al eerder een CT-scan gemaakt. Dus eigenlijk zou een CT-scan voor ieder bezoek van de longarts gepland kunnen worden. Er geldt wel dat de CT-scans verdeeld over het jaar gepland blijven, wanneer er geen aanleiding is om deze eerder te plannen. Er wordt verder aangenomen dat, wanneer de CT-scan vervroegd wordt, de patiënt ook het systeem verlaat. Er wordt niet meegenomen dat de vervroegde CT-scan ook positief kan zijn en de patiënt dus in het systeem kan blijven. Hieruit volgens dus verschillende type patiënten, namelijk de verschillende paden door het systeem. Er kan nu voor een nieuwe patiënt in het systeem bepaald worden hoe groot de kans is dat deze patiënt van een bepaald type is. Dit wordt gedaan met behulp van de geometrische verdeling zoals te zien is in geval (1a). Er is echter nog een geval (1b), namelijk op het moment dat de patiënt aan het einde van het behandelingsproces is gekomen. Hierbij is N het aantal verschillende type patiënten. Voor het type patiënt die het langst in het behandelingsproces zit, dus het geval (1b), geldt dat deze patiënt het systeem verlaat omdat het proces dan eindigt. Dus deze patiënt bleef $n - 1$ keer in het systeem en verlaat dan automatisch het systeem. In deze formule staat y voor de kans op de verschillende types.

$$y(n) = \begin{cases} p(1-p)^{n-1} & \text{voor } n < N \text{ (1a)} \\ (1-p)^{n-1} & \text{voor } n = N \text{ (1b)} \end{cases}$$

Hierbij geldt dat p de kans is dat een patiënt het systeem verlaat. Dus $1 - p$ is de kans dat een patiënt in het systeem blijft. Verder geldt dat N het aantal verschillende type patiënten is. Wanneer de kans op de verschillende type patiënten bekend is, kan de aankomstintensiteit van de verschillende type patiënten per week bepaald worden. Dit is namelijk de kans op het type vermenigvuldigd met de aankomstintensiteit van nieuwe patiënten per week.

Vervolgens wordt er een model gemaakt met de verschillende type patiënten [2]. Hierbij wordt voor ieder station aangenomen dat er één server is. Om dit model te maken wordt er aangenomen dat de patiënten in wachtrij j ($j=1,2,\dots,J$) gerangschikt zijn. Dit wil zeggen dat wachtrij j patiënten heeft op posities $1,2,\dots,n_j$. Waarbij n_j het aantal patiënten in wachtrij j is. In wachtrij j heeft de volgende eigenschappen:

- Patiënten hebben een willekeurige bedieningsduur die exponentieel verdeeld is met een gemiddelde ($\frac{1}{\mu_j}$).
- De totale service prestatie $\phi_j(n_j) = 1$, want er is één server.
- Een deel $\gamma_j(l, n_j)$ van deze prestatie komt van de patiënt op positie l ($l=1,2,\dots,n_j$). In het algemeen geldt

$$\gamma_j(l, n) = \begin{cases} \frac{1}{n} & l = 1, 2, \dots, n; \\ & n = 1, 2, \dots, K \\ \frac{1}{K} & l = 1, 2, \dots, K; \\ & n = K + 1, K + 2, \dots \\ 0 & \text{anders} \end{cases}$$

Hierbij geldt dat er K servers per station zijn. In dit model is er bij ieder station één server, dus $K=1$. Hieruit volgt de $\gamma_j(l, n)$ zoals hieronder weergegeven

$$\gamma_j(l, n) = \begin{cases} 1 & l = 1; \\ & n = 1, 2, 3, \dots \\ 0 & \text{anders} \end{cases}$$

Wanneer de patiënt op positie l de wachtrij verlaat, dit is het geval als zijn behandeling van wachtrij j volbracht is, verschuiven patiënten van positie $l+1, l+2, \dots, n_j$ naar de posities $l, l+1, \dots, n_j$. Aangezien de bedieningsvolgorde FIFO is, geldt dat alleen de patiënt op positie 1 de wachtrij kan verlaten.

- Wanneer een patiënt arriveert in wachtrij j wordt zijn positie l met kans $\delta_j(l, n_j + 1)$. Aangezien een nieuwe patiënt in de wachtrij altijd de laatste positie krijgt geldt $\delta_j(n_j + 1, n_j + 1) = 1$ en voor alle andere posities l geldt $\delta_j(l, n_j + 1) = 0$.

Noem totale bediening die een patiënt bij een wachtrij nodig heeft zijn benodigde bediening. Vervolgens wordt er aangenomen dat de benodigde bediening van patiënten onafhankelijk is van elkaar en van het tijdstip waarop de patiënt in het systeem komt. Neem nu $t_j(l)$ het type patiënt in positie l van wachtrij j en zij $s_j(l)$ de bijbehorende fase van de route van de patiënt. Noem $c_j(t_j(l), s_j(l))$ de klasse van deze patiënt. Aangezien patiënten vaker dezelfde wachtrij bezoeken, geeft de klasse van een patiënt meer informatie dan alleen het type patiënt. De vector $\mathbf{c}_j = (c_j(1), c_j(2), \dots, c_j(n_j))$ beschrijft de toestand van wachtrij j en $\mathbf{C} = (\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_J)$ is een Markov proces representatie van de toestand van het systeem. Definieer $\alpha_j(i, s)$ en a_j als volgt

$$\alpha_j(i, s) = \begin{cases} \nu(i) & \text{als } r(i, s) = j \\ 0 & \text{anders} \end{cases}$$

en

$$a_j = \sum_{i=1}^I \sum_{s=1}^{S(i)} \alpha_j(i, s)$$

Wanneer het systeem in evenwicht is, is a_j het gemiddelde aantal patiënten dat arriveert bij wachtrij j per tijdseenheid. Zij

$$b_j^{-1} = \sum_{n=0}^{\infty} \frac{a_j^n}{\prod_{l=1}^n \phi_j(l)}$$

Vervolgens wordt er aangenomen dat $b_1, b_2, \dots, b_J$ niet gelijk aan nul zijn. Deze aanname is nodig om zeker te weten dat de evenwichtsverdeling bestaat. Definieer

$$\pi_j(\mathbf{c}_j) = b_j \prod_{l=1}^{n_j} \frac{\alpha_j(t_j(l), s_j(l))}{\phi_j(l)}$$

Dan is de evenwichtsverdeling van het netwerk gelijk aan

$$\pi(\mathbf{C}) = \prod_{j=1}^J \pi_j(\mathbf{c}_j) \quad (2)$$

Uit 2 kan afgeleid worden dat ieder station j kan opgevat worden als een onafhankelijk M/M/1 station, wanneer het stationair is.

Dit komt omdat de stationaire verdeling van het systeem het product van de marginalen is. Dus geldt dat er op een vast moment, wanneer het systeem in evenwicht is, dat het aantal klanten bij de stations onafhankelijk is. Hieruit volgt dus dat ieder station apart bekeken kan worden.

De aankomstintensiteit moet hiervoor per station bepaald worden. Dit kan gedaan worden door het aantal keer dat een type patiënt een bepaald station bezoekt in het behandelingsproces te bekijken. Vervolgens kunnen nu de stromingsvergelijkingen opgesteld worden en hieruit volgen de aankomstintensiteiten per station.

Nu kan er bepaald worden wanneer het systeem in evenwicht is. Dit is het geval, wanneer alle stations in evenwicht zijn. Aangezien de stations M/M/s zijn, geldt er dat de stations in evenwicht zijn op het moment dat $\rho = \frac{\lambda}{\mu} < s$. De gemiddelde aankomstintensiteit ligt vast, maar de gemiddelde bedieningsduur kan gevarieerd worden door de capaciteit. Neem hiervoor aan dat een M/M/s station j gezien kan worden als een M/M/1 station met een gemiddelde bedieningsduur $\frac{1}{s \cdot \mu_j}$. Hierbij geldt dat $\frac{1}{\mu_j}$ de gemiddelde bedieningsduur is wanneer er maar een server is. Dit is natuurlijk niet hetzelfde, want de snelheid van de bediening wordt dus s keer zo snel. Op het moment dat er bij een station met $s > 1$ servers een server geen patiënt in behandeling heeft, wordt deze wel als werkend beschouwd in deze aanname. Wanneer $s < 1$ kan dit worden gezien als 1 server met een kortere werkweek. De werkweek is dan $s \cdot$ normale werkweek.

Vervolgens wordt de formule van Little, (3), toegepast om de gemiddelde verblijftijd te bepalen. Hierbij is $E\{L\}$ het gemiddeld aantal patiënten in systeem en $E\{F\}$ de gemiddelde verblijftijd per patiënt [7].

$$E\{L\} = \lambda \cdot E\{F\} \quad (3)$$

Aangezien het model uit M/M/1 stations bestaat, kan voor ieder M/M/1 station (4) gebruikt worden om de gemiddelde verblijftijd

te bepalen [7].

$$E\{L\} = \lambda \cdot E\{F\} = \frac{\rho}{1 - \rho} \quad (4)$$

Nu kan er ook bekeken hoe groot de invloed van het aantal nieuwen patiënten dat iedere week op wordt geroepen is op de gemiddelde verblijftijd per station. Om dit te kunnen doen moet het systeem wel in evenwicht zijn.

4.2. VERFIJND MODEL

In deze paragraaf wordt het verfijnde model beschreven. Het verfijnde model is een simulatie met behulp van discrete event simulation. Dit is een simulatiemethode waarbij er een simulatieklok bijgehouden wordt en deze simulatieklok iedere keer verhoogd wordt naar het tijdstip waarop de eerst volgende gebeurtenis plaatsvindt. Dus de simulatieklok gaat van gebeurtenis naar gebeurtenis. Om dit te kunnen doen moeten de verschillende gebeurtenissen die in het systeem voorkomen beschreven worden en precies bepaald worden welke stappen er bij iedere gebeurtenis doorlopen moeten worden. Hieronder zijn de verschillende gebeurtenissen weergegeven.

1. Een aankomst bij de longarts
2. Een vertrek bij de longarts
3. Een aankomst bij het priklab
4. Een vertrek bij het priklab
5. Een aankomst bij de dagbehandeling
6. Een vertrek bij de dagbehandeling
7. Een aankomst bij de verpleegkundig specialist
8. Een vertrek bij de verpleegkundig specialist
9. Een aankomst bij de CT-scan
10. Een vertrek bij de CT-scan

In Appendix B.1 is er beschreven wat de stappen zijn per gebeurtenis. Ook is er een flowchart gemaakt. Deze flowcharts zijn te zien in Appendix B.3.

De simulatie wordt gedaan voor een week, dus de tijd loopt tot het einde van de week. Na deze week wordt de tijd weer op nul gesteld en kan er een nieuwe week gesimuleerd worden. Hieruit volgt dat de tijdschaal van de simulatie een week is.

Er is voor gekozen om aan het einde van deze week de aankomstgebeurtenissen bij de longarts en verpleegkundig specialist op oneindig te stellen, hierdoor komen er aan het einde van de week geen nieuwe patiënten meer in het behandelingsproces en kunnen de patiënten die al in het behandelingsproces van deze behandelingsweek zitten de behandelingen van deze week ook afmaken. Wanneer alle patiënten de behandelingsweek beëindigd hebben, zijn alle geplande gebeurtenissen oneindig en wordt de simulatie van deze week beëindigd. De tijdschaal van een week was bij het analytische model ook het geval.

Nu moet er gekeken worden welke stations wanneer open zijn en wanneer de patiënten met de behandeling van Nivolumab hier terecht kunnen. Zo kunnen de patiënten maar tot 14:30 uur naar de verpleegkundig specialist of longarts komen, want zo is de kans groter dat ze nog meer stations van het behandelingsproces op deze dag kunnen bezoeken. Terwijl het voor andere patiënten wel mogelijk is om later bij de verpleegkundig specialist of longarts te komen. De week begint op maandag en alle patiënten beginnen bij de longarts of bij de verpleegkundig specialist. Dus alle stations die maandag eerder open zijn, krijgen geen patiënten van de behandeling van Nivolumab. Deze capaciteit wordt dus niet meegenomen. Hieruit volgt dat de week begint op maandag om 8:30 uur. Wanneer een station gesloten is, wordt er een tijd bij de volgende geplande aankomst opgeteld, zodat deze pas weer plaatsvindt op het moment dat het station geopend is. De tijd die er bij op geteld wordt, is dan natuurlijk de tijd dat het station gesloten is. Wanneer er nog een patiënt in bediening is, wordt de behandeling van deze patiënt nog afgemaakt. Daarna komen geen patiënten meer, dus zijn er ook geen gebeurtenissen van vertrek bij dit station meer in de tijd dat dit station gesloten is. Dus er zijn geen nieuwe aankomsten gedurende de sluiting van een station, maar er wordt aangenomen dat de patiënten in de wachtrij wel geholpen worden in deze periode. Een overzicht van welk station geopend is voor ieder uur is weergegeven in Appendix B.2. Er

wordt aangenomen dat de dagbehandeling en het prikklab over kunnen werken op zaterdag, zodat alle patiënten de behandelingsweek binnen een week af kunnen maken. Vervolgens kan er aan het einde van de week opgeslagen worden hoeveel patiënten van iedere klasse de volgende behandelingsweek weer terug komen. Deze gegevens kunnen gebruikt worden om de volgende behandelingsweek te simuleren. Na de simulatie van een aantal weken wordt de gemiddelde verblijftijd per patiënt bepaald. Er zijn verschillende behandelingsweken, namelijk de behandelingsweek met een intake bij de longarts, een behandelingsweek met een controle bij de longarts, een behandelingsweek bij de verpleegkundig specialist, een behandelingsweek met een controle bij de verpleegkundig specialist en een behandelingsweek met alleen een controle bij de longarts. Voor ieder van deze behandelingsweken kan een gemiddelde verblijftijd bepaald worden. Deze verblijftijd is in uren dat het ziekenhuis open is, hierbij wordt dus geen rekening met verschillende dagen gehouden. Om dit toch mee te kunnen nemen wordt er bijgehouden hoeveel dagen een patiënt over de behandelingsweek doet. Ook wordt de tijd bijgehouden dat iedere server bezet is. Met iedere server wordt de longarts, het prikklab, de dagbehandeling, de verpleegkundig specialist en de CT-scan bedoeld. Na de simulatie van een aantal weken kan voor iedere server de gemiddelde tijd bepaald worden dat deze server bezet is. Het aankomstproces is niet een Poisson proces, zoals in het analytische model, maar de patiënten komen aan met deterministisch verdeelde tussentijden. Bij een Poisson proces met gemiddeld vijf patiënten per week is het namelijk mogelijk dat er meer of minder dan vijf patiënten per week komen. Wanneer er voor een deterministische verdeling van de tussenaankomsttijden gekozen wordt, is dit niet het geval. Er komen dan namelijk precies vijf patiënten per week. Dit is realistischer, omdat er van een bepaald type in een bepaalde fase n patiënten de volgende behandelingsweek terug moeten komen.

Er moeten dan niet gemiddeld n patiënten van deze klasse volgende behandelingsweek terugkomen, maar precies deze n patiënten. Verder zijn de bedieningsduren niet exponentieel verdeeld, maar deterministisch. Dit is het geval, omdat de exponentiële verdeling een grote variantie heeft. Dit is niet realistisch voor de bedieningsduren. Deterministisch is natuurlijk ook niet de werkelijkheid, want de bedieningsduren van de verschillende patiënten wijken wel wat af en bij een deterministische verdeling zijn de bedieningsduren van patiënten met dezelfde behandeling allemaal hetzelfde.

Nu kan deze simulatie van een aantal weken meerdere keren gedaan worden, ook wel replicaties genoemd. Nadat deze replicaties gemaakt zijn kan er een 95% betrouwbaarheidsinterval opgesteld worden voor de gemiddelde verblijftijden en de bezetting van de stations.

4.3. GEVOELIGHEIDSANALYSE

Met behulp van het analytische model zoals beschreven in paragraaf 4.1 kan een gevoeligheidsanalyse gedaan worden. Hiervoor wordt een vaste capaciteit genomen, vervolgens wordt er gekeken hoeveel nieuwe patiënten er maximaal opgeroepen kunnen worden zodat het station in evenwicht is. Dit wordt voor ieder station gedaan.

4.4. DYNAMISCH PATIËNTEN OPROEPEN

Het knelpunt van dit behandelingsproces is de dagbehandeling. Hierdoor wordt er gekeken of er meer patiënten behandeld kunnen worden op het moment dat er dynamisch patiënten opgeroepen worden. Dit wordt gedaan door iedere week het aantal nieuwe patiënten die opgeroepen worden gelijk te stellen aan het verschil tussen het maximale aankomstintensiteit bij de dagbehandeling zodat deze nog in evenwicht is en het aantal patiënten die nog in het systeem zitten. Er zijn natuurlijk ook nog patiënten die in het systeem zitten en die niet meer bij de dagbehandeling komen. Maar dit aantal is vooraf niet bekend. Er wordt dus voor gekozen om dit niet mee te nemen, aangezien er dan een schatting gedaan moet worden voor

het aantal patiënten die het systeem verlaten per week. Wanneer er een week is dat er minder patiënten het systeem verlaten dan geschat, is het systeem niet meer in evenwicht. Alleen de patiënten die al de maximale duur van het behandelingsproces in behandeling zijn worden hiervoor meegenomen. Van deze patiënten is namelijk bekend dat ze het systeem gaan verlaten. Met behulp van het analytische model zoals beschreven in paragraaf 4.1 wordt de maximale aankomstintensiteit bij de dagbehandeling bepaald. Nu wordt het aantal patiënten dat opgeroepen wordt in de simulatie zoals beschreven in paragraaf 4.2 de maximale aankomstintensiteit bij de dagbehandeling minus het aantal patiënten dat nog in het systeem zit. Vervolgens wordt hier het aantal patiënten bij opgeteld die de maximale duur van de behandeling gehad hebben en dus alleen nog maar naar de longarts gaan.

4.5. HUIDIGE BEHANDELINGSPROCES

In lid 4.5.1 wordt eerst het analytische model toegepast. Vervolgens wordt er in lid 4.5.2 de simulatie toegepast op het huidige behandelingsproces.

4.5.1 Analytisch model

Om de theorie zoals beschreven in paragraaf 4.1 toe te kunnen passen, moet de p in (1a) en (1b) bepaald worden. Er is bekend dat na drie maanden behandelen nog 65% van de patiënten in het systeem is. Dus 35% van de patiënten heeft bij de eerste of de tweede mogelijkheid het systeem verlaten, want dit zijn de enige mogelijkheden om maximaal drie maanden behandeld te worden. De kans dat een patiënt de eerste keer het systeem verlaat, is p . De kans dat een patiënt de tweede keer het systeem verlaat, is de kans dat de patiënt de eerste keer het systeem nog niet verlaat, deze kans gelijk is aan $1 - p$, en de tweede keer het systeem verlaat. Dus deze kans is $p(1 - p)$. Hieruit volgt dat $p + (1 - p)p = 0.35$ en dus $p = 0.2$ of $p = 1.8$, aangezien p een kans is moet p kleiner dan één zijn. Hieruit volgt dat $p = 0.2$.

Met deze p kan de functie voor de kans dat een patiënten van een bepaald type is opgesteld worden. Dit is hieronder te zien.

$$y(n) = \begin{cases} 0.2(0.8)^{n-1} & \text{voor } n < N \quad (5a) \\ 0.8^{n-1} & \text{voor } n = N \quad (5b) \end{cases}$$

Wanneer de behandeling maximaal een jaar duurt, zijn er elf mogelijkheden om door het systeem zoals te zien is in Figuur 1 te lopen. Hieruit ontstaan 11 type patiënten met verschillende paden door het systeem, zoals hieronder te zien is.

- Type 1:
L-V+CT-Longarts
- Type 2:
L-V-L-V+CT-Longarts
- Type 3:
L-V-L-V+CT-L-V-longarts
- Type 4:
L-V-L-V+CT-L-V-L-V+CT-longarts
- Type 5:
L-V-L-V+CT-L-V-L-V+CT-L-V+CT-longarts
- Type 6:
L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-longarts
- Type 7:
L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-L-V+CT-longarts
- Type 8:
L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-longarts
- Type 9:
L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-L-V+CT-longarts
- Type 10:
L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-longarts
- Type 11:
L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-longarts

De precieze route van ieder type patiënt is weergegeven in Tabel 18 en Tabel 19 in Appendix E. In Tabel 18 staat de route voor type 1 tot en met 6 en in Tabel 19 staat de route voor type

7 tot en met 11.

Nu kunnen (5a) en (5b) met $N = 11$ gebruikt worden om de kans te bepalen dat een nieuwe patiënt van een bepaald type is. Door de aankomstintensiteit van gemiddeld vijf per week te gebruiken, kan dan vervolgens de gemiddelde aankomstintensiteit van ieder type patiënt bepaald worden. Dit is gedaan met behulp van MATLAB, zoals te zien is in Appendix F.

- De kans dat een patiënt van type 1 is, is de kans dat de patiënt het systeem bij de eerste mogelijkheid verlaat. Hieruit volgt $y(1) = 0.2(0.8)^0 = 0.2$. Dus de aankomstintensiteit van type 1 patiënten is gemiddeld $5 * 0.2 = 1$ per week.
- De kans dat een patiënt van type 2 is, is de kans dat de patiënt het systeem bij de tweede mogelijkheid verlaat. Hieruit volgt $y(2) = 0.2(0.8)^1 = 0.16$. Dus de aankomstintensiteit van type 2 patiënten is gemiddeld $5 * 0.16 = 0.8$ per week.
- De kans dat een patiënt van type 3 is, is de kans dat de patiënt het systeem bij de derde mogelijkheid verlaat. Hieruit volgt $y(3) = 0.2(0.8)^2 = 0.128$. Dus de aankomstintensiteit van type 3 patiënten is gemiddeld $5 * 0.128 = 0.64$ per week.
- De kans dat een patiënt van type 4 is, is de kans dat de patiënt het systeem bij de vierde mogelijkheid verlaat. Hieruit volgt $y(4) = 0.2(0.8)^3 = 0.1024$. Dus de aankomstintensiteit van type 4 patiënten is gemiddeld $5 * 0.1024 = 0.512$ per week.
- De kans dat een patiënt van type 5 is, is de kans dat de patiënt het systeem bij de vijfde mogelijkheid verlaat. Hieruit volgt $y(5) = 0.2(0.8)^4 = 0.0819$. Dus de aankomstintensiteit van type 5 patiënten is gemiddeld $5 * 0.0819 = 0.4096$ per week.
- De kans dat een patiënt van type 6 is, is de kans dat de patiënt het systeem bij de zesde mogelijkheid verlaat. Hieruit volgt $y(6) = 0.2(0.8)^5 = 0.0655$. Dus de aankomstintensiteit van type 6 patiënten is gemiddeld $5 * 0.0655 = 0.3277$ per week.

- De kans dat een patiënt van type 7 is, is de kans dat de patiënt het systeem bij de zevende mogelijkheid verlaat. Hieruit volgt $y(7) = 0.2(0.8)^6 = 0.0524$. Dus de aankomstintensiteit van type 7 patiënten is gemiddeld $5 * 0.0524 = 0.2621$ per week.
- De kans dat een patiënt van type 8 is, is de kans dat de patiënt het systeem bij de achtste mogelijkheid verlaat. Hieruit volgt $y(8) = 0.2(0.8)^7 = 0.0419$. Dus de aankomstintensiteit van type 8 patiënten is gemiddeld $5 * 0.0419 = 0.2097$ per week.
- De kans dat een patiënt van type 9 is, is de kans dat de patiënt het systeem bij de negende mogelijkheid verlaat. Hieruit volgt $y(9) = 0.2(0.8)^8 = 0.0336$. Dus de aankomstintensiteit van type 9 patiënten is gemiddeld $5 * 0.0336 = 0.1678$ per week.
- De kans dat een patiënt van type 10 is, is de kans dat de patiënt het systeem bij de tiende mogelijkheid verlaat. Hieruit volgt $y(10) = 0.2(0.8)^9 = 0.0268$. Dus de aankomstintensiteit van type 10 patiënten is gemiddeld $5 * 0.0268 = 0.1342$ per week.
- De kans dat een patiënt van type 11 is, is de kans dat de patiënt het systeem bij de elfde mogelijkheid verlaat. Dit is dus de kans dat de patiënt de eerste tien keer in het systeem bleef. Dit is dus gelijk aan $y(11) = 0.8^{10} = 0.1074$. Dus de aankomstintensiteit van type 11 patiënten is gemiddeld $5 * 0.1074 = 0.5369$ per week.

Wanneer nu de stations apart bekeken worden, wordt Tabel 1 opgesteld met daarin het aantal keer dat een type patiënt een bepaald station bezoekt. Dit is gedaan met behulp van Tabel 18 en Tabel 19 in Appendix E. Hierin staat de route door Figuur 1 voor ieder type patiënt. In Tabel 18 staat de route voor type 1 tot en met 6 en in Tabel 19 staat de route voor type 7 tot en met 11. Met behulp van Tabel 1 kunnen de aankomstintensiteiten (λ_j) per station bepaald worden, zoals hiernaast te zien is.

$$\begin{aligned}\lambda_1 &= 2\nu(1) + 3\nu(2) + 4\nu(3) + 5\nu(4) \\ &\quad + 6\nu(5) + 7\nu(6) + 8\nu(7) + 9\nu(8) \\ &\quad + 10\nu(9) + 11\nu(10) + 14\nu(11) \\ &= 28.9\end{aligned}$$

$$\begin{aligned}\lambda_2 &= 2\nu(1) + 4\nu(2) + 6\nu(3) + 8\nu(4) \\ &\quad + 10\nu(5) + 12\nu(6) + 14\nu(7) + 16\nu(8) \\ &\quad + 18\nu(9) + 20\nu(10) + 24\nu(11) \\ &= 46.8\end{aligned}$$

$$\begin{aligned}\lambda_3 &= 2\nu(1) + 4\nu(2) + 6\nu(3) + 8\nu(4) \\ &\quad + 10\nu(5) + 12\nu(6) + 14\nu(7) \\ &\quad + 16\nu(8) + 18\nu(9) + 20\nu(10) \\ &\quad + 24\nu(11) = 46.8\end{aligned}$$

$$\begin{aligned}\lambda_4 &= \nu(1) + 2\nu(2) + 3\nu(3) + 4\nu(4) \\ &\quad + 5\nu(5) + 6\nu(6) + 7\nu(7) + 8\nu(8) \\ &\quad + 9\nu(9) + 10\nu(10) + 13\nu(11) \\ &= 23.9\end{aligned}$$

$$\begin{aligned}\lambda_5 &= \nu(1) + \nu(2) + 2\nu(3) + 2\nu(4) \\ &\quad + 3\nu(5) + 3\nu(6) + 4\nu(7) + 4\nu(8) \\ &\quad + 5\nu(9) + 5\nu(10) + 5\nu(11) \\ &= 12.4\end{aligned}$$

Nu kan er bepaald worden wanneer het systeem in evenwicht is. Dit is het geval, wanneer alle stations in evenwicht zijn. In (6), (7), (8), (9) en (10) is respectievelijk voor de longarts, het priklab, de dagbehandeling, de verpleegkundig specialist en de CT-scan te zien wanneer deze in evenwicht zijn.

$$\rho_1 = \frac{\lambda_1}{s_1 \cdot \mu_1} = \frac{28.9 \cdot 0.008}{s_1} < 1 \quad (6)$$

$$\rho_2 = \frac{\lambda_2}{s_2 \cdot \mu_2} = \frac{46.8 \cdot 0.007}{s_2} < 1 \quad (7)$$

$$\rho_3 = \frac{\lambda_3}{s_3 \cdot \mu_3} = \frac{46.8 \cdot 0.019}{s_3} < 1 \quad (8)$$

$$\rho_4 = \frac{\lambda_4}{s_4 \cdot \mu_4} = \frac{23.9 \cdot 0.008}{s_4} < 1 \quad (9)$$

$$\rho_5 = \frac{\lambda_5}{s_5 \cdot \mu_5} = \frac{12.4 \cdot 0.01}{s_5} < 1 \quad (10)$$

Tabel 1: Het aantal keer dat een patiënt van een bepaald type een wachtrij bezoekt.

Wachtrij	Type 1	Type 2	Type 3	Type 4	Type 5	Type 6	Type 7	Type 8	Type 9	Type 10	Type 11
1	2	3	4	5	6	7	8	9	10	11	14
2	2	4	6	8	10	12	14	16	18	20	24
3	2	4	6	8	10	12	14	16	18	20	24
4	1	2	3	4	5	6	7	8	9	10	13
5	1	1	2	2	3	3	4	4	5	5	5

Uit (6), (7), (8), (9) en (10) kan afgeleid worden hoeveel servers er nodig zijn voor een evenwicht bij respectievelijk de longarts, het priklab, de dagbehandeling, de verpleegkundig specialist en de CT-scan. Dit is te zien in (11), (12), (13), (14) en (15).

$$s_1 > 28.9 \cdot 0.008 = 0.24 \quad (11)$$

$$s_2 > 46.8 \cdot 0.007 = 0.34 \quad (12)$$

$$s_3 > 46.8 \cdot 0.019 = 0.88 \quad (13)$$

$$s_4 > 23.9 \cdot 0.008 = 0.20 \quad (14)$$

$$s_5 > 12.4 \cdot 0.01 = 0.26 \quad (15)$$

Hieruit volgt dus dat er een longarts minimaal 7.2 uur, het priklab 15.6 uur, de dagbehandeling 35.1 uur, de verpleegkundig specialist 6.0 uur en de CT-scan 6.2 uur een server beschikbaar moeten hebben.

4.5.2 Simulatie

Voor de simulatie kunnen de type patiënten en de kans op deze type patiënten zoals beschreven in 4.5.1 gebruikt worden. Hiervoor moet natuurlijk ook wel gelden dat de verschillende stations in evenwicht zijn. Wanneer dit niet het geval is komen er namelijk meer patiënten aan dan het systeem aan kan. De eerste 26 weken moet het systeem nog in evenwicht komen, dus deze weken worden niet meegenomen. Er worden 41 replicaties van een simulatie van 126 weken gemaakt. De simulatie is gedaan in PYTHON en is te zien in Appendix H.

4.5.3 Dynamisch patiënten oproepen

Om de maximale aankomstintensiteit te bepalen waarvoor de dagbehandeling in evenwicht

is wordt (8) gebruikt. Maar in dit geval is λ_3 de onbekende in plaats van s_3 . Dit wordt namelijk gedaan voor een vaste capaciteit.

4.6. BEHANDELING MAXIMAAL EEN HALF JAAR

Er is een mogelijkheid dat het behandelingsproces in plaats van maximaal een jaar, maximaal een half jaar wordt. In deze paragraaf wordt deze mogelijkheid bekeken. Eerst wordt er in lid 4.6.1 gekeken naar het analytische model en vervolgens in lid 4.6.2 naar de simulatie van deze mogelijkheid.

4.6.1 Analytisch model

Wanneer de behandeling nog maar maximaal een half jaar is, is het model hetzelfde als in paragraaf 4.5. Alleen zijn er dan natuurlijk minder verschillende type patiënten, omdat de patiënt op minder momenten het systeem kan verlaten. De type patiënten die in dit model zijn, zijn ook in het model van het huidige behandelingsproces. De verschillende type patiënten zijn namelijk type 1 tot en met 6 zoals beschreven in paragraaf 4.5. Dit zijn namelijk de enige type die het behandelingsproces binnen een half jaar afronden. De aankomstintensiteiten van type 1 tot en met 5 blijven hetzelfde, maar voor type 6 patiënten geldt dat de kans hierop niet gelijk is aan die van het huidige behandelingsproces, want de types 7 tot en met 11 van het huidige behandelingsproces behoren nu tot type 6. Dus de kans dat een patiënt een patiënt van type 6 is, is gelijk aan 0.3277. Hieruit volgt dat de aankomstintensiteit van type 6 patiënten gemiddeld $0.3277 \cdot 5 = 1.6384$ per week is.

Nu kunnen de aankomstintensiteiten per station bepaald worden, hiervoor is Tabel 1 gebruikt.

$$\begin{aligned}\lambda_1 &= 2\nu(1) + 3\nu(2) + 4\nu(3) + 5\nu(4) \\ &\quad + \nu(5) + 7\nu(6) \\ &= 23.4\end{aligned}$$

$$\begin{aligned}\lambda_2 &= 2\nu(1) + 4\nu(2) + 6\nu(3) + 8\nu(4) \\ &\quad + 10\nu(5) + 12\nu(6) \\ &= 36.9\end{aligned}$$

$$\begin{aligned}\lambda_3 &= 2\nu(1) + 4\nu(2) + 6\nu(3) + 8\nu(4) \\ &\quad + 10\nu(5) + 12\nu(6) \\ &= 36.9\end{aligned}$$

$$\begin{aligned}\lambda_4 &= \nu(1) + 2\nu(2) + 3\nu(3) + 4\nu(4) \\ &\quad + 5\nu(5) + 6\nu(6) \\ &= 18.4\end{aligned}$$

$$\begin{aligned}\lambda_5 &= \nu(1) + \nu(2) + 2\nu(3) + 2\nu(4) \\ &\quad + 3\nu(5) + 3\nu(6) \\ &= 10.2\end{aligned}$$

Nu kan weer bepaald worden wanneer het systeem stationair is, dus wanneer de verschillende stations stationair zijn. In (16), (17), (18), (19) en (20) is respectievelijk voor de longarts, het priklab, de dagbehandeling, de verpleegkundig specialist en de CT-scan te zien wanneer deze in evenwicht zijn.

$$\rho_1 = \frac{\lambda_1}{s_1 \cdot \mu_1} = \frac{23.4 \cdot 0.008}{s_1} < 1 \quad (16)$$

$$\rho_2 = \frac{\lambda_2}{s_2 \cdot \mu_2} = \frac{36.9 \cdot 0.007}{s_2} < 1 \quad (17)$$

$$\rho_3 = \frac{\lambda_3}{s_3 \cdot \mu_3} = \frac{36.9 \cdot 0.019}{s_3} < 1 \quad (18)$$

$$\rho_4 = \frac{\lambda_4}{s_4 \cdot \mu_4} = \frac{18.4 \cdot 0.008}{s_4} < 1 \quad (19)$$

$$\rho_5 = \frac{\lambda_5}{s_5 \cdot \mu_5} = \frac{10.2 \cdot 0.01}{s_5} < 1 \quad (20)$$

Uit (16), (17), (18), (19) en (20) kan afgeleid worden hoeveel servers er nodig zijn voor een evenwicht bij respectievelijk de longarts, het priklab, de dagbehandeling, de verpleegkundig specialist en de CT-scan.

Dit is te zien in (21), (22), (23), (24) en (25).

$$s_1 > 23.4 \cdot 0.008 = 0.20 \quad (21)$$

$$s_2 > 36.9 \cdot 0.007 = 0.26 \quad (22)$$

$$s_3 > 36.9 \cdot 0.019 = 0.69 \quad (23)$$

$$s_4 > 18.4 \cdot 0.008 = 0.15 \quad (24)$$

$$s_5 > 10.2 \cdot 0.01 = 0.11 \quad (25)$$

Hieruit volgt dus dat er een longarts minimaal 5.9 uur, het priklab 12.3 uur, de dagbehandeling 27.7 uur, de verpleegkundig specialist 4.6 uur en de CT-scan 5.1 uur een server beschikbaar moeten hebben.

4.6.2 Simulatie

Voor de simulatie kunnen de type patiënten en kansen zoals beschreven in 4.2 gebruikt worden. Hiervoor moet natuurlijk ook wel gelden dat de verschillende stations in evenwicht zijn. Wanneer dit niet het geval is komen er namelijk meer patiënten aan dan het systeem aan kan. Uit (21), (22), (23), (24) en (25) volgt dat het systeem in evenwicht is, op het moment dat er voor ieder station één server beschikbaar is. Dit wordt gebruikt in de simulatie.

5. RESULTATEN

In dit hoofdstuk worden de resultaten van het in hoofdstuk 4 beschreven analytische model en de simulatie gegeven. Dit wordt eerst gedaan voor het huidige behandelingsproces en daarna voor het behandelingsproces van maximaal een jaar. .

5.1. HUIDIGE BEHANDELINGSPROCES

In deze paragraaf staan de resultaten van het huidige behandelingsproces. In lid 5.1.1 staan de resultaten van de benodigde capaciteit. Vervolgens staat er in lid 5.1.2 het maximale aantal nieuwe patiënten die per week opgeroepen kunnen worden bij een vaste capaciteit, zodat het systeem in evenwicht is. Vervolgens worden deze resultaten gebruikt voor de simulatie. De resultaten hiervan staan in lid 5.1.3. Tenslotte staan in lid 5.1.4 de resultaten van dynamisch patiënten oproepen.

5.1.1 Benodigde capaciteit

In Figuur 16, 17, 18, 19 en 20 in Appendix D.1 is de gemiddelde verblijftijd tegen het aantal servers te zien bij respectievelijk de longarts, prik-lab, dagbehandeling, verpleegkundig specialist en de CT-scan. Het aantal servers is hierbij gevarieerd tussen het minimale aantal servers zodat het station in evenwicht is en het aantal servers dat het Antoni van Leeuwenhoekziekenhuis heeft. Deze figuren zijn gemaakt met behulp van MATLAB en Appendix F.

Uit Figuur 16, 17, 18, 19 en 20 in Appendix D.1. blijkt dat de gemiddelde verblijftijd van de verschillende stations vooral veel toeneemt ten opzichte van het maximale aantal servers, wanneer het aantal servers minder is dan één. Vervolgens wordt er gekeken naar het verschil in de gemiddelde verblijftijd tussen het geval van één server per station en het maximale aantal servers per station. Wanneer er één server per station de gehele werkweek beschikbaar is voor de behandeling, is de gemiddelde verblijftijd per station 19.8 minuten, 30.1 minuten, 366.2 minuten, 18.7 minuten en 34.4 minuten voor respectievelijk de longarts, het prik-lab, de dagbehandeling, de verpleegkundig specialist en de CT-scan. Wanneer nu het maximale aantal servers genomen wordt zijn deze verblijftijden respectievelijk 15.5 minuten, 21.8 minuten, 45.8 minuten, 16.1 minuten en 32.1 minuten. De verschillen tussen de verblijftijden met één server en het maximale aantal servers voor de longarts, het prik-lab, de dagbehandeling, de verpleegkundig specialist en de CT-scan zijn respectievelijk 4.3 minuten, 8.3 minuten, 320.3 minuten, 2.7 minuten en 2.4 minuten. Hieruit volgt dus dat bij de longarts, het prik-lab, de verpleegkundig specialist en de CT-scan de gemiddelde verblijftijd niet veel afneemt wanneer er meer servers bij het station komen. Bij de dagbehandeling wordt de gemiddelde verblijftijd wel aanzienlijk verminderd wordt. Toch wordt er in de simulatie maar één server genomen. Wanneer blijkt dat de meeste patiënten de behandelingsweek niet binnen één dag afronden, kan er gekozen worden om servers bij de dagbehandeling toe te voegen.

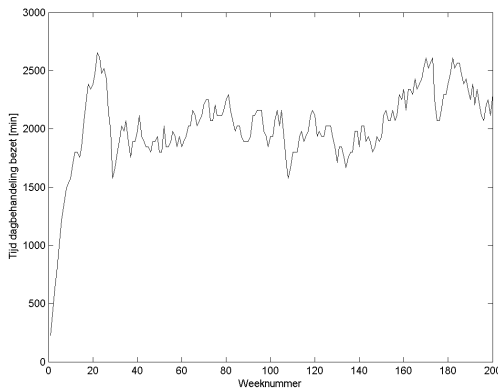
5.1.2 Gevoeligheidsanalyse

Voor de gevoeligheidsanalyse wordt er voor ieder station één server genomen. Met behulp van MATLAB en Appendix G wordt er bepaald wat de maximale aankomstintensiteit van nieuwe patiënten is. Hieruit volgt dat de aankomstintensiteit maximaal 20 is voor de longarts, maximaal 15 is voor het prik-lab, maximaal 5 is voor de dagbehandeling, maximaal 25 is voor de verpleegkundig specialist en maximaal 38 is voor de CT-scan. Dus wanneer er voor ieder station één server de gehele week beschikbaar is kunnen er maximaal vijf nieuwe patiënten per week opgeroepen worden, want dan zijn alle stations stationair. Hieruit volgt dus dat er niet meer nieuwe patiënten opgeroepen kunnen worden dan de strategie van het Antoni van Leeuwenhoekziekenhuis. Wanneer er nu meer capaciteit is, kunnen er misschien ook wel meer patiënten opgeroepen worden per week. Dit hangt er natuurlijk van af bij wel station er meer capaciteit is. Hoeveel patiënten er dan opgeroepen kunnen worden is te zien in Tabel 13 in Appendix C.1

5.1.3 Verfijnd model

Voor de simulatie is voor ieder station één server genomen, zoals beschreven is in lid 5.1.1 en er worden vijf nieuwe patiënten per week opgeroepen zoals beschreven is in lid 5.1.2. In Tabel 2 staan de resultaten per station van de simulatie. In Tabel 3 staan de resultaten per behandelingsweek, dus de gemiddelde verblijftijden en het gemiddeld aantal dagen dat hierover gedaan wordt. Het prik-lab en de dagbehandeling sluiten eigenlijk op 2850 minuten. Uit Tabel 2 volgt dus dat het 95% betrouwbaarheidsinterval voor het gemiddelde aantal minuten overwerken van de dagbehandeling op zaterdag [18 27] is. Verder volgt uit het 95% betrouwbaarheidsinterval van de sluiting van het prik-lab, dat in 95% van de weken het prik-lab gemiddeld niet overwerkt op zaterdag. Uit Tabel 2 volgt dat de dagbehandeling wel minder dan 2850 minuten bezet is.

Verder blijkt uit Tabel 2 dat het maximum van het 95% betrouwbaarheidsinterval van de gemiddelde sluiting nog op vrijdag is, dus dan hoeft de CT scan niet op zaterdag open te zijn. Uit Tabel 3 volgt dat de meeste behandelingsweken van de patiënten één dag duurt. Dit is het geval, omdat de hoogste bovengrens van de 95% betrouwbaarheidsintervallen van het gemiddeld aantal dagen dat een patiënt over een behandelingsweek doet 1.13 is. Dit is bij een behandelingsweek met een intake bij de longarts. Hieruit kan geconcludeerd worden dat één server bij de dagbehandeling voldoende is. Aangezien het knelpunt de dagbehandeling is, is er voor de dagbehandeling een grafiek gemaakt van het aantal minuten dat de dagbehandeling in een week bezet is tegen het weeknummer. Hiervoor is een simulatie van 200 weken gemaakt, waarbij er begonnen wordt in een leeg systeem. Dit is te zien in Figuur 2. In de grafiek is goed te zien dat er met een leeg systeem begonnen wordt. Hierdoor is er namelijk tussen de 0 en 20 weken een stijging van de bezetting van de dagbehandeling te zien. De maximale tijd dat de dagbehandeling bezet is, is 2655 minuten per week. Dit is dus meer dan 40 uur per week, terwijl dit eigenlijk de werkweek van de dagbehandeling is.



Figuur 2: De tijd dat de dagbehandeling bezet is tegen het weeknummer.

Wanneer er naar het maximum gekeken wordt, waarbij de opwarm periode niet meegenomen wordt is deze 2610 minuten. Dus dit is ook meer dan 40 uur.

5.1.4 Dynamisch patiënten oproepen

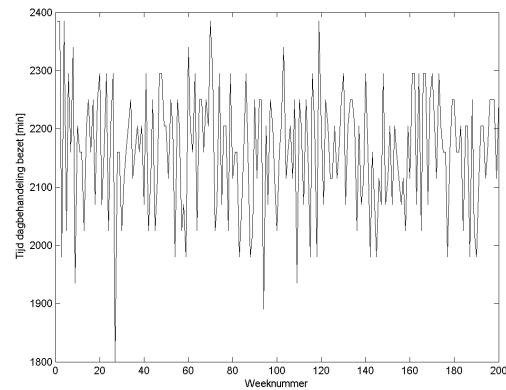
De maximale aankomstintensiteit van de dagbehandeling bepaald worden, aangezien nu de capaciteit vast staat zoals te zien is in lid 5.1.3. Dit is te zien in (26).

$$\rho_3 = \frac{\lambda_3}{s_3 \cdot \mu_3} = \frac{\lambda_3 \cdot 0.019}{1} < 1 \quad (26)$$

Met behulp van (26) kan λ_3 bepaald worden, dit is te zien in (27).

$$\lambda_3 < \frac{1}{0.0019} = 53.3 \quad (27)$$

Uit (27) volgt dat de dagbehandeling in evenwicht is op het moment dat er 53 patiënten aankomen. Er worden dus 53 minus het aantal patiënten dat de komende behandelingsweek terug komen opgeroepen. Het aantal patiënten die al een jaar in het systeem zitten wordt hierbij opgeteld. De resultaten hiervan zijn te zien in Tabel 4 en Tabel 5. Verder is het 95% betrouwbaarheidsinterval voor het gemiddeld aantal nieuwe patiënten [5.0 5.1].



Figuur 3: De tijd dat de dagbehandeling bezet is tegen het weeknummer.

Uit Tabel 4 en Tabel 5 volgt dat er niet veel meer patiënten opgeroepen worden, maar ongeveer hetzelfde. De betrouwbaarheidsintervallen in Tabel 4 en Tabel 5 zijn wel kleiner geworden, hieruit volgt dat de variantie kleiner is geworden door het dynamisch patiënten oproepen. Dit is ook te zien in Figuur 3. Het maximum is 2385 minuten per week bezet.

Tabel 2: Een tabel met de resultaten van de simulatie per station, met een vaste aankomstintensiteit van nieuwe patiënten en het huidige behandelingsproces

Station	Prestatiemaat	95% betrouwbaarheidsinterval
Longarts	Bezetting [min]	[579 583]
Longarts	aankomstintensiteit [per week]	[28.6 29.2]
Priklab	Bezetting [min]	[945 969]
Priklab	aankomstintensiteit [per week]	[47.3 48.5]
Priklab	sluiting [min]	[2761 2763]
Dagbehandeling	Bezetting [min]	[2127 2180]
Dagbehandeling	aankomstintensiteit [per week]	[47.3 48.5]
Dagbehandeling	sluiting [min]	[2868 2877]
Verpleegkundig specialist	Bezetting [min]	[354 364]
Verpleegkundig specialist	aankomstintensiteit [per week]	[23.6 24.2]
CT-scan	Bezetting [min]	[368 375]
CT-scan	aankomstintensiteit [per week]	[12.3 12.5]
CT-scan	sluiting [min]	[2677 2692]

Tabel 3: Een tabel met de resultaten van de simulatie per behandelingsweek, met een vaste aankomstintensiteit van nieuwe patiënten en het huidige behandelingsproces

Behandelingsweek	Prestatiemaat	95% betrouwbaarheidsinterval
Alleen longarts	Verblijftijd [min]	[15.0 15.0]
Alleen longarts	Aantal dagen	[1.00 1.00]
Met controle longarts	Verblijftijd [min]	[126.2 130.5]
Met controle longarts	Aantal dagen	[1.02 1.02]
Met intake longarts	Verblijftijd [min]	[149.8 153.4]
Met intake longarts	Aantal dagen	[1.13 1.13]
Met controle verpleegkundig specialist	Verblijftijd [min]	[123.9 128.3]
Met controle verpleegkundig specialist	Aantal dagen	[1.03 1.03]
Met CT-scan	Verblijftijd [min]	[151.0 155.0]
Met CT-scan	Aantal dagen	[1.04 1.04]

Ook wanneer de opwarm periode niet meege-
nomen wordt, terwijl dit bij de strategie van
het Antoni van Leeuwenhoekziekenhuis 2610
minuten is. Dus bij deze strategie is de dag-
behandeling in geen enkele week meer dan 40
uur nodig.

5.2. BEHANDELING MAXIMAAL EEN HALF JAAR

In deze paragraaf staan de resultaten van het
behandelingsproces van maximaal een half jaar.
Hier worden dezelfde stappen genomen als
bij het huidige behandelingsproces. In lid
5.2.1 staan namelijk de resultaten van de beno-
digde capaciteit. Vervolgens staat er in lid 5.2.2

het maximale aantal nieuwe patiënten die per
week opgeroepen kunnen worden bij een vaste
capaciteit, zodat het systeem in evenwicht is.
Vervolgens worden deze resultaten gebruikt
voor de simulatie. De resultaten hiervan staan
in lid 5.2.3. Tenslotte staan in lid 5.2.4 de resul-
taten van dynamisch patiënten oproepen.

5.2.1 Benodigde capaciteit

In Figuur 21, 22, 23, 24 en 25 in Appendix D.2
is de gemiddelde verblijftijd tegen het aantal
servers te zien bij respectievelijk de longarts,
priklab, dagbehandeling, verpleegkundig spe-
cialist en de CT-scan.

Tabel 4: Een tabel met de resultaten van de simulatie per station wanneer patiënten dynamisch opgeroepen worden en het huidige behandelingsproces bekeken wordt.

Station	Prestatiemaat	95% betrouwbaarheidsinterval
Longarts	Bezetting [min]	[591 595]
Longarts	aankomstintensiteit [per week]	[29.3 29.4]
Priklab	Bezetting [min]	[969 971]
Priklab	aankomstintensiteit [per week]	[48.4 48.6]
Priklab	sluiting [min]	[2763 2764]
Dagbehandeling	Bezetting [min]	[2180 2185]
Dagbehandeling	aankomstintensiteit [per week]	[48.4 48.6]
Dagbehandeling	sluiting [min]	[2876 2877]
Verpleegkundig specialist	Bezetting [min]	[363 364]
Verpleegkundig specialist	aankomstintensiteit [per week]	[24.2 24.3]
CT-scan	Bezetting [min]	[375 377]
CT-scan	aankomstintensiteit [per week]	[12.5 12.6]
CT-scan	sluiting [min]	[2679 2693]

Tabel 5: Een tabel met de resultaten van de simulatie per behandelingsweek wanneer patiënten dynamisch opgeroepen worden en het huidige behandelingsproces bekeken wordt.

Behandelingsweek	Prestatiemaat	95% betrouwbaarheidsinterval
Alleen longarts	Verblijftijd [min]	[15.0 15.1]
Alleen longarts	Aantal dagen	[1.00 1.00]
Met controle longarts	Verblijftijd [min]	[126.8 127.4]
Met controle longarts	Aantal dagen	[1.01 1.01]
Met intake longarts	Verblijftijd [min]	[158.8 159.6]
Met intake longarts	Aantal dagen	[1.18 1.19]
Met controle verpleegkundig specialist	Verblijftijd [min]	[126.0 126.6]
Met controle verpleegkundig specialist	Aantal dagen	[1.03 1.03]
Met CT-scan	Verblijftijd [min]	[152.7 153.5]
Met CT-scan	Aantal dagen	[1.05 1.05]

Het aantal servers is hierbij gevarieerd tussen het minimale aantal servers zodat het station in evenwicht is en het aantal servers dat het Antoni van Leeuwenhoekziekenhuis heeft. Wanneer er één server per station de gehele werkweek beschikbaar is voor de behandeling, is de gemiddelde verblijftijd per station respectievelijk 18.6 minuten, 27.2 minuten, 146.0 minuten, 17.7 minuten en 34.0 minuten. In deze grafieken is te zien dat de gemiddelde verblijftijd bij ieder station niet veel meer afneemt op het moment dat er meer dan één server per station is. Dus ook in dit geval wordt er voor ieder station één server genomen. Dit was ook te

verwachten, omdat dit ook gold bij het huidige behandelingsproces en nu zijn er minder patiënten in een behandelingsweek, omdat de patiënten minder lang in het systeem kunnen blijven.

5.2.2 Gevoeligheidsanalyse

Voor de gevoeligheidsanalyse wordt er voor ieder station één server genomen, zoals besproken is in lid 5.2.1. Met behulp van MATLAB en Appendix G wordt er bepaald wat de maximale aankomstintensiteit van nieuwe patiënten is.

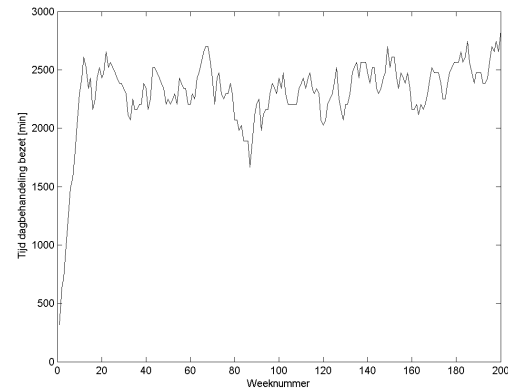
Hieruit volgt dat de aankomstintensiteit maximaal 25 is voor de longarts, maximaal 18 is voor het priklab, maximaal 7 is voor de dagbehandeling, maximaal 32 is voor de verpleegkundig specialist en maximaal 46 is voor de CT-scan. Wanneer er voor ieder station één server de gehele week beschikbaar is kunnen er dus maximaal zeven nieuwe patiënten per week opgeroepen worden, want dan zijn alle stations stationair. Wanneer er nu meer capaciteit is, kunnen er misschien ook wel meer patiënten opgeroepen worden per week. Dit hangt er natuurlijk van af bij wel station er meer capaciteit is. Hoeveel patiënten er dan opgeroepen kunnen worden is te zien in Tabel 14 in Appendix C.2

5.2.3 Verfijnd model

Voor de simulatie is voor ieder station één server genomen, zoals beschreven is in lid 5.2.1 en er worden vijf nieuwe patiënten per week opgeroepen zoals beschreven is in lid 5.2.2. In Tabel 6 staan de resultaten per station van de simulatie. In Tabel 7 staan de resultaten per behandelingsweek, dus de gemiddelde verblijftijden en het gemiddeld aantal dagen dat hierover gedaan wordt. Het priklab en de dagbehandeling sluiten eigenlijk op 2850 minuten. Uit Tabel 6 volgt dat het 95% betrouwbaarheidsinterval voor het gemiddelde aantal minuten overwerken van de dagbehandeling op zaterdag [46 51] is. Verder volgt uit het 95% betrouwbaarheidsinterval van de sluiting van het priklab, dat met 95% zekerheid het priklab gemiddeld niet overwerkt op zaterdag. Verder blijkt uit Tabel 6 dat het maximum van het 95% betrouwbaarheidsinterval nog op vrijdag is, dus dan hoeft de CT scan met 95% zekerheid gemiddeld niet op zaterdag open te zijn.

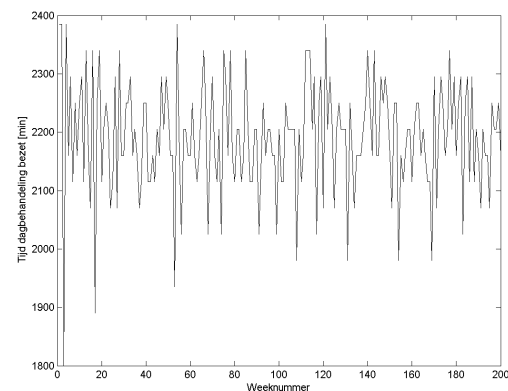
Uit Tabel 7 volgt dat de meeste behandelingsweken van de patiënten één dag duurt. Dit is het geval, omdat de hoogste bovengrens van de 95% betrouwbaarheidsintervallen 1.15 is. Dit is bij een behandelingsweek met een intake bij de longarts. Hieruit kan geconcludeerd worden dat één server bij de dagbehandeling voldoende is. In Figuur 4 is de bezetting van de

dagbehandeling per week te zien. Hieruit blijkt dat de dagbehandeling maximaal 2835 minuten per week open is. Dit maximum valt buiten de opwarm periode. Dit is dus meer dan de werkweek van 40 uur.



Figuur 4: De tijd dat de dagbehandeling bezet is tegen het weeknummer.

5.2.4 Dynamisch patiënten oproepen



Figuur 5: De tijd dat de dagbehandeling bezet is tegen het weeknummer.

Wanneer er dynamisch patiënten opgeroepen worden zoals beschreven in lid 5.1.4, is het 95% betrouwbaarheidsinterval voor het gemiddeld aantal nieuwe aankomsten per week is [6.2 6.3]. Hieruit blijkt dus dat er bij deze strategie minder patiënten behandeld worden dan bij de strategie zeven nieuwe patiënten per week. Dit is bij het dynamisch patiënten

Tabel 6: Een tabel met de resultaten van de simulatie per station, met een vaste aankomstintensiteit van nieuwe patiënten en het behandelingsproces van maximaal een half jaar

Station	Prestatiemaat	95% betrouwbaarheidsinterval
Longarts	Bezetting [min]	[702 706]
Longarts	aankomstintensiteit [per week]	[32.8 33.1]
Priklab	Bezetting [min]	[1032 1043]
Priklab	aankomstintensiteit [per week]	[51.6 52.2]
Priklab	sluiting [min]	[2770 2770]
Dagbehandeling	Bezetting [min]	[2322 2347]
Dagbehandeling	aankomstintensiteit [per week]	[51.6 52.2]
Dagbehandeling	sluiting [min]	[2896 2901]
Verpleegkundig specialist	Bezetting [min]	[387 391]
Verpleegkundig specialist	aankomstintensiteit [per week]	[25.8 26.1]
CT-scan	Bezetting [min]	[430 434]
CT-scan	aankomstintensiteit [per week]	[14.3 14.5]
CT-scan	sluiting [min]	[2877 2882]

Tabel 7: Een tabel met de resultaten van de simulatie per behandelingsweek, met een vaste aankomstintensiteit van nieuwe patiënten en het behandelingsproces van maximaal een half jaar

Behandelingsweek	Prestatiemaat	95% betrouwbaarheidsinterval
Alleen longarts	Verblijftijd [min]	[15.0 15.1]
Alleen longarts	Aantal dagen	[1.00 1.00]
Met controle longarts	Verblijftijd [min]	[142.7 145.1]
Met controle longarts	Aantal dagen	[1.05 1.05]
Met intake longarts	Verblijftijd [min]	[169.0 171.4]
Met intake longarts	Aantal dagen	[1.15 1.15]
Met controle verpleegkundig specialist	Verblijftijd [min]	[137.9 140.2]
Met controle verpleegkundig specialist	Aantal dagen	[1.04 1.04]
Met CT-scan	Verblijftijd [min]	[167.7 170.0]
Met CT-scan	Aantal dagen	[1.11 1.11]

oproepen niet het geval. Dit is te verklaren, doordat $\rho = 0.9684$ wanneer er zeven patiënten per week opgeroepen worden. Bij het huidige behandelingsproces geldt $\rho = 0.8771$ bij het oproepen van vijf nieuwe patiënten per week. In Tabel 8 staan de resultaten per station van de simulatie. Hierbij is voor ieder station één server genomen, zoals beschreven is in lid 5.2.1. In Tabel 9 staan de resultaten per behandelingsweek, dus de gemiddelde verblijftijden en het gemiddeld aantal dagen dat hierover gedaan wordt. Het is te zien dat de waarden wat lager geworden zijn. Dit is logisch omdat er gemiddeld ook minder nieuwe patiënten op-

geroepen worden. Verder is er ook te zien dat de breedte van de betrouwbaarheidsintervallen kleiner geworden is, dit komt omdat het systeem stabiel is. Verder is er nog een figuur gemaakt voor het gemiddeld aantal minuten dat de dagbehandeling bezet is. Dit is te zien in Figuur 5. Het maximale aantal minuten dat de dagbehandeling bezet is, is 2385 minuten. Terwijl dit bij het oproepen van zeven patiënten per week 2835 minuten is.

5.3. VERIFICATIE

In deze paragraaf worden het analytische model en de simulatie geverifieerd.

Tabel 8: Een tabel met de resultaten van de simulatie per station wanneer patiënten dynamisch opgeroepen worden.

Station	Prestatiemaat	95% betrouwbaarheidsinterval
Longarts	Bezetting [min]	[659 661]
Longarts	aankomstintensiteit [per week]	[30.8 30.9]
Priklab	Bezetting [min]	[970 973]
Priklab	aankomstintensiteit [per week]	[48.5 48.6]
Priklab	sluiting [min]	[2766 2766]
Dagbehandeling	Bezetting [min]	[2184 2189]
Dagbehandeling	aankomstintensiteit [per week]	[48.5 48.6]
Dagbehandeling	sluiting [min]	[2868 2870]
Verpleegkundig specialist	Bezetting [min]	[364 365]
Verpleegkundig specialist	aankomstintensiteit [per week]	[24.2 24.3]
CT-scan	Bezetting [min]	[404 406]
CT-scan	aankomstintensiteit [per week]	[13.5 13.5]
CT-scan	sluiting [min]	[2843 2846]

Tabel 9: Een tabel met de resultaten van de simulatie per behandelingsweek wanneer patiënten dynamisch opgeroepen worden.

Behandelingsweek	Prestatiemaat	95% betrouwbaarheidsinterval
Alleen longarts	Verblijftijd [min]	[15.1 15.2]
Alleen longarts	Aantal dagen	[1.00 1.00]
Met controle longarts	Verblijftijd [min]	[128.0 128.7]
Met controle longarts	Aantal dagen	[1.02 1.02]
Met intake longarts	Verblijftijd [min]	[160.1 160.8]
Met intake longarts	Aantal dagen	[1.16 1.16]
Met controle verpleegkundig specialist	Verblijftijd [min]	[126.7 127.6]
Met controle verpleegkundig specialist	Aantal dagen	[1.02 1.02]
Met CT-scan	Verblijftijd [min]	[156.1 157.0]
Met CT-scan	Aantal dagen	[1.08 1.08]

In lid 5.3.1 wordt eerst het analytische model geverifieerd. Vervolgens wordt in lid 5.3.2 de simulatie geverifieerd door dit te vergelijken met het analytische model.

5.3.1 Analytisch model

Bij het analytisch model is de gemiddelde aankomstintensiteit bij de verpleegkundig specialist 23.9 patiënten per week. Door de samenwerking met het Antoni van Leeuwenhoek ziekenhuis blijkt, dat er ongeveer 25 patiënten per week bij de verpleegkundig specialist komen. Hieruit kan geconcludeerd worden dat de aanname dat de kans geometrisch verdeeld is met

het gegeven dat er na drie maanden nog 65% van de patiënten in het systeem zijn een goede benadering van de werkelijkheid is.

5.3.2 Simulatie

Voor de verificatie wordt de simulatie van het huidige behandelingsproces genomen. Verder wordt er een werkweek genomen zodat ieder station stationair is. Dus voor de verificatie is een werkweek van 40 uur genomen. Ook is ervoor gekozen om na 40 uur geen patiënten meer aan te laten komen bij de longarts en de verpleegkundig specialist.

De andere stations zijn open totdat alle patiënten de behandelingsweek voltooid hebben. Voor de simulatie wordt er geen rekening gehouden met de openingstijden van de verschillende stations, zodat het beter te vergelijken is met het analytische model. Er wordt een simulatie van 126 weken gemaakt, waarbij de eerst 26 weken niet meegenomen worden. Hierdoor is er een opwarm periode van 26 weken genomen. Er zijn 41 replicaties gemaakt van deze simulatie. Hierdoor zijn er 40 vrijheidsgraden voor de betrouwbaarheidsintervallen. Verder zijn de bedieningsduren exponentieel verdeeld en het aankomstproces is Poisson. Ook is er in plaats van een intake een controle. Hierdoor kunnen de simulatie en het analytische model vergeleken worden. De resultaten zijn te zien in Tabel 10. Zoals hier te zien is liggen alle analytisch berekende aankomstintensiteiten in de betrouwbaarheidsintervallen van de simulatie. De verblijftijd ligt alleen bij de dagbehandeling in het betrouwbaarheidsinterval. Bij de andere stations wijkt deze minder dan vijf minuten van het betrouwbaarheidsinterval af. Dit is gedeeltelijk te verklaren doordat de stations niet precies 40 uur geopend zijn, omdat er nog een uitloop meegenomen wordt. Wanneer vervolgens de analytische aankomstintensiteiten vergeleken worden met de aankomstintensiteiten van het huidige behandelingsproces met een vaste aankomstintensiteit zoals te zien is in Tabel 2, is te zien dat de analytische aankomstintensiteiten van de longarts, verpleegkundig specialist en de CT scan in de 95% betrouwbaarheidsintervallen van Tabel 2 liggen. De analytische aankomstintensiteiten van het prik-lab en de dagbehandeling liggen niet in de betrouwbaarheidsintervallen van Tabel 2. De analytische waardes zijn net wat lager. Hieruit kan geconcludeerd worden dat de simulatie correct werkt.

6. DISCUSSIE

Wanneer dit onderzoek uitgebreid wordt, zou er gekeken kunnen worden naar de sluiting van de verschillende stations. In dit onderzoek wordt de sluiting van een station gezien als een

tijd waarin er geen patiënten aankomen op dit station. Maar in werkelijkheid wordt het station gesloten en is de server niet meer beschikbaar. Ook zou er gekeken kunnen worden naar een andere verdeling voor de bedieningsduur van de patiënten, bijvoorbeeld gebaseerd op data van de bedieningsduren van verschillende patiënten. Er zou gedacht kunnen worden over een andere strategie voor het oproepen en inplannen van patiënten. Verder zou er beter gekeken kunnen worden naar de benodigde capaciteit van de longarts, in dit onderzoek zijn er namelijk onnodige consulten meegenomen. Gedurende dit onderzoek is het behandelingsproces wat veranderd. Er wordt namelijk vaker een CT-scan gemaakt, dan in het model van dit onderzoek. Eerst werden er vijf CT-scans per jaar gemaakt, maar nu wordt er iedere zes weken een CT scan gemaakt. Hiervoor zou het model aangepast kunnen worden. Tenslotte zou er onderzoek gedaan kunnen worden naar de toepasbaarheid van dit model op andere ziekenhuizen die de behandeling van Nivolumab aanbieden.

7. CONCLUSIE

Wanneer het Antoni van Leeuwenhoekziekenhuis niet meer dan één bed beschikbaar heeft voor de behandeling van Nivolumab, is de huidige strategie bij het huidige behandelingsproces de strategie die het maximaal aantal patiënten oproept bij een vaste aankomstintensiteit van nieuwe patiënten. Er zou ook gekozen kunnen worden om patiënten dynamisch op te roepen. Hiermee wordt bedoeld dat het aantal nieuwe patiënten de maximale aankomstintensiteit is bij de dagbehandeling minus het aantal patiënten dat nog in het behandelingsproces is. Hierbij wordt het aantal patiënten dat al de maximale duur van het behandelingsproces bereikt hebben opgeteld. In dit geval is het betrouwbaarheidsinterval van de gemiddelde aankomstintensiteit [5.0 5.1]. Dit zijn dus ongeveer evenveel nieuwe patiënten. Het voordeel hiervan is wel, dat de maximale bezetting van de dagbehandeling in een week 2385 minuten is. Terwijl dit bij een vaste aankomstintensiteit

Tabel 10: Een tabel met prestatie-maten voor de verificatie van de simulatie.

Station	Prestatiemaat	Analytisch	Simulatie
Longarts	Verblijftijd [min]	18.3	[19.1 20.1]
Longarts	Aankomstintensiteit [per week]	28.9	[27.8 30.3]
Priklab	Verblijftijd [min]	32.8	[37.5 43.7]
Priklab	Werkweek [uur]	2400	[2394 2403]
Priklab	Aankomstintensiteit [per week]	46.8	[45.7 50.2]
Dagbehandeling	Verblijftijd [min]	366.2	[354.4 376.2]
Dagbehandeling	Werkweek [min]	2400	[2766 2884]
Dagbehandeling	Aankomstintensiteit [per week]	46.8	[45.7 50.2]
Verpleegkundig specialist	Verblijftijd [min]	17.6	[18.4 19.1]
Verpleegkundig specialist	Aankomstintensiteit [per week]	23.9	[22.8 25.2]
CT-scan	Verblijftijd [min]	35.5	[36.4 37.4]
CT-scan	Werkweek [min]	2400	[2459 2581]
CT-scan	Aankomstintensiteit [per week]	12.4	[11.9 13.0]

van nieuwe patiënten 2610 is. Dus wanneer patiënten dynamisch opgeroepen worden is één bed bij de dagbehandeling altijd voldoende, terwijl het bij een vaste aankomstintensiteit voorkomt dat er in een week meer capaciteit nodig is. Wanneer er meer capaciteit is, kunnen er misschien ook meer nieuwe patiënten opgeroepen worden. De maximale gemiddelde aankomstintensiteit van nieuwe patiënten per week is weergegeven in Tabel 13 in Appendix C.1.

Wanneer nu de het huidige behandelingsproces veranderd in een behandelingsproces dat maximaal een half jaar duurt, is de maximale vaste aankomstintensiteit van nieuwe patiënten zeven. Wanneer patiënten dynamisch opgeroepen worden, is het betrouwbaarheidsinterval van de gemiddelde aankomstintensiteit [6.2 6.3]. In dit geval worden er dus minder patiënten opgeroepen wanneer dit dynamisch gebeurt. Het voordeel hiervan is wel, dat net zoals bij het huidige behandelingsproces één bed bij de dagbehandeling altijd voldoende capaciteit is, terwijl dat bij een vaste aankomstintensiteit van nieuwe patiënten niet het geval is. De maximale bezetting van de dagbehandeling in een week is 2385 minuten wanneer patiënten dynamisch opgeroepen worden. Terwijl dit bij een vaste aankomstintensiteit van nieuwe patiënten 2835 is. Wanneer er meer capaci-

teit is, kunnen er misschien ook meer nieuwe patiënten opgeroepen worden. De maximale gemiddelde aankomstintensiteit van nieuwe patiënten per week is weergegeven in Tabel 14 in Appendix C.2.

Wanneer er gekozen wordt voor de zekerheid dat de capaciteit altijd voldoende is, zou er gekozen moeten worden voor het dynamisch patiënten oproepen. Wanneer het te veel werk is om patiënten dynamisch op te roepen, dus om bij te houden hoeveel patiënten er opgeroepen moet worden, zou er gekozen moeten worden voor een vaste aankomstintensiteit van nieuwe patiënten

8. DANKWOORD

Ik wil graag mijn begeleider Richard J. Boucherie bedanken voor de hulp gedurende mijn bachelor opdracht. Verder wil ik Paul Baas, Marion Zimmerman en Mirte Muller van het Antoni van Leeuwenhoek ziekenhuis bedanken voor de medewerking aan mijn bachelor opdracht. Ook wil ik Jasmijn Manders bedanken voor het meedenken, wanneer ik vast zat.

REFERENTIES

- [1] Lakshmi C en Sivakumar Appa Iyer. Application of queueing theory in health care:

- A literature review. *Operations Research for Health Care*, maart 2013.
- [2] F.P. Kelly. *Reversibility and stochastic networks*. Cambridge University Press, 1979.
 - [3] James F. Giglio en Robert A. Green Linda V. Green, João Soares. Using queueing theory to increase the effectiveness of emergency department provider staffing. *Society for Academic Emergency Medicine*, januari 2006.
 - [4] Eri Kuno en Tony E. Smith Naoru Koizumi. Modeling patient flows using a queueing network with blocking. *Health Care Management Science*, juli 2005.
 - [5] David Ito en John E. Mueller Susan L. Albin, Jeffrey Barrett. A queueing network analysis of a health center. *Queueing Systems*, maart 1990.
 - [6] Qinan Wang. Modeling and analysis of high risk patient queues. *European journal of operational research*, oktober 2002.
 - [7] Wayne L. Winston. *Operations Research Applications and Algorithms*. Brooks/cole, 2004.

APPENDIX

A. SYMBOLENIJST

Tabel 11: Een overzicht van de gebruikte symbolen

Betekenis	Symbool
Behandelingsweek met controle bij de verpleegkundig specialist	V
Behandelingsweek met controle bij de longarts	L
Behandelingsweek met controle bij de verpleegkundig specialist en een CT-scan	V+CT
De kans op type n	$y(n)$
Het aantal verschillende type patiënten	N
De kans op het verlaten van het systeem	p
Aantal verschillende wachtrijen	J
Aantal patiënten in wachtrij j	n_j
Totale service prestatie wachtrij j	$\phi_j(n_j)$
Deel van de totale service prestatie van wachtrij j veroorzaakt door patiënt op positie l	$\gamma_j(l, n)$
Aantal servers bij wachtrij j	K_j
Kans dat een patiënt die arriveert bij wachtrij j op positie l gaat staan	$\delta(l, n_j + 1)$
Het type van de patiënt in wachtrij j op positie l	$t_j(l)$
De fase van de patiënt in wachtrij j op positie l	$s_j(l)$
De klasse van de patiënt in wachtrij j op positie l	$c_j(t_j(l), s_j(l))$
Toestand wachtrij j	$\mathbf{c}_j$
Markov proces representatie van de toestand van het systeem	$\mathbf{C}$
Gemiddeld aantal patiënten dat arriveert bij wachtrij met toestand i en fase s	$\alpha_j(i, s)$
Gemiddeld aantal patiënten dat arriveert bij wachtrij j per tijds-eenheid	α_j
Evenwichtsverdeling wachtrij j	$\pi_j(\mathbf{C})$
Evenwichtsverdeling netwerk	$\pi(\mathbf{C})$
Wachtrij met Poisson aankomstproces, exponentieel verdeelde bediening en s servers	M/M/s
Verwerkingsintensiteit	μ
Aankomstintensiteit	λ
Bezettingsgraad	ρ
Gemiddeld aantal patiënten in het systeem	$E\{L\}$
Gemiddelde verblijftijd per patiënte	$E\{F\}$

B. DISCRETE EVENT SIMULATION

In deze bijlage wordt de simulatie die gedaan is met behulp van discrete event simulation beschreven. Eerst worden de gebeurtenissen beschreven, vervolgens de tijd en tenslotte is er een flowchart gemaakt.

B.1. BESCHRIJVING GEBEURTENISSEN

1. Wanneer er een aankomst bij de longarts is, moet er bepaald worden wat de klasse van de patiënt is. Er wordt gekeken welke klasse de maximale aankomstintensiteit heeft. De aankomende patiënt is van deze klasse. Het aantal patiënten van deze klasse dat nog moet komen in deze behandelingsweek wordt met één vermindert. Wanneer de patiënt nieuw is moet het type van de patiënt bepaald worden. Dit wordt gedaan met de Bernoulli verdeling zoals hierboven genoemd is. Wanneer de patiënt geen nieuwe patiënt is, heeft deze patiënt al een type. Dus hoeft deze niet meer bepaald te worden. Aangezien de behandelingsweek begint bij de longarts wordt het aankomsttijdstip van de patiënt opgeslagen, zodat de verblijftijd van de patiënt in het systeem bepaald kan worden. Vervolgens wordt de bedieningsduur bepaald door een trekking te doen uit de deterministische verdeling met $\mu = \frac{1}{45}$ bij een nieuwe patiënt, aangezien de gemiddelde bedieningsduur 45 minuten is. Voor een patiënt van een andere klasse wordt er een trekking uit de deterministische verdeling gedaan met $\mu = \frac{1}{15}$, aangezien een controle gemiddeld 15 minuten duurt. Nu kan de volgende aankomst gepland worden. Dit wordt gedaan door een trekking te doen uit de deterministische verdeling met $\mu = \frac{\text{aantal aankomsten longarts}}{\text{lengte werkweek}}$. Wanneer hier de huidige tijd bij opgeteld wordt, is dit de volgende aankomst. Er moet natuurlijk wel eerst gekeken worden of de longarts op dit moment wel open is. Wan-

neer dit niet het geval is of wanneer er een periode tussen de huidige tijd en de nieuwe aankomst is waarin de longarts gesloten is, wordt de lengte van deze periode opgeteld bij de volgende aankomst. Vervolgens wordt er gekeken of de longarts bezet is. Wanneer dit niet het geval is kan de patiënt die is aangekomen meteen in behandeling bij de longarts. Nu is de longarts dus wel bezet en wordt er opgeslagen dat de longarts gedurende de bedieningsduur aan het werk is. Hierdoor kan later de bezettingsgraad van de longarts bepaald worden. De gebeurtenis 2 (Een vertrek bij de longarts) kan nu ook gepland worden. Dit is namelijk de tijd plus de bedieningsduur. Wanneer de longarts wel bezet is, sluit de patiënt achteraan in de wachtrij. De lengte van de wachtrij is nu verhoogd met 1.

2. Er wordt gekeken of dit de laatste fase van deze patiënt is. Wanneer dit het geval is verlaat de patiënt het systeem en wordt de verblijftijd van deze patiënt in deze behandelingsweek opgeslagen. Wanneer dit niet het geval is gaat de patiënt door naar het priklab, dus de patiënt is een fase verder. Er wordt gekeken of er al meer patiënten onderweg zijn naar het priklab. Wanneer dit niet het geval is, is gebeurtenis 3 (aankomst priklab) nu, er wordt dus geen rekening gehouden met looptijden. Wanneer er al wel een patiënt onderweg is naar het priklab, wordt deze patiënt met zijn type en huidige tijd toegevoegd aan de patiënten die onderweg zijn naar het priklab. Vervolgens is gebeurtenis 3 het minimum van de aankomsttijdstippen van de verschillende patiënten bij het priklab.

Tenslotte wordt er gekeken of de wachtrij bij de longarts leeg is. Wanneer dit niet het geval is, wordt de eerste patiënt in de wachtrij behandeld bij de longarts. Er staat nu dus 1 patiënt minder in de wachtrij. Vervolgens wordt er een trekking uit de deterministische verdeling gedaan met $\mu = \frac{1}{15}$. Hieruit volgt

de bedieningsduur van de patiënt. De tijd dat de longarts bezet is, is nu dus de tijd die hij al bezet is geweest plus de bedieningsduur. De nieuwe gebeurtenis 2 (vertrek longarts) is de huidige tijd plus de bediening. Wanneer er niemand in de wachtrij staat, is de nieuwe gebeurtenis 2 (vertrek longarts) oneindig. Aangezien er geen patiënt kan vertrekken, wanneer er niemand in bediening is.

3. Er wordt eerst gekeken of het priklab bezet is. Wanneer dit niet het geval is, kan de patiënt meteen in bediening. Hierdoor is het priklab nu wel bezet. Vervolgens wordt er een trekking uit de deterministische verdeling gedaan met $\mu = \frac{1}{20}$. De tijd dat het priklab bezet is, is nu de huidige tijd dat het priklab bezet was plus de bedieningsduur. Gebeurtenis 4 (vertrek priklab) is de huidige tijd plus de bedieningsduur. Wanneer het priklab wel bezet is, sluit deze patiënt achter aan in de wachtrij. Het aantal patiënten in de wachtrij is nu dus met 1 verhoogd. Vervolgens wordt er gekeken of er nog meer patiënten onderweg naar het priklab zijn. Wanneer dit niet het geval is, wordt gebeurtenis 3 (aankomst priklab) oneindig, omdat er nog geen patiënten zijn die een aankomst bij het priklab hebben. Wanneer er wel patiënten onderweg zijn naar het priklab, wordt het minimum van deze aankomsten genomen. Dit minimum is de nieuwe gebeurtenis 3 (aankomst priklab). Het aantal patiënten dat onderweg is, is nu dus verminderd met 1.
4. De patiënt gaat weer een fase verder. Eerst wordt er gekeken of er nog meer patiënten onderweg naar de dagbehandeling zijn. Wanneer dit niet het geval is, is gebeurtenis 5 (aankomst dagbehandeling) de huidige tijd. Wanneer er wel meer patiënten onderweg zijn naar de dagbehandeling, wordt deze patiënt hieraan toegevoegd. Vervolgens wordt het minimum genomen over de aankomsten van de patiënten bij de dagbehandeling.

Dit is dan gebeurtenis 5 (aankomst dagbehandeling). Vervolgens wordt er gekeken of de wachtrij van het priklab leeg is. Wanneer dit niet het geval is, gaat de eerste patiënt in de wachtrij in bediening. Er staat nu dus 1 patiënt minder in de wachtrij. Vervolgens wordt een trekking gedaan uit de deterministische verdeling met $\mu = \frac{1}{20}$ om de bedieningsduur van deze patiënt te bepalen. De tijd dat het priklab nu bezet is, is de tijd dat het priklab al bezet is geweest plus de bedieningsduur. De nieuwe gebeurtenis 4 (vertrek priklab) is de huidige tijd plus de bedieningsduur. Wanneer er niemand in de wachtrij staat voor het priklab, is de nieuwe gebeurtenis 4 (vertrek priklab) oneindig, aangezien er niemand kan vertrekken als er niemand in bediening is.

5. Eerst wordt er gekeken of de dagbehandeling bezet is. Wanneer dit niet het geval is, kan de patiënt meteen in bediening. De dagbehandeling is nu dus wel bezet. Er kan nu een trekking gedaan worden uit de deterministische verdeling met $\mu = \frac{1}{45}$ om de bedieningsduur van deze patiënt te bepalen. De tijd dat de dagbehandeling bezet is, is de tijd dat de dagbehandeling tot nu toe bezet is geweest plus de bedieningsduur. Gebeurtenis 6 (vertrek dagbehandeling) is de huidige tijd plus de bedieningsduur. Wanneer de dagbehandeling wel bezet is, sluit de patiënt achteraan in de wachtrij. Het aantal patiënten in de wachtrij is nu dus met 1 verhoogd. Vervolgens wordt er gekeken of er nog meer patiënten onderweg zijn naar de dagbehandeling. Wanneer dit het geval is, wordt het minimum van deze aankomsten genomen en is gebeurtenis 5 (aankomst dagbehandeling) dit minimum. Het aantal patiënten dat onderweg is, wordt nu dus verminderd met 1. Wanneer er niet meer patiënten onderweg zijn naar de dagbehandeling, wordt gebeurtenis 5 (aankomst dagbehandeling) oneindig.

6. De patiënt die nu gaat vertrekken bij de dagbehandeling, is weer een fase verder wanneer dit gebeurt. Er wordt nu bekeken wat de volgende fase van de patiënt is. Wanneer dit de longarts is, wordt er opgeslagen dat deze patiënt de volgende behandelingsweek weer bij de longarts komt. Vervolgens wordt zijn verblijftijd in het systeem deze behandelingsweek bepaald. Wanneer de volgende fase de verpleegkundig specialist is, wordt er opgeslagen dat deze patiënt de volgende behandelingsweek weer bij de verpleegkundig specialist begint. Vervolgens wordt de verblijftijd van de patiënt bepaald van afgelopen behandelingsweek. Wanneer de patiënt als volgende fase niet de longarts of verpleegkundig specialist heeft, moet de patiënt naar de CT-scan. Deze CT-scan wordt nog wel deze behandelingsweek gemaakt. Dus wordt er gekeken of er meer patiënten onderweg zijn naar de CT-scan. Wanneer dit niet het geval is, is gebeurtenis 9 (aankomst CT-scan) de huidige tijd. Wanneer er wel meer patiënten onderweg zijn naar de CT-scan, wordt deze patiënt daaraan toegevoegd en wordt hier het minimum over de aankomsttijdstippen genomen. Dit minimum is de gebeurtenis 9 (aankomst CT-scan).
Tenslotte wordt er gekeken of de wachtrij van de dagbehandeling leeg is. Wanneer dit niet het geval is, kan de eerste patiënt in de wachtrij geholpen worden. Het aantal patiënten in de wachtrij voor de dagbehandeling is dan met 1 verminderd. Er wordt weer een trekking gedaan uit de deterministische verdeling met $\mu = \frac{1}{45}$ om de bedieningsduur van deze patiënt te bepalen. Vervolgens wordt gebeurtenis 6 (vertrek dagbehandeling) gepland, dit is de huidige tijd plus de bedieningsduur. Wanneer er niemand in de wachtrij staat, kan er geen gebeurtenis 6 (vertrek dagbehandeling) gepland worden. Dus is deze gebeurtenis oneindig.
7. Wanneer er een patiënt aankomt bij de verpleegkundig specialist, wordt er eerst gekeken wat de klasse is van de patiënt. Dit wordt op dezelfde manier gedaan als bij de longarts. Vervolgens wordt er net zoals bij de longarts een nieuwe aankomst gepland worden. Nu kan er gekeken worden of de verpleegkundig specialist bezet is. Wanneer dit het geval is, sluit de patiënt achteraan in de wachtrij. Het aantal wachtende patiënten wordt dus verhoogd met 1. Wanneer de verpleegkundig specialist niet bezet is, kan de patiënt meteen in bediening. Er wordt een trekking gedaan uit de deterministische verdeling met $\mu = \frac{1}{15}$ om de bedieningsduur te bepalen. De tijd dat de verpleegkundig specialist bezet is wordt verhoogd met de bedieningsduur. Tenslotte wordt een gebeurtenis 8 (vertrek verpleegkundig specialist) gepland, dit is namelijk de huidige tijd plus de bedieningsduur.
8. Wanneer een patiënt vertrekt bij de verpleegkundig specialist, is de patiënt weer een fase verder. Vervolgens wordt er gekeken of er meer patiënten onderweg zijn naar het priklab. Wanneer dit niet het geval is, kan een gebeurtenis 2 (aankomst priklab) gepland worden, dit is namelijk de huidige tijd. Wanneer er meer patiënten onderweg zijn naar het priklab, wordt deze patiënt hierbij gevoegd. Hier wordt vervolgens het minimum over de geplande aankomsttijden genomen. De uitkomst hiervan is de nieuwe gebeurtenis 3 (aankomst priklab). Vervolgens wordt er gekeken of de wachtrij bij de verpleegkundig specialist leeg is. Wanneer dit niet het geval is, wordt de eerste patiënt van de wachtrij geholpen. Er staat nu dus 1 patiënt minder in de wachtrij. Om de bedieningsduur te bepalen, wordt er nu een trekking uit de deterministische verdeling met $\mu = \frac{1}{15}$. De tijd dat de verpleegkundig specialist aan het werk is geweest, wordt verhoogd met de bedieningsduur. Verder wordt gebeurtenis 8 (vertrek verpleegkundig specialist) ge-

pland, dit is de huidige tijd plus de bedieningsduur. Wanneer er niemand in de wachtrij staat, is gebeurtenis 8 oneindig.

9. Wanneer een patiënt aankomt bij de CT-scan, wordt ook hier eerst gekeken of de CT-scan bezet is. Wanneer dit niet het geval is, kan de patiënt meteen in bediening. Er wordt voor de bedieningsduur een trekking uit de deterministische verdeling met $\mu = \frac{1}{30}$ gedaan. De tijd dat de CT-scan bezet is wordt nu verhoogd met de bedieningsduur. De gebeurtenis 10 (vertrek CT-scan) kan gepland worden, dit is namelijk de huidige tijd plus de bedieningsduur. Wanneer de CT-scan wel bezet is, sluit de patiënt achteraan in de wachtrij. Het aantal patiënten in de wachtrij wordt nu met 1 verhoogd. Vervolgens wordt er gekeken of er nog meer patiënten onderweg waren naar de CT-scan. Wanneer dit het geval is, kan het minimum over de aankomsttijdstippen genomen worden. Dit minimum is gebeurtenis 9 (aankomst CT-scan). Wanneer er verder geen patiënten onderweg zijn naar de CT-scan, komt er geen nieuwe patiënt bij de CT-scan aan. Dus is gebeurtenis 9 (aankomst CT-scan) oneindig.

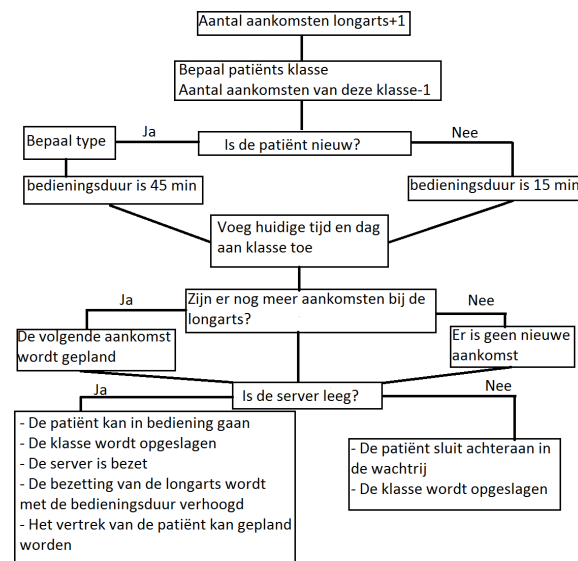
10. Wanneer een patiënt vertrekt bij de CT-scan, gaat de patiënt weer een fase verder. Deze patiënt wordt volgende week weer opgeroepen bij de longarts. Vervolgens kan de verblijftijd van de patiënt bepaald worden. Wanneer er nog patiënten in de wachtrij voor de CT-scan staan, kan de eerste in bediening gaan. Er staat dan dus 1 patiënt minder in de wachtrij. Er kan nu een trekking uit de deterministische verdeling gedaan worden met $\mu = \frac{1}{30}$ om de bedieningsduur te bepalen. De tijd dat de CT-scan bezet is, wordt verhoogd met de bedieningsduur. Vervolgens is gebeurtenis 10 (vertrek CT-scan) de huidige tijd plus de bedieningsduur. Wanneer de wachtrij leeg is, kan er geen patiënt bediend worden. Hierdoor kan er ook geen patiënt vertrekken bij de CT-scan,

dus gebeurtenis 10 (vertrek CT-scan) is oneindig.

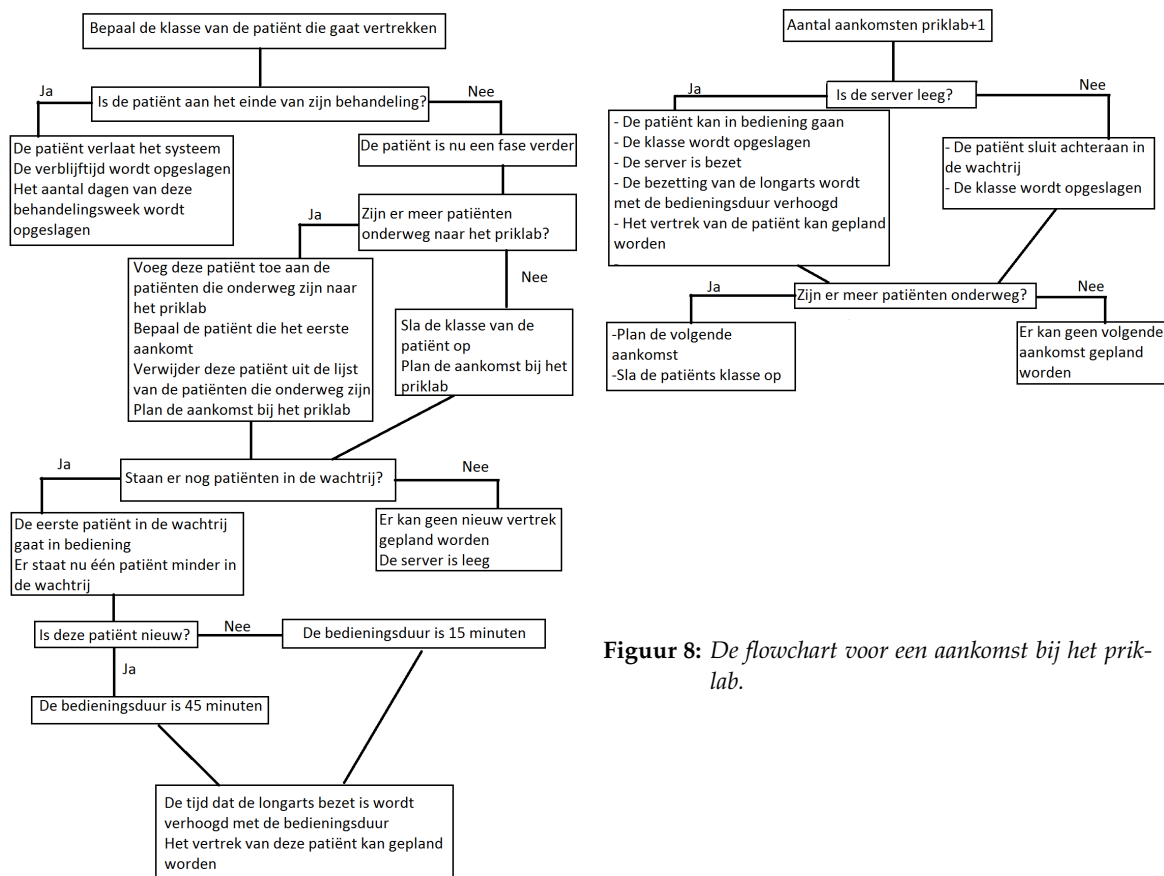
B.2. DE TIJD

In deze paragraaf wordt er gekeken welke stations op een bepaald tijdstip open zijn. De week begint op maandag 8:30 uur. Dit wordt dus uur 0 van de simulatie. Vervolgens is dus 9:30 uur het tweede uur van de simulatie enzovoorts. Dit is te zien in Tabel 12

B.3. FLOWCHART

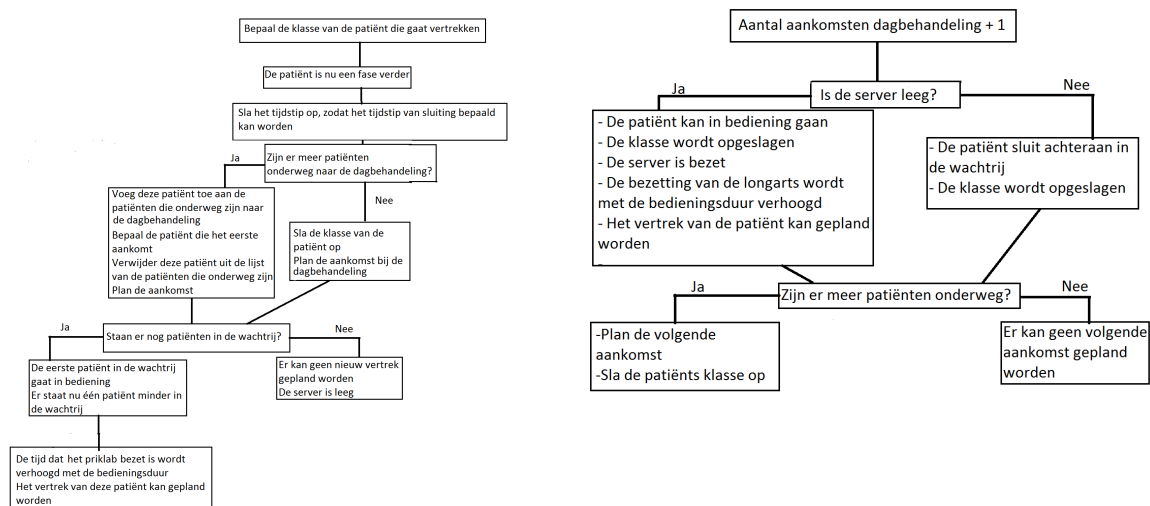


Figuur 6: De flowchart voor een aankomst bij de longarts.



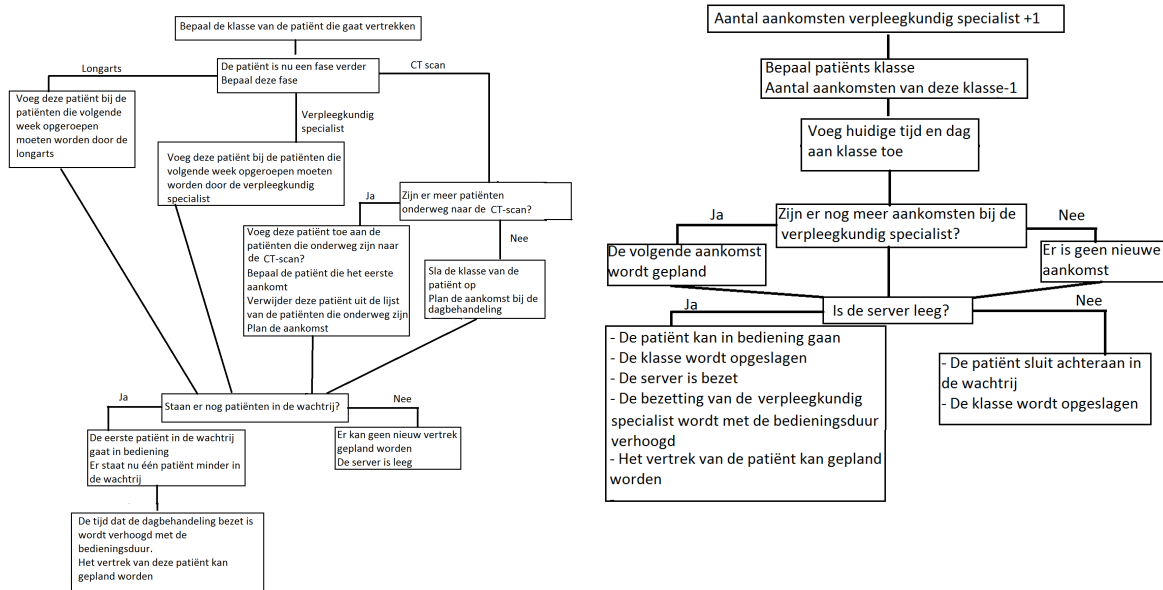
Figuur 8: De flowchart voor een aankomst bij het prik-lab.

Figuur 7: De flowchart voor een vertrek bij de longarts.



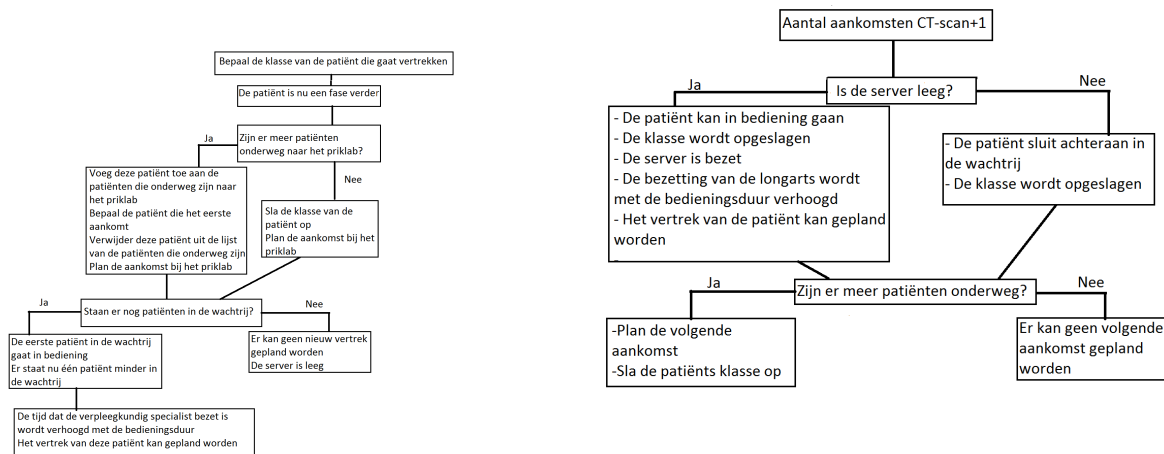
Figuur 10: De flowchart voor een aankomst bij de dagbehandeling.

Figuur 9: De flowchart voor een vertrek bij het priklab.



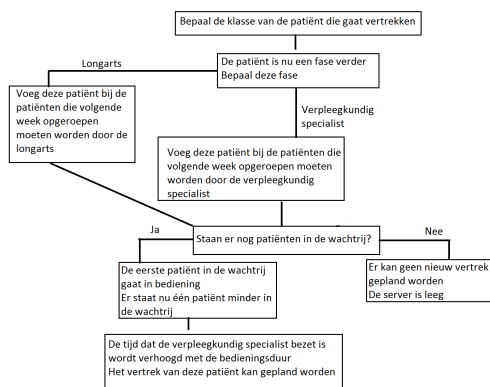
Figuur 12: De flowchart voor een aankomst bij de verpleegkundig specialist.

Figuur 11: De flowchart voor een vertrek bij de dagbehandeling.



Figuur 14: De flowchart voor een aankomst bij de CT-scan.

Figuur 13: De flowchart voor een vertrek bij de verpleegkundig specialist.



Figuur 15: De flowchart voor een vertrek bij de CT-scan.

Tabel 12: Een tabel met een overzicht van de opening van de verschillende stations in uren van de simulatie en tijdstippen in de week.

Dag	Klok uren	Uren simulatie	Geopende stations
Maandag	8:30-14:30	0-6	Alle stations
Maandag	14:30-16:00	6-7.5	Priklab, dagbehandeling en CT-scan
Maandag	16:00-17:30	7.5-9	Priklab
Dinsdag	7:30-8:00	9-9.5	Priklab
Dinsdag	8:00-8:30	9.5-10	Priklab, dagbehandeling en CT-scan
Dinsdag	8:30-14:30	10-16	Alle stations
Dinsdag	14:30-16:00	16-17.5	Priklab, dagbehandeling en CT-scan
Dinsdag	16:00-17:30	17.5-19	Priklab
Woensdag	7:30-8:00	19-19.5	Priklab
Woensdag	8:00-8:30	19.5-20	Priklab, dagbehandeling en CT-scan
Woensdag	8:30-14:30	20-26	Alle stations
Woensdag	14:30-16:00	26-27.5	Priklab, dagbehandeling en CT-scan
Woensdag	16:00-17:30	27.5-29	Priklab
Donderdag	7:30-8:00	29-29.5	Priklab
Donderdag	8:00-8:30	29.5-30	Priklab, dagbehandeling en CT-scan
Donderdag	8:30-14:30	30-36	Alle stations
Donderdag	14:30-16:00	36-37.5	Priklab, dagbehandeling en CT-scan
Donderdag	16:00-17:30	37.5-39	Priklab
Vrijdag	7:30-8:00	39-39.5	Priklab
Vrijdag	8:00-8:30	39.5-40	Priklab, dagbehandeling en CT-scan
Vrijdag	8:30-14:30	40-46	Alle stations
Vrijdag	14:30-16:00	46-47.5	Priklab, dagbehandeling en CT-scan
Zaterdag	8:00-16:00	47.5-54.5	CT-scan

C. AANTAL NIEUWE PATIËNTEN

C.1. HUIDIGE BEHANDELINGSPROCES

Tabel 13: Een tabel met een overzicht van het aantal nieuwe patiënten per week bij verschillende capaciteiten.

Minimaal aantal servers longarts	Minimaal aantal servers priklab	Minimaal aantal servers dagbehande- ling	Minimaal aantal servers verpleegkun- dig specialist	Minimaal aan- tal servers CT- scan	Aantal pati- ënten
1	1	1	1	1	5
1	1	2	1	1	11
1	1	3	1	1	14
1	2	3	1	1	17
1	2	4	1	1	20
2	2	4	1	1	22
2	2	5	1	1	25
2	2	5	2	1	28
2	2	6	2	1	29
2	3	6	2	1	34
2	3	7	2	1	38
2	3	7	2	2	39
2	3	8	2	2	41
3	3	8	2	2	44
3	4	8	2	2	45
3	4	9	2	2	50
3	4	9	3	2	51
3	4	10	3	2	59

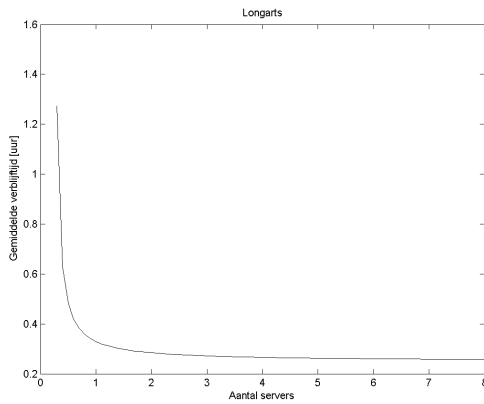
C.2. BEHANDELINGSPROCES VAN MAXIMAAL EEN HALF JAAR

Tabel 14: Een tabel met een overzicht van het aantal nieuwe patiënten per week bij verschillende capaciteiten bij een behandelingsproces van maximaal een half jaar

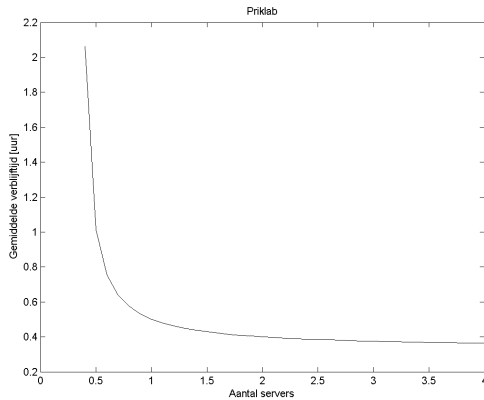
Minimaal aantal servers longarts	Minimaal aantal servers priklab	Minimaal aantal servers dagbehande- ling	Minimaal aantal servers verpleegkun- dig specialist	Minimaal aan- tal servers CT- scan	Aantal pati- ënten
1	1	1	1	1	7
1	1	2	1	1	14
1	1	3	1	1	18
1	2	3	1	1	21
1	2	4	1	1	25
2	2	4	1	1	28
2	2	5	1	1	32
2	2	5	2	1	36
2	2	6	2	1	37
2	3	6	2	1	43
2	3	7	2	1	46
2	3	7	2	2	50
2	3	8	2	2	51
3	3	8	2	2	56
3	4	8	2	2	57
3	4	9	2	2	65
3	4	10	3	2	72
3	4	11	3	2	75

D. GEMIDDELDE VERBLIJFTIJD BIJ EEN STATION BIJ VERSCHILLENDE SERVERS.

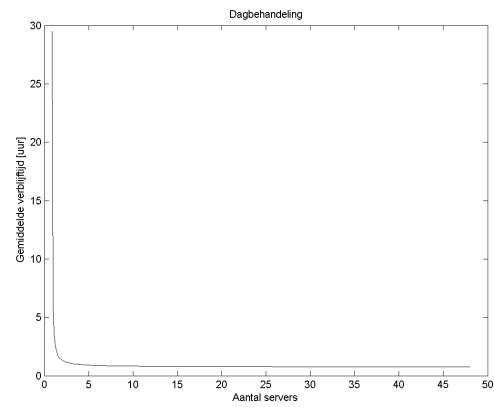
D.1. HUIDIGE BEHANDELINGSPROCES



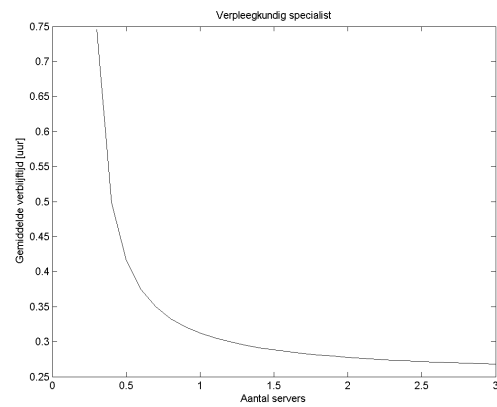
Figuur 16: De gemiddelde verblijftijd tegen het aantal servers bij de longarts met een behandelingsproces met elf verschillende types.



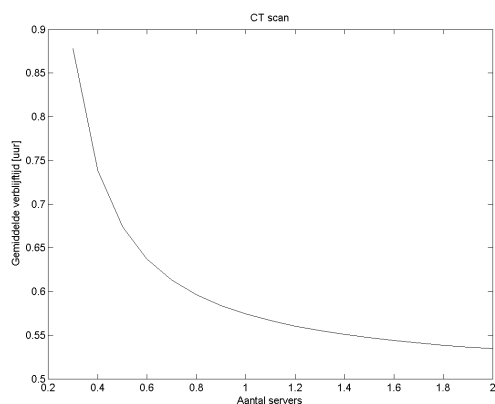
Figuur 17: De gemiddelde verblijftijd tegen het aantal servers bij het priklab met een behandelingsproces met elf verschillende types.



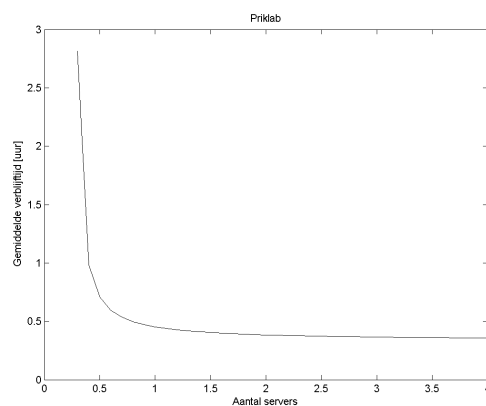
Figuur 18: De gemiddelde verblijftijd tegen het aantal servers bij de dagbehandeling met een behandelingsproces met elf verschillende types.



Figuur 19: De gemiddelde verblijftijd tegen het aantal servers bij de verpleegkundig specialist met een behandelingsproces met elf verschillende types.

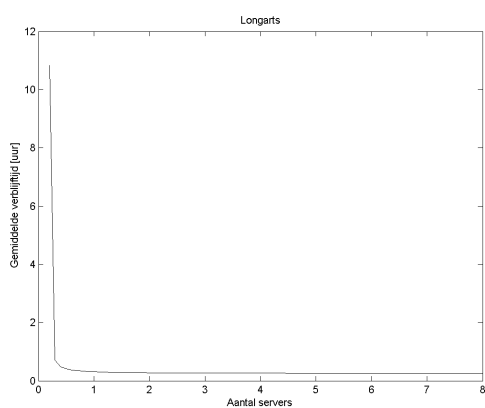


Figuur 20: De gemiddelde verblijftijd tegen het aantal servers bij de CT-scan met een behandelingsproces met elf verschillende types.

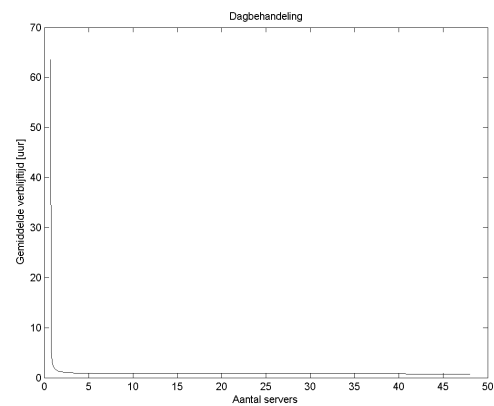


Figuur 22: De gemiddelde verblijftijd tegen het aantal servers bij het priklab met een behandelingsproces met zes verschillende types en maximaal zes maanden.

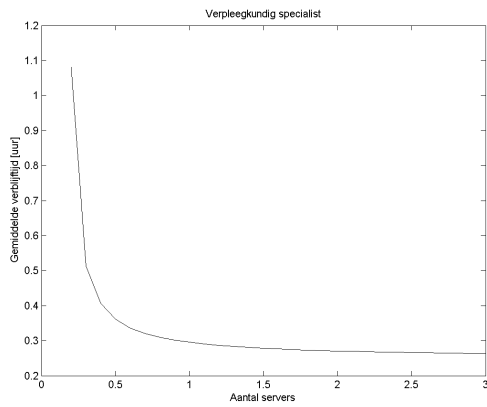
D.2. BEHANDELINGSPROCES VAN MAXIMAAL EEN HALF JAAR



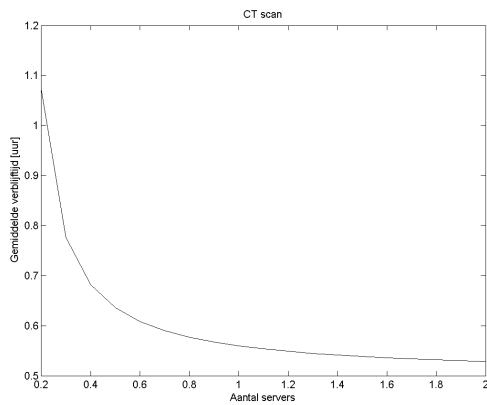
Figuur 21: De gemiddelde verblijftijd tegen het aantal servers bij de longarts met een behandelingsproces met zes verschillende types en maximaal zes maanden.



Figuur 23: De gemiddelde verblijftijd tegen het aantal servers bij de dagbehandeling met een behandelingsproces met zes verschillende types en maximaal zes maanden.



Figuur 24: De gemiddelde verblijftijd tegen het aantal servers bij de verpleegkundig specialist met een behandelingsproces met zes verschillende types en maximaal zes maanden.



Figuur 25: De gemiddelde verblijftijd tegen het aantal servers bij de CT-scan met een behandelingsproces met zes verschillende types en maximaal zes maanden.

E. FASES

Tabel 15: Een tabel met de fases per type patiënt

Fase	Type a	Type b	Type c	Type d	Type e	Type f
1	longarts	longarts	longarts	longarts	longarts	longarts
2	priklab	priklab	priklab	priklab	priklab	priklab
3	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
4	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
5	priklab	priklab	priklab	priklab	priklab	priklab
6	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
7	longarts	longarts	longarts	longarts	longarts	longarts
8	priklab	priklab	priklab	priklab	priklab	priklab
9	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
10	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
11	priklab	priklab	priklab	priklab	priklab	priklab
12	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
13	CT-scan	CT-scan	CT-scan	CT-scan	CT-scan	CT-scan
14	longarts	longarts	longarts	longarts	longarts	longarts
15		priklab	priklab	priklab	priklab	priklab
16		dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
17		verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
18		priklab	priklab	priklab	priklab	priklab
19		dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
20		longarts	longarts	longarts	longarts	longarts
21		priklab	priklab	priklab	priklab	priklab
22		dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
23		verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
24		priklab	priklab	priklab	priklab	priklab
25		dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
26		CT-scan	CT-scan	CT-scan	CT-scan	CT-scan
27		longarts	longarts	longarts	longarts	longarts
28			priklab	priklab	priklab	priklab
29			dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
30			verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
31			priklab	priklab	priklab	priklab
32			dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
33			longarts	longarts	longarts	longarts
33			longarts	longarts	longarts	longarts
34			priklab	priklab	priklab	priklab

Fase	Type a	Type b	Type c	Type d	Type e	Type f
35			dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
36			verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
37			priklab	priklab	priklab	priklab
38			dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
39			CT-scan	CT-scan	CT-scan	CT-scan
40			longarts	longarts	longarts	longarts
41				priklab	priklab	priklab
42				dagbehandeling	dagbehandeling	dagbehandeling
43				verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
44				priklab	priklab	priklab
45				dagbehandeling	dagbehandeling	dagbehandeling
46				longarts	longarts	longarts
47				priklab	priklab	priklab
48				dagbehandeling	dagbehandeling	dagbehandeling
49				verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
50				priklab	priklab	priklab
51				dagbehandeling	dagbehandeling	dagbehandeling
52				CT-scan	CT-scan	CT-scan
53				longarts	longarts	longarts
54					priklab	priklab
55					dagbehandeling	dagbehandeling
56					verpleegkundig specialist	verpleegkundig specialist
57					priklab	priklab
58					dagbehandeling	dagbehandeling
59					longarts	longarts
60					priklab	priklab
61					dagbehandeling	dagbehandeling
62					verpleegkundig specialist	verpleegkundig specialist
63					priklab	priklab
64					dagbehandeling	dagbehandeling

Fase	Type a	Type b	Type c	Type d	Type e	Type f
65					CT-scan	CT-scan
66					longarts	longarts
67						priklab
68						dagbehandeling
69						verpleegkundig specialist
70						priklab
71						dagbehandeling
72						longarts
73						priklab
74						dagbehandeling
75						verpleegkundig specialist
76						priklab
77						dagbehandeling
78						longarts
79						priklab
80						dagbehandeling
81						verpleegkundig specialist
82						priklab
83						dagbehandeling
84						longarts

Tabel 16: Een tabel met de fases per type patiënt volgens het netwerk met stations per week zoals te zien is in Figuur 28

Fase	Type a	Type b	Type c	Type d	Type e	Type f
1	behandelweek met intake	behandelweek met intake	behandelweek met intake	behandelweek met intake	behandelweek met intake	behandelweek met intake
2	behandelweek met verpleegkundig specialist	behandelweek met verpleegkundig specialist	behandelweek met verpleegkundig specialist	behandelweek met verpleegkundig specialist	behandelweek met verpleegkundig specialist	behandelweek met verpleegkundig specialist
3	behandelweek met longarts	behandelweek met longarts	behandelweek met longarts	behandelweek met longarts	behandelweek met longarts	behandelweek met longarts
4	behandelweek met CT-scan	behandelweek met CT-scan	behandelweek met CT-scan	behandelweek met CT-scan	behandelweek met CT-scan	behandelweek met CT-scan
5	longarts	behandelweek met longarts	behandelweek met longarts	behandelweek met longarts	behandelweek met longarts	behandelweek met longarts
6		behandelweek met verpleegkundig specialist	behandelweek met verpleegkundig specialist	behandelweek met verpleegkundig specialist	behandelweek met verpleegkundig specialist	behandelweek met verpleegkundig specialist
7		behandelweek met longarts	behandelweek met longarts	behandelweek met longarts	behandelweek met longarts	behandelweek met longarts
8		behandelweek met CT-scan	behandelweek met CT-scan	behandelweek met CT-scan	behandelweek met CT-scan	behandelweek met CT-scan
9		longarts	behandelweek met longarts	behandelweek met longarts	behandelweek met longarts	behandelweek met longarts
10			behandelweek met verpleegkundig specialist	behandelweek met verpleegkundig specialist	behandelweek met verpleegkundig specialist	behandelweek met verpleegkundig specialist
11			behandelweek met longarts	behandelweek met longarts	behandelweek met longarts	behandelweek met longarts
12			behandelweek met CT-scan	behandelweek met CT-scan	behandelweek met CT-scan	behandelweek met CT-scan
13			longarts	behandelweek met longarts	behandelweek met longarts	behandelweek met longarts
14				behandelweek met verpleegkundig specialist	behandelweek met verpleegkundig specialist	behandelweek met verpleegkundig specialist
15				behandelweek met longarts	behandelweek met longarts	behandelweek met longarts
16				behandelweek met CT-scan	behandelweek met CT-scan	behandelweek met CT-scan
17				longarts	behandelweek met longarts	behandelweek met longarts
18					behandelweek met verpleegkundig specialist	behandelweek met verpleegkundig specialist

Fase	Type a	Type b	Type c	Type d	Type e	Type f
19					behandelweek met longarts	behandelweek met longarts
20					behandelweek met CT-scan	behandelweek met CT-scan
21					longarts	behandelweek met longarts
22						behandelweek met verpleegkundig specialist
23						behandelweek met longarts
24						behandelweek met verpleegkundig specialist
25						behandelweek met longarts
26						behandelweek met verpleegkundig specialist
27						longarts

Tabel 17: Een tabel met de fases per type patiënt

Fase	Type a	Type b	Type c
1	longarts	longarts	longarts
2	priklab	priklab	priklab
3	dagbehandeling	dagbehandeling	dagbehandeling
4	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
5	priklab	priklab	priklab
6	dagbehandeling	dagbehandeling	dagbehandeling
7	longarts	longarts	longarts
8	priklab	priklab	priklab
9	dagbehandeling	dagbehandeling	dagbehandeling
10	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
11	priklab	priklab	priklab
12	dagbehandeling	dagbehandeling	dagbehandeling
13	CT-scan	CT-scan	CT-scan
14	longarts	longarts	longarts
15		priklab	priklab
16		dagbehandeling	dagbehandeling
17		verpleegkundig specialist	verpleegkundig specialist
18		priklab	priklab
19		dagbehandeling	dagbehandeling
20		longarts	longarts
21		priklab	priklab
22		dagbehandeling	dagbehandeling
23		verpleegkundig specialist	verpleegkundig specialist
24		priklab	priklab
25		dagbehandeling	dagbehandeling
26		CT-scan	CT-scan
27		longarts	longarts
28			priklab
29			dagbehandeling
30			verpleegkundig specialist
31			priklab
32			dagbehandeling
33			longarts
34			priklab
35			dagbehandeling
36			verpleegkundig specialist
37			priklab
38			dagbehandeling
39			longarts

Tabel 18: Een tabel met de fases per type patiënt 1 tot en met 6

Fase	Type 1	Type 2	Type 3	Type 4	Type 5	Type 6
1	longarts	longarts	longarts	longarts	longarts	longarts
2	priklab	priklab	priklab	priklab	priklab	priklab
3	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
4	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
5	priklab	priklab	priklab	priklab	priklab	priklab
6	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
7	CT-scan	longarts	longarts	longarts	longarts	longarts
8	longarts	priklab	priklab	priklab	priklab	priklab
9		dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
10		verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
11		priklab	priklab	priklab	priklab	priklab
12		dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
13		CT-scan	CT-scan	CT-scan	CT-scan	CT-scan
14		longarts	longarts	longarts	longarts	longarts
15			priklab	priklab	priklab	priklab
16			dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
17			verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
18			priklab	priklab	priklab	priklab
19			dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
20			CT-scan	longarts	longarts	longarts
21			longarts	priklab	priklab	priklab
22				dagbehandeling	dagbehandeling	dagbehandeling
23				verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
24				priklab	priklab	priklab
25				dagbehandeling	dagbehandeling	dagbehandeling
26				CT-scan	CT-scan	CT-scan
27				longarts	longarts	longarts
28					priklab	priklab
29					dagbehandeling	dagbehandeling
30					verpleegkundig specialist	verpleegkundig specialist
31					priklab	priklab
32					dagbehandeling	dagbehandeling
33					CT-scan	longarts
34					longarts	priklab
35						dagbehandeling
36						verpleegkundig specialist
37						priklab
38						dagbehandeling
39						CT-scan
40						longarts

Tabel 19: Een tabel met de fases per type patiënt 7 tot en met 11

Fase	Type 7	Type 8	Type 9	Type 10	Type 11
1	longarts	longarts	longarts	longarts	longarts
2	priklab	priklab	priklab	priklab	priklab
3	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
4	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
5	priklab	priklab	priklab	priklab	priklab
6	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
7	longarts	longarts	longarts	longarts	longarts
8	priklab	priklab	priklab	priklab	priklab
9	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
10	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
11	priklab	priklab	priklab	priklab	priklab
12	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
13	CT-scan	CT-scan	CT-scan	CT-scan	CT-scan
14	longarts	longarts	longarts	longarts	longarts
15	priklab	priklab	priklab	priklab	priklab
16	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
17	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
18	priklab	priklab	priklab	priklab	priklab
19	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
20	longarts	longarts	longarts	longarts	longarts
21	priklab	priklab	priklab	priklab	priklab
22	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
23	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
24	priklab	priklab	priklab	priklab	priklab
25	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
26	CT-scan	CT-scan	CT-scan	CT-scan	CT-scan
27	longarts	longarts	longarts	longarts	longarts
28	priklab	priklab	priklab	priklab	priklab
29	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
30	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
31	priklab	priklab	priklab	priklab	priklab
32	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
33	longarts	longarts	longarts	longarts	longarts
34	priklab	priklab	priklab	priklab	priklab
35	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
36	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
37	priklab	priklab	priklab	priklab	priklab
38	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
39	CT-scan	CT-scan	CT-scan	CT-scan	CT-scan

Fase	Type 7	Type 8	Type 9	Type 10	Type 11
40	longarts	longarts	longarts	longarts	longarts
41	priklab	priklab	priklab	priklab	priklab
42	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
43	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
44	priklab	priklab	priklab	priklab	priklab
45	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
46	CT-scan	longarts	longarts	longarts	longarts
47	longarts	priklab	priklab	priklab	priklab
48		dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
49		verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
50		priklab	priklab	priklab	priklab
51		dagbehandeling	dagbehandeling	dagbehandeling	dagbehandeling
52		CT-scan	CT-scan	CT-scan	CT-scan
53		longarts	longarts	longarts	longarts
54			priklab	priklab	priklab
55			dagbehandeling	dagbehandeling	dagbehandeling
56			verpleegkundig specialist	verpleegkundig specialist	verpleegkundig specialist
57			priklab	priklab	priklab
58			dagbehandeling	dagbehandeling	dagbehandeling
59			CT-scan	longarts	longarts
60			longarts	priklab	priklab
61				dagbehandeling	dagbehandeling
62				verpleegkundig specialist	verpleegkundig specialist
63				priklab	priklab
64				dagbehandeling	dagbehandeling
65				CT-scan	CT-scan
66				longarts	longarts
67					priklab

Fase	Type 7	Type 8	Type 9	Type 10	Type 11
68					dagbehandeling
69					verpleegkundig specialist
70					priklab
71					dagbehandeling
72					longarts
73					priklab
74					dagbehandeling
75					verpleegkundig specialist
76					priklab
77					dagbehandeling
78					longarts
79					priklab
80					dagbehandeling
81					verpleegkundig specialist
82					priklab
83					dagbehandeling
84					longarts

F. GEMIDDELDE VERBLIJFTIJD BEPALEN

```

lambda=5; % Aantal nieuwe patienten per week
Werkweek1=30;%Aantal uur in werkweek longarts
Werkweek2=46.5;%priklab
Werkweek3=40;%dagbehandeling
Werkweek4=30;%verpleegkundig specialist
Werkweek5=48;%CT scan
Bedieningsduur1=15/(60*Werkweek1);%De gemiddelde bedieningsduur van de longarts
Bedieningsduur2=20/(60*Werkweek2);%van de labprik
Bedieningsduur3=45/(60*Werkweek3);%van de dagbehandeling
Bedieningsduur4=15/(60*Werkweek4);%van de verpleegkundig specialist
Bedieningsduur5=30/(60*Werkweek5);%van de CT scan
i=11; % Aantal verschillende type patiënten
nu=zeros(1,i);% Aankomstintensiteiten per type
p=zeros(1,i);% Kans op een bepaald type
for i=1:11
    if i==11
        p(i)=0.8^(i-1);
    else
        p(i)=0.2*0.8^(i-1);
    end

    nu(i)=p(i)*lambda;

end
a1=2*nu(1)+3*nu(2)+4*nu(3)+5*nu(4)+6*nu(5)+7*nu(6)+8*nu(7)+9*nu(8)+...
    10*nu(9)+11*nu(10)+14*nu(11)
a2=2*nu(1)+4*nu(2)+6*nu(3)+8*nu(4)+10*nu(5)+12*nu(6)+14*nu(7)+16*nu(8)+...
    18*nu(9)+20*nu(10)+24*nu(11)
a3=2*nu(1)+4*nu(2)+6*nu(3)+8*nu(4)+10*nu(5)+12*nu(6)+14*nu(7)+16*nu(8)+...
    18*nu(9)+20*nu(10)+24*nu(11)
a4=nu(1)+2*nu(2)+3*nu(3)+4*nu(4)+5*nu(5)+6*nu(6)+7*nu(7)+8*nu(8)+...
    9*nu(9)+10*nu(10)+13*nu(11)
a5=nu(1)+nu(2)+2*nu(3)+2*nu(4)+3*nu(5)+3*nu(6)+4*nu(7)+4*nu(8)+...
    5*nu(9)+5*nu(10)+5*nu(11)
S1=a1*Bedieningsduur1
S2=a2*Bedieningsduur2
S3=a3*Bedieningsduur3
S4=a4*Bedieningsduur4
S5=a5*Bedieningsduur5
Uren_1=S1*Werkweek1
Uren_2=S2*Werkweek2
Uren_3=S3*Werkweek3
Uren_4=S4*Werkweek4
Uren_5=S5*Werkweek5
%% longarts
x=0.3:0.1:8;%Aantal servers variëren

```

```

mu=1/Bedieningsduur1;%mu bepalen
rho=a1./(mu.*x);%rho bepalen
y1=rho./(a1*(1-rho));%gemiddelde verblijftijd
y=y1.*Werkweek1.*x;%gemiddelde verblijftijd in uren
plot(x,y,'k-')
xlabel('Aantal_servers')
ylabel('Gemiddelde_verblijftijd_[uur]')
title('Longarts')
saveas(gcf,'longartskans.png')
Longarts=y(8)*60 %gemiddelde verblijftijd in minuten met 1 server gehele ...
%week beschikbaar in minuten
MaxServerLongarts=y(length(x))*60
VerschilLong=Longarts-MaxServerLongarts
%% lab prik
x=0.4:0.1:4;
mu=1/Bedieningsduur2;
rho=a2./(mu.*x);
y1=rho./(a2*(1-rho));
y=y1.*Werkweek2.*x;
plot(x,y,'k-')
xlabel('Aantal_servers')
ylabel('Gemiddelde_verblijftijd_[uur]')
title('Priklab')
saveas(gcf,'labprikkans.png')
lab=y(7)*60
MaxServerLab=y(length(x))*60
VerschilLab=lab-MaxServerLab
%% dagbehandeling
x=0.9:0.1:48;
mu=1/Bedieningsduur3;
rho=a3./(mu.*x);
y1=rho./(a3*(1-rho));
y=y1.*Werkweek3.*x;
plot(x,y,'k-')
xlabel('Aantal_servers')
ylabel('Gemiddelde_verblijftijd_[uur]')
title('Dagbehandeling')
saveas(gcf,'dagbehandelingkans.png')
dagbehandeling=y(2)*60
MaxServerDag=y(length(x))*60
VerschilDag=dagbehandeling-MaxServerDag
%% Verpleegkundig specialist
x=0.3:0.1:3;
mu=1/Bedieningsduur4;
rho=a4./(mu.*x);
y1=rho./(a4*(1-rho));
y=y1.*Werkweek4.*x;
plot(x,y,'k-')

```

```
xlabel( 'Aantal_servers ')\nylabel( 'Gemiddelde_verblijftijd_[uur] ')\ntitle( 'Verpleegkundig_specialist ')\nsaveas(gcf, 'verpleegkundigkans.png')\nverpleeg=y(8)*60\nMaxServerVerpleeg=y(length(x))*60\nVerschilVerpleeg=verpleeg-MaxServerVerpleeg\n%% CT Scan\nx=0.3:0.1:2;\nmu=1/Bedieningsduur5;\nrho=a5./(mu.*x);\ny1=rho./(a5*(1-rho));\ny=y1.*Werkweek5.*x;\nplot(x,y,'k-')\nxlabel( 'Aantal_servers ')\nylabel( 'Gemiddelde_verblijftijd_[uur] ')\ntitle( 'CT_scan ')\nsaveas(gcf, 'ctkans.png')\nCTscan=y(8)*60\nMaxServerCT=y(length(x))*60\nVerschilCT=CTscan-MaxServerCT
```

G. AANTAL NIEUWE PATIËNTEN VARIËREN

```

Werkweek1=30;%Aantal uur in werkweek longarts
Werkweek2=46.5;%labprik
Werkweek3=40;%dagbehandeling
Werkweek4=30;%verpleegkundig specialist
Werkweek5=48;%CT scan
Bedieningsduur1=15/(60*Werkweek1);%De gemiddelde bedieningsduur van de longarts
Bedieningsduur2=20/(60*Werkweek2);%van de labprik
Bedieningsduur3=45/(60*Werkweek3);%van de dagbehandeling
Bedieningsduur4=15/(60*Werkweek4);%van de verpleegkundig specialist
Bedieningsduur5=30/(60*Werkweek5);%van de CT scan
i=11; % Aantal verschillende type patiënten
nu=zeros(1,i);% Aankomstintensiteiten per type
p=zeros(1,i);% Kans op een bepaald type
max=11
x=zeros(1,max);
a1=zeros(1,max);
a2=zeros(1,max);
a3=zeros(1,max);
a4=zeros(1,max);
a5=zeros(1,max);
for i=1:11
    if i==11
        p(i)=0.8^(i-1);
    else
        p(i)=0.2*0.8^(i-1);
    end
end

end
for lambda=1:1:max
    for k=1:11
        nu(k)=p(k)*lambda;
    end
    x(lambda)=lambda;
    a1(lambda)=2*nu(1)+3*nu(2)+4*nu(3)+5*nu(4)+6*nu(5)+7*nu(6)+8*nu(7)+9*nu(8)+...
    10*nu(9)+11*nu(10)+14*nu(11);
    a2(lambda)=2*nu(1)+4*nu(2)+6*nu(3)+8*nu(4)+10*nu(5)+12*nu(6)+14*nu(7)+16*nu(8)+...
    18*nu(9)+20*nu(10)+24*nu(11);
    a3(lambda)=2*nu(1)+4*nu(2)+6*nu(3)+8*nu(4)+10*nu(5)+12*nu(6)+14*nu(7)+16*nu(8)+...
    18*nu(9)+20*nu(10)+24*nu(11);
    a4(lambda)=nu(1)+2*nu(2)+3*nu(3)+4*nu(4)+5*nu(5)+6*nu(6)+7*nu(7)+8*nu(8)+...
    9*nu(9)+10*nu(10)+13*nu(11);
    a5(lambda)=nu(1)+nu(2)+2*nu(3)+2*nu(4)+3*nu(5)+3*nu(6)+4*nu(7)+4*nu(8)+...
    5*nu(9)+5*nu(10)+5*nu(11);
end
%% longarts

```

```

mu=1/Bedieningsduur1;%mu bepalen
rho=a1./(mu)%rho bepalen
y1=rho./(a1.*(1-rho));%gemiddelde verblijftijd
y=y1.*Werkweek1;%gemiddelde verblijftijd in uren
plot(x,y,'k-')
xlabel('Aantal_nieuwe_patiënten_oproepen_per_week')
ylabel('Gemiddelde_verblijftijd_[uur]')
title('Longarts')
saveas(gcf,'longartsgev2.png')
%% lab prik
mu=1/Bedieningsduur2;
rho=a2./(mu)
y1=rho./(a2.*(1-rho));
y=y1.*Werkweek2;
plot(x,y,'k-')
xlabel('Aantal_nieuwe_patiënten_oproepen_per_week')
ylabel('Gemiddelde_verblijftijd_[uur]')
title('Priklab')
saveas(gcf,'labprikgev2.png')
%% dagbehandeling
mu=1/Bedieningsduur3;
rho=a3./(mu.*2)
y1=rho./(a3.*(1-rho));
y=y1.*Werkweek3*2;
plot(x,y,'k-')
xlabel('Aantal_nieuwe_patiënten_oproepen_per_week')
ylabel('Gemiddelde_verblijftijd_[uur]')
title('Dagbehandeling')
saveas(gcf,'dagbehandelinggev2.png')
%% Verpleegkundig specialist
mu=1/Bedieningsduur4;
rho=a4./(mu)
y1=rho./(a4.*(1-rho));
y=y1.*Werkweek4;
plot(x,y,'k-')
xlabel('Aantal_nieuwe_patiënten_oproepen_per_week')
ylabel('Gemiddelde_verblijftijd_[uur]')
title('Verpleegkundig_specialist')
saveas(gcf,'verpleegkundiggev2.png')
%% CT Scan
mu=1/Bedieningsduur5;
rho=a5./(mu)
y1=rho./(a5.*(1-rho));
y=y1.*Werkweek5;
plot(x,y,'k-')
xlabel('Aantal_nieuwe_patiënten_oproepen_per_week')
ylabel('Gemiddelde_verblijftijd_[uur]')
title('CT_scan')

```

```
saveas(gcf, 'ctgev2.png')
```

H. SIMULATIE

H.1. DE SIMULATIE

```

import random
import math
from FunctiesKans import *
NieuweAankomst=5 #Nieuwe patienten oproepen per week
week=30*60 #Aantal minuten in een werkweek
N=41 #Aantal replicaties
M=126 #Aantal weken per replicatie
KansEen=0.2#kans dat een patient van type 1 is
KansTwee=0.16#kans dat een patient van type 2 is
KansDrie=0.128#kans dat een patient van type 3 is
KansVier=0.1024#kans dat een patient van type 4 is
KansVijf=0.0819#kans dat een patient van type 5 is
KansZes=0.0655#kans dat een patient van type 6 is
KansZeven=0.0524#kans dat een patient van type 7 is
KansAcht=0.0419#kans dat een patient van type 8 is
KansNegen=0.0336#kans dat een patient van type 9 is
KansTien=0.0268#kans dat een patient van type 10 is
KansElf=0.1074#kans dat een patient van type 11 is
WachtrijLongarts=[]#hier komen de patienten die in de wachtrij voor de longarts staan
WachtrijLabprik=[]#hier komen de patienten die in de wachtrij voor het priklab staan
WachtrijDagbehandeling=[]#hier komen de patienten die in de wachtrij voor de...
#dagbehandeling staan
WachtrijCTscan=[]#hier komen de patienten die in de wachtrij voor de CT scan staan
WachtrijVerpleegkundig=[]#hier komen de patienten die in de wachtrij voor de...
#verpleegkundig specialist staan
AankomstDagbehandeling=[] #hier komen de tijdstippen van patienten die aankomen...
#bij de dagbehandeling
AankomstLabprik=[] #hier komen de tijdstippen van patienten die aankomen bij het prikla
AankomstCTscan=[] #hier komen de tijdstippen van patienten die aankomen bij de CT scan
GemVerblijftijdMetCTscan=[0 for i in range(N)] # de gemiddelde verblijftijd van een...
#behandelingsweek met een CT scan
GemVerblijftijdMetLongarts=[0 for i in range(N)]# de gemiddelde verblijftijd van een...
#behandelingsweek met controle bij de longarts
GemVerblijftijdMetVerpleegkundig=[0 for i in range(N)]# de gemiddelde verblijftijd van...
#een behandelingsweek met controle bij de verpleegkundig specialist
GemVerblijftijdLongarts=[0 for i in range(N)]# de gemiddelde verblijftijd van een...
#behandelingsweek met alleen een controle bij de longarts
GemVerblijftijdMetLongartsNieuw=[0 for i in range(N)]# de gemiddelde verblijftijd...
#van een behandelingsweek met een intake
GemDagenLongarts=[0 for i in range(N)]# gemiddeld aantal behandelingsdagen van...
#behandelweek met controle bij de longarts
GemDagenMetCTscan=[0 for i in range(N)]#gemiddeld aantal behandelingsdagen van...
#behandelweek met CT scan
GemDagenMetLongarts=[0 for i in range(N)]# gemiddeld aantal behandelingsdagen van...

```

```

#behandelweek met controle bij de longarts
GemDagenMetVerpleegkundig=[0 for i in range(N)]# gemiddeld aantal behandelingsdagen...
#van behandelweek met controle bij verpleegkundig specialist
GemDagenLongarts=[0 for i in range(N)]# gemiddeld aantal behandelingsdagenbehandelweek
#met alleen longarts
GemDagenMetLongartsNieuw=[0 for i in range(N)]#gemiddeld aantal behandelingsdagen...
#behandelweek met intake bij de longarts
VerblijftijdMetCTscan=[]#verblijftijd patient van behandelweek met CT scan
VerblijftijdMetLongarts=[]# van behandelweek met controle bij de longarts
VerblijftijdMetVerpleegkundig=[]# van behandelweek met controle bij...
#verpleegkundig specialist
VerblijftijdLongarts=[]# behandelweek met alleen longarts
VerblijftijdMetLongartsNieuw=[]#behandelweek met intake bij de longarts
DagenLongarts=[]# van behandelweek met controle bij de longarts
DagenMetCTscan=[]#verblijftijd patient van behandelweek met CT scan
DagenMetLongarts=[]# van behandelweek met controle bij de longarts
DagenMetVerpleegkundig=[]# van behandelweek met controle bij verpleegkundig specialist
DagenLongarts=[]# behandelweek met alleen longarts
DagenMetLongartsNieuw=[]#behandelweek met intake bij de longarts
GemLongartsBezet=[0 for i in range(N)]#Gemiddeld aantal minuten dat de longarts bezet..
#is in een behandelingsweek
GemLabprikBezet=[0 for i in range(N)]#Gemiddeld aantal minuten dat het priklab bezet is
#in een behandelingsweek
GemDagbehandelingBezet=[0 for i in range(N)]#Gemiddeld aantal minuten dat de...
#dagbehandeling bezet is in een behandelingsweek
GemVerpleegkundigBezet=[0 for i in range(N)]#Gemiddeld aantal minuten dat de...
#verpleegkundig specialist bezet is in een behandelingsweek
GemCTscanBezet=[0 for i in range(N)]#Gemiddeld aantal minuten dat de CT scan bezet is..
#in een behandelingsweek
gemgeslotenlab=[0 for i in range(N)]#Gemiddelde sluitingstijd van het priklab
gemgeslotendag=[0 for i in range(N)]#Gemiddelde sluitingstijd van de dagbehandeling
gemgeslotenct=[0 for i in range(N)]#Gemiddelde sluitingstijd van de CT scan
gemAankomstLab=[0 for i in range(N)]# gemiddelde aankomstintensiteit priklab
gemAankomstLong=[0 for i in range(N)]# gemiddelde aankomstintensiteit longarts
gemAankomstVerpleeg=[0 for i in range(N)]# gemiddelde aankomstintensiteit...
#verpleegkundig specialist
gemAankomstdag=[0 for i in range(N)]# gemiddelde aankomstintensiteit dagbehandeling
gemAankomstCT=[0 for i in range(N)]# gemiddelde aankomstintensiteit CT scan
for j in range(N): #replicaties
    LambdaL=[0 for i in range(69)]#Aankomstintensiteit longarts per type
    LambdaV=[0 for i in range(68)]#Aankomstintensiteit verpleegkundig...
    #specialist per type
    LambdaL[68]=NieuweAankomst#Aankomstintensiteit nieuwe patienten
    geslotenlab=[0 for i in range(M)]#sluiting priklab
    geslotendag=[0 for i in range(M)]#sluiting dagbehandeling
    geslotenct=[0 for i in range(M)]#sluiting CT scan
    LongartsBezet=[0 for i in range(M)]#tijd dat de longarts bezet is
    LabprikBezet=[0 for i in range(M)]#tijd dat het priklab bezet is

```

```

DagbehandelingBezet=[0 for i in range(M)]#tijd dat de dagbehandeling bezet is
VerpleegkundigBezet=[0 for i in range(M)]#tijd dat de verpleegkundig...
#specialist bezet is
CTscanBezet=[0 for i in range(M)]#tijd dat de CT scan bezet is
AankomstLab=[0 for i in range(M)]#aankomstintensiteit priklab
AankomstLong=[0 for i in range(M)]#aankomstintensiteit lonarts
AankomstVerpleeg=[0 for i in range(M)]#aankomstintensiteit...
#verpleegkundig specialist
Aankomstdag=[0 for i in range(M)]#aankomstintensiteit dagbehandeling
AankomstCT=[0 for i in range(M)]#aankomstintensiteit CT scan
for k in range(M):#Aantal weken
    TIME=0#iedere week begint op tijdstip 0
    AantalAankomstenLong=sum(LambdaL)#totaal aantal aankomsten longarts
    AantalAankomstenVerpleeg=sum(LambdaV)#totaal aantal aankomsten...
    #verpleegkundig specialist
    EVENT=[10e100 for i in range(10)]#alle gebeurtenissen initieel 10e100
    EVENT[0]=0#er komt meteen een patient
    if AantalAankomstenVerpleeg!=0:#wanneer er minimaal een aankomst bij...
        #de verpleegkundig specialist is
        EVENT[6]=0# de eerste patient komt meteen in het begin van de week
    else:
        EVENT[6]=10e100#wanneer er geen aankomsten bij de verpleegkundig...
        #specialist zijn, komt deze gebeurtenis niet voor
    Server=[0 for i in range(5)] #Server[i]=0 server i vrij,
    #Server[i]=1 server bezet
    TypeInBediening=[0 for i in range(5)]#hier worden de type patienten...
    #opgeslagen die in bediening zijn bij de verschillende stations
    nNul=0 #Aantal patienten in wachtrij longarts
    nEen=0 #Aantal patienten in wachtrij priklab
    nTwee=0 #Aantal patienten in wachtrij dagbehandeling
    nDrie=0#Aantal patienten in wachtrij verpleegkundig specialist
    nVier=0#Aantal patienten in wachtrij CT scan
    TypeOnderweg=[0 for i in range(5)]#hier worden de type patienten opgeslagen...
    #die onderweg zijn naar een station en als eerste bij dat station aankomen
    TypeOnderwegEen=[]# alle patienten die onderweg zijn naar het priklab
    TypeOnderwegTwee=[]#alle patienten die onderweg zijn naar de dagbehandeling
    TypeOnderwegDrie=[]#alle patienten die onderweg zijn naar CT scan
    NuL=[0 for i in range(69)] #Patienten die terug komen bij de longarts
    NuV=[0 for i in range(68)] # bij de verpleegkundig specialist
    if k==26: #verwijder eerste jaar. De warm up period
        for i in range(26):
            LongartsBezet[i]=0
            LabprikBezet[i]=0
            VerpleegkundigBezet[i]=0
            DagbehandelingBezet[i]=0
            CTscanBezet[i]=0
            VerblijftijdMetCTscan=[]
            VerblijftijdMetLongarts=[]

```

```

VerblijftijdMetVerpleegkundig=[]
VerblijftijdLongarts=[]
VerblijftijdMetLongartsNieuw=[]
AankomstLab[i]=0
AankomstLong[i]=0
AankomstVerpleeg[i]=0
Aankomstdag[i]=0
AankomstCT[i]=0
geslotenlab[i]=0
geslotendag[i]=0
geslotenct[i]=0
TIME=min(EVENT)#plan de eerste gebeurtenis
while TIME<10e100:#zolang er nog patienten in het systeem zitten
    if TIME<9*60:#maandag
        dag=1
    elif TIME<19*60:#dinsdag
        dag=2
    elif TIME<29*60:#woensdag
        dag=3
    elif TIME<39*60:#donderdag
        dag=4
    elif TIME<47.5*60:#vrijdag
        dag=5
    else:
        dag=6#zaterdag
    if TIME==EVENT[0]: #Aankomst longarts
        AankomstLong[k]=AankomstLong[k]+1#er is 1 patient meer aangekomen...
        #bij de longarts
        for i in range(69):
            if max(LambdaL)==LambdaL[i]:#bepaal het type van deze patient
                Index=i
                break
        LambdaL[Index]=LambdaL[Index]-1#van dit type patienten hoeft er nu...
        #nog maar een minder te komen
        if Index==68: #patient is nieuw
            Type=Typebepalen()#bepaal het type
            Type=Type+[TIME]#voeg de aankomsttijd toe
            Type=Type+[dag]#voeg de dag toe
            Bediening=det(1/45)#intake
        else:
            Bediening=det(1/15)#controle
            Type=IndexTypeL(Index)#het type volgt uit de index
            Type=Type+[TIME]#voeg de tijd toe
            Type=Type+[dag]#voeg de dag toe
        if sum(LambdaL)==0:#wanneer er geen patienten meer hoeven te komen...
            #bij de longarts
            EVENT[0]=10e100#er is geen nieuwe aankomst meer
    else:

```

```

EVENT[0]=TIME+det(AantalAankomstenLong/week)#de volgende aankomst
if EVENT[0]>=6*60 and TIME<10*60:#deze aankomst kan alleen gepland.
    #worden op tijdstippen dat de longarts werkt
    EVENT[0]=EVENT[0]+4*60
if EVENT[0]>=16*60 and TIME<20*60:
    EVENT[0]=EVENT[0]+4*60
if EVENT[0]>=26*60 and TIME<30*60:
    EVENT[0]=EVENT[0]+4*60
if EVENT[0]>=36*60 and TIME<40*60:
    EVENT[0]=EVENT[0]+4*60
if EVENT[0]>=46*60:#einde van de werkweek
    EVENT[0]=10e100 #station gesloten
if Server[0]==0: #server is leeg
    Server[0]=1 #patient gaat in bediening, dus server bezet
    LongartsBezet[k]=LongartsBezet[k]+Bediening#de tijd dat de longarts
    #bezet is wordt verhoogd met de bedieningsduur
    EVENT[1]=TIME+Bediening#na de bediening vertrekt de patient bij de.
    #longarts
    TypeInBediening[0]=Type#sla op welke patient in bediening is
else:#server is bezet
    WachtrijLongarts.insert(nNul,Type)#patient sluit achteraan...
    #in de wachtrij
    nNul=nNul+1#het aantal patienten in de wachtrij wordt...
    #verhoogd met 1
elif TIME==EVENT[1]: # vertrek longarts
    Type=TypeInBediening[0]#het type patient dat gaat vertrekken
    TypeInBediening[0]=0#nu geen patient in bediening
    verlaten=Systeemverlaten(Type)#kijk of het behandelingsproces erop zit.
    #voor deze patient
    if verlaten==1:#einde van het behandelingsproces
        VerblijftijdLongarts.insert(0,TIME-Type[2])#sla de duur van deze...
        #behandelingsweek op
        DagenLongarts.insert(0,dag-Type[3]+1)#sla het aantal dagen dat de..
        #patient over de behandelingsweek gedaan heeft op
    else:#de patient blijft in het systeem
        Type=[Type[0],Type[1]+1,Type[2],Type[3]]#patient is een fase verder
        if len(AankomstLabprik)==0:#wanneer er niet meer patienten...
            #onderweg zijn naar het priklab
            EVENT[2]=TIME#de volgende aankomst bij het priklab
            TypeOnderweg[1]=Type#sla het type op
        else:#er zijn meer patienten onderweg naar het priklab
            AankomstLabprik.insert(0,TIME)#sla de aankomsttijd op
            TypeOnderwegEen.insert(0,Type)#sla ook het type op
            EVENT[2]=min(AankomstLabprik)#de volgende aankomst bij het...
            #priklab is degene die als eerste plaatsvindt
            for i in range(len(AankomstLabprik)):
                if min(AankomstLabprik)==AankomstLabprik(i):#bepaal welk...
                    #type er bij de eerste aankomst hoort

```

```

        IndexMin=i
        break
        TypeOnderweg[1]=TypeOnderwegEen[IndexMin]
        del AankomstLabprik[IndexMin]
        #verwijder de aankomsttijd van deze patient uit de lijst
        del TypeOnderwegEen[IndexMin]
        #verwijder het type van deze patient uit de lijst
    if nNul!=0:
        #wanneer er iemand in de wachtrij staat
        Type=WachtrijLongarts[0]
        #gaat de eerst volgende patient in bediening
        del WachtrijLongarts[0]
        #deze patient verlaat de wachtrij
        nNul=nNul-1
        #het aantal patienten in de wachtrij is nu 1 lager
        if Type[0]==1:
            #wanneer de patient die in bediening gaat nieuw is
            Bediening=det(1/45)#krijgt deze patient een intake
        else:#wanneer het geen nieuwe patient is
            Bediening=det(1/15)#krijgt deze patient een controle
        LongartsBezet[k]=LongartsBezet[k]+Bediening
        # de tijd dat de longarts bezet is , wordt verhoogd
        EVENT[1]=TIME+Bediening
        #na deze bedieningsduur verlaat de patient de longarts
        TypeInBediening[0]=Type
        #sla het type dat in bediening gaat op
    else:#wanneer er niemand in de wachtrij staat
        EVENT[1]=10e100#kan er ook niemand in bediening gaan
        Server[0]=0#dus is de server leeg
elif TIME==EVENT[2]: #aankomst labprik
    AankomstLab[k]=AankomstLab[k]+1
    #er is 1 patient meer aangekomen bij het priklab
    if Server[1]==0:#wanneer de server leeg was
        Server[1]=1#is de server nu bezet
        Bediening=det(1/20)#bepaal de bedieningsduur
        LabprikBezet[k]=LabprikBezet[k]+Bediening
        # de tijd dat het priklab bezet is , wordt verhoogd
        EVENT[3]=TIME+Bediening
        #na deze bedieningsduur verlaat de patient het priklab
        TypeInBediening[1]=TypeOnderweg[1]
        #sla het type dat in bediening gaat op
    else:#wanneer de server bezet is
        WachtrijLabprik.insert(nEen,TypeOnderweg[1])
        #sluit de patient achteraan in de wachtrij
        nEen=nEen+1#er staat nu 1 patient meer in de wachtrij
if len(TypeOnderwegEen)==0:
    #wanneer er geen patienten meer onderweg zijn naar het priklab
    TypeOnderweg[1]=0

```

```

EVENT[2]=10e100#kan er geen volgende aankomst gepland worden
else:#wanneer er nog wel patienten onderweg zijn naar het priklab
TypeOnderweg[1]=TypeOnderwegEen[len(TypeOnderwegEen)-1]
#kan de aankomst van de eerstvolgende patient gepland worden
del TypeOnderwegEen[len(TypeOnderwegEen)-1]
#verwijder deze patient uit de lijst meerder patienten onderweg
EVENT[2]=AankomstLabprik[len(AankomstLabprik)-1]
#de volgende aankomst is de tijd dat deze patient aankomt
del AankomstLabprik[len(AankomstLabprik)-1]
#verwijder deze tijd uit meerdere patienten onderweg
elif TIME==EVENT[3]: #vertrek labprik
Type=TypeInBediening[1]#het type patient dat gaat vertrekken
TypeInBediening[1]=0#nu geen patient in bediening
Type=[Type[0],Type[1]+1,Type[2],Type[3]]#de patient is een fase verder
geslotenlab[k]=TIME#deze tijd wordt telkens opnieuw opgeslagen.
#De laatste tijd is de sluitingstijd van het priklab
if len(AankomstDagbehandeling)==0:
#wanneer er niet meer patienten onderweg zijn
EVENT[4]=TIME
#dan is de volgende aankomst bij de dagbehandeling deze patient
#de dagbehandeling moet wel open zijn
if EVENT[4]>=7.5*60 and EVENT[4]<10:
EVENT[4]=EVENT[4]+2.5*60
if EVENT[4]>=17.5*60 and EVENT[4]<20:
EVENT[4]=EVENT[4]+2.5*60
if EVENT[4]>=27.5*60 and EVENT[4]<30:
EVENT[4]=EVENT[4]+2.5*60
if EVENT[4]>=37.5*60 and EVENT[4]<40:
EVENT[4]=EVENT[4]+2.5*60
TypeOnderweg[2]=Type#sla het type op
else:#er zijn meer patienten onderweg naar de dagbehandeling
TIME2=TIME
#bepaal de aankomsttijd van deze patient op
#ook in dit geval moet de dagbehandeling wel open zijn
if TIME>=7.5*60 and TIME<10:
TIME2=TIME+2.5*60
if TIME>=17.5*60 and TIME<20:
TIME2=TIME+2.5*60
if TIME>=27.5*60 and TIME<30:
TIME2=TIME+2.5*60
if TIME>=37.5*60 and TIME<40:
TIME2=TIME+2.5*60
AankomstDagbehandeling.insert(0,TIME2)#sla de aankomsttijd op
TypeOnderwegTwee.insert(0,Type)#sla ook het type van de patient op
EVENT[4]=AankomstDagbehandeling[len(AankomstDagbehandeling)-1]
#de volgende aankomst bij de dagbehandeling is degene die als...
#eerste plaatsvindt
TypeOnderweg[2]=TypeOnderwegTwee[len(TypeOnderwegTwee)-1]

```

```

#bepaal welk type er bij de eerste aankomst hoort
del AankomstDagbehandeling[len(AankomstDagbehandeling)-1]
#verwijder de aankomsttijd van deze patient uit de lijst
del TypeOnderwegTwee[len(TypeOnderwegTwee)-1]
#verwijder het type van deze patient uit de lijst
if nEen!=0:#wanneer er iemand in de wachtrij staat
    Type=WachtrijLabprik[0]
    #gaat de eerst volgende patient in bediening
    del WachtrijLabprik[0]
    #deze patient verlaat de wachtrij
    nEen=nEen-1
    #het aantal patienten in de wachtrij is nu met 1 verminderd
    Bediening=det(1/20)#bepaal de bedieningsduur
    LabprikBezet[k]=LabprikBezet[k]+Bediening
    #de tijd dat het priklab bezet is, wordt verhoogd
    EVENT[3]=TIME+Bediening
    #na deze bedieningsduur verlaat de patient het priklab
    TypeInBediening[1]=Type
    #sla het type dat in bediening gaat op
else:#wanneer er niemand in de wachtrij staat
    EVENT[3]=10e100#kan er ook niemand in bediening gaan
    Server[1]=0#dus is de server leeg
elif TIME==EVENT[4]: #Aankomst dagbehandeling
    Aankomstdag[k]=Aankomstdag[k]+1
    #er is 1 patient meer aangekomen bij de dagbehandeling
    if Server[2]==0:#wanneer de server leeg was
        Server[2]=1#is de server nu bezet
        Bediening=det(1/45)#bepaal de bedieningsduur
        DagbehandelingBezet[k]=DagbehandelingBezet[k]+Bediening
        #de tijd dat de dagbehandeling bezet is, wordt verhoogd
        EVENT[5]=TIME+Bediening
        #na deze bedieningsduur verlaat de patient de dagbehandeling
        TypeInBediening[2]=TypeOnderweg[2]
        #sla het type dat in bediening gaat op
    else:#wanneer de server bezet is
        WachtrijDagbehandeling.insert(nTwee,TypeOnderweg[2])
        #sluit de patient achteraan in de wachtrij
        nTwee=nTwee+1#er staat nu 1 patient meer in de wachtrij
    if len(TypeOnderwegTwee)==0:
        #wanneer er geen patienten meer onderweg zijn
        TypeOnderweg[2]=0
        EVENT[4]=10e100#kan er geen volgende aankomst gepland worden
    else:#wanneer er nog patienten onderweg zijn naar de dagbehandeling
        TypeOnderweg[2]=TypeOnderwegTwee[len(TypeOnderwegTwee)-1]
        #kan de aankomst van de eerstvolgende patient gepland worden
        del TypeOnderwegTwee[len(TypeOnderwegTwee)-1]
        #verwijder deze patient uit de lijst meerder patienten onderweg
        EVENT[4]=AankomstLabprik[len(AankomstLabprik)-1]

```

```

#de volgende aankomst is de tijd dat deze patient aankomt
del AankomstLabprik[len(AankomstLabprik)-1]
#verwijder deze tijd uit meerdere patienten onderweg
elif TIME==EVENT[5]: #vertrek dagbehandeling
    geslotendag[k]=TIME#deze tijd wordt telkens opnieuw opgeslagen.
    #De laatste tijd is de sluitingstijd van de dagbehandeling
    Type=TypeInBediening[2]#het type patient dat gaat vertrekken
    TypeInBediening[2]=0#nu geen patient in bediening
    Type=[Type[0],Type[1]+1,Type[2],Type[3]]#de patient is een fase verder
    Fase=Fasebepalen(Type)#bepaal de volgende fase
    if Fase==1: #longarts
        Index=TypeIndexL(Type)#bepaal de bijbehorende index
        NuL[Index]=NuL[Index]+1#het aantal patienten van dit type...
        #dat volgende behandelingsweek terug komen wordt met 1 verhoogd
        VerblijftijdMetVerpleegkundig.insert(0,TIME-Type[2])
        #sla de verblijftijd van deze patient op
        DagenMetVerpleegkundig.insert(0,dag-Type[3]+1)
        #sla het aantal dagen dat de patient hierover heeft gedaan op
    elif Fase==4: #verpleegkundig specialist
        Index=TypeIndexV(Type)#bepaal de bijbehorende index
        NuV[Index]=NuV[Index]+1#het aantal patienten van dit type...
        #dat volgende behandelingsweek terug komen wordt met 1 verhoogd
        Verblijftijd=TIME-Type[2]#bepaal de verblijftijd van deze patient
        if Type[1]==4:#wanneer de patient een intake heeft gehad
            VerblijftijdMetLongartsNieuw.insert(0,Verblijftijd)
            #sla de verblijftijd van deze patient op
            DagenMetLongartsNieuw.insert(0,dag-Type[3]+1)
            #sla het aantal dagen dat de patient hierover heeft gedaan op
        else:#wanneer de patient een controle heeft gehad
            VerblijftijdMetLongarts.insert(0,Verblijftijd)
            #sla de verblijftijd van deze patient op
            DagenMetLongarts.insert(0,dag-Type[3]+1)
            #sla het aantal dagen dat de patient hierover heeft gedaan op
    elif Fase==5: #CT scan
        if len(AankomstCTscan)==0:
            #wanneer er niet meer patienten onderweg zijn naar de CT scan
            EVENT[8]=TIME
            #dan is de volgende aankomst bij de CT scan deze patient
            #de CT scan moet dan wel open zijn
            if EVENT[8]>=7.5*60 and EVENT[8]<10:
                EVENT[8]=EVENT[8]+2.5*60
            if EVENT[8]>=17.5*60 and EVENT[8]<20:
                EVENT[8]=EVENT[8]+2.5*60
            if EVENT[8]>=27.5*60 and EVENT[8]<30:
                EVENT[8]=EVENT[8]+2.5*60
            if EVENT[8]>=37.5*60 and EVENT[8]<40:
                EVENT[8]=EVENT[8]+2.5*60
            TypeOnderweg[3]=Type#sla het type op

```

```

else:#er zijn meer patienten onderweg naar de CT scan
    TIME3=TIME#bepaal de aankomsttijd van deze patient op
    if TIME>=7.5*60 and TIME<10:#de CT scan moet dan open zijn
        TIME3=TIME+2.5*60
    if TIME>=17.5*60 and TIME<20:
        TIME3=TIME+2.5*60
    if TIME>=27.5*60 and TIME<30:
        TIME3=TIME+2.5*60
    if TIME>=37.5*60 and TIME<40:
        TIME3=TIME+2.5*60
    AankomstCTscan.insert(0,TIME3)#sla de aankomsttijd op
    TypeOnderwegDrie.insert(0,Type)#sla ook het type op
    EVENT[8]=AankomstCTscan[len(AankomstCTscan)-1]
    #de volgende aankomst bij de CT scan
    TypeOnderweg[3]=TypeOnderwegDrie[len(TypeOnderwegDrie)-1]
    #bepaal welk type er bij de eerste aankomst hoort
    del AankomstCTscan[len(AankomstCTscan)-1]
    #verwijder de aankomsttijd van deze patient uit de lijst
    del TypeOnderwegDrie[len(TypeOnderwegDrie)-1]
    #verwijder het type van deze patient uit de lijst
if nTwee!=0:#wanneer er iemand in de wachtrij staat
    Type=WachtrijDagbehandeling[0]
    #gaat de eerst volgende patient in bediening
    del WachtrijDagbehandeling[0]
    #deze patient verlaat de wachtrij
    nTwee=nTwee-1
    #het aantal patienten in de wachtrij is met 1 verminderd
    Bediening=det(1/45)#bepaal de bedieningsduur
    DagbehandelingBezet[k]=DagbehandelingBezet[k]+Bediening
    #de tijd dat het priklab bezet is, wordt verhoogd
    EVENT[5]=TIME+Bediening
    #na deze bedieningsduur vertrekt de patient
    TypeInBediening[2]=Type
    #sla het type dat in bediening gaat op
else:#wanneer er niemand in de wachtrij staat
    EVENT[5]=10e100#kan er ook niemand in bediening gaan
    Server[2]=0#dus is de server leeg
elif TIME==EVENT[6]: #aankomst verpleegkundig specialist
    AankomstVerpleeg[k]=AankomstVerpleeg[k]+1
    #er is 1 patient meer aangekomen
    for i in range(68):
        if max(LambdaV)==LambdaV[i]:
            #bepaal het type van deze patient
            Index=i
            break
    LambdaV[i]=LambdaV[i]-1
    #van dit type patienten hoeft er nu nog maar een minder te komen
    Type=IndexTypeV(Index)#het type volgt uit de index

```

```

Type=Type+[TIME]#voeg de tijd toe
Type=Type+[dag]#voeg de dag toe
if sum(LambdaV)==0:
    #wanneer er geen patienten meer hoeven te komen bij
    EVENT[6]=10e100#er is geen nieuwe aankomst meer
else:
    EVENT[6]=TIME+det(AantalAankomstenVerpleeg/week)
    #plan de volgende aankomst
    #de verpleegkundig specialist moet wel open zijn
    if EVENT[6]>=6*60 and TIME<10*60:
        EVENT[6]=EVENT[6]+4*60
    if EVENT[6]>=16*60 and TIME<20*60:
        EVENT[6]=EVENT[6]+4*60
    if EVENT[6]>=26*60 and TIME<30*60:
        EVENT[6]=EVENT[6]+4*60
    if EVENT[6]>=36*60 and TIME<40*60:
        EVENT[6]=EVENT[6]+4*60
    if EVENT[6]>=46*60:
        EVENT[6]=10e100 #station gesloten
if Server[3]==0: #server is leeg
    Server[3]=1
    #patient gaat in bediening, dus server bezet
    Bediening=det(1/15)#bepaal de bedieningsduur
    VerpleegkundigBezet[k]=VerpleegkundigBezet[k]+Bediening
    #de tijd dat de verpleegkundig specialist bezet is wordt verhoogd
    EVENT[7]=TIME+Bediening
    #na de bediening vertrekt de patient bij de...
    #verpleegkundig specialist
    TypeInBediening[3]=Type#sla op welke patient in bediening is
else:#server is bezet
    WachtrijVerpleegkundig.insert(nDrie,Type)
    #patient sluit achteraan in de wachtrij
    nDrie=nDrie+1
    #het aantal patienten in de wachtrij wordt verhoogd
elif TIME==EVENT[7]: #vertrek verpleegkundig specialist
    Type=TypeInBediening[3]#het type patient dat gaat vertrekken
    TypeInBediening[3]=0#nu geen patient in bediening
    Type=[Type[0],Type[1]+1,Type[2],Type[3]]
    #de patient is nu een fase verder
    if len(AankomstLabprik)==0:
        #wanneer er niet meer patienten onderweg zijn naar het priklab
        EVENT[2]=TIME
        #dan is de volgende aankomst bij het priklab deze patient
        TypeOnderweg[1]=Type#sla het type op
    else:#er zijn meer patienten onderweg naar het priklab
        AankomstLabprik.insert(0,TIME)
        #sla de aankomsttijd van deze patient op
        TypeOnderwegEen.insert(0,Type)

```

```

#sla ook het type van deze patient op
EVENT[2]=min(AankomstLabprik)
#de volgende aankomst bij het prik lab
for i in range(len(AankomstLabprik)):
    if min(AankomstLabprik)==AankomstLabprik(i):
        #bepaal welk type er bij de eerste aankomst hoort
        IndexMin=i
        break
TypeOnderweg[1]=TypeOnderwegEen[IndexMin]
del AankomstLabprik[IndexMin]
#verwijder de aankomsttijd van deze patient uit de lijst
del TypeOnderwegEen[IndexMin]
#verwijder het type van deze patient uit de lijst
if nDrie!=0:#wanneer er iemand in de wachtrij staat
    Type=WachtrijVerpleegkundig[0]
    #gaat de eerst volgende patient in bediening
    del WachtrijVerpleegkundig[0]
    #deze patient verlaat de wachtrij
    nDrie=nDrie-1
    #het aantal patienten in de wachtrij is nu 1 lager
    Bediening=det(1/15)#bepaal de bedieningsduur
    VerpleegkundigBezet[k]=VerpleegkundigBezet[k]+Bediening
    #de tijd dat de verpleegkundig specialist bezet is ,
    #wordt verhoogd met deze bedieningsduur
    EVENT[7]=TIME+Bediening
    #na deze bedieningsduur vertrekt de patient
    TypeInBediening[3]=Type
    #sla het type dat in bediening gaat op
else:#wanneer er niemand in de wachtrij staat
    EVENT[7]=10e100#kan er ook niemand in bediening gaan
    Server[3]=0#dus is de server leeg
elif TIME==EVENT[8]:#Aankomst CT scan
    AankomstCT[k]=AankomstCT[k]+1
    #er is 1 patient meer aangekomen bij de CT scan
    if Server[4]==0:#wanneer de server leeg was
        Server[4]=1#is de server nu bezet
        Bediening=det(1/30)#bepaal de bedieningsduur
        CTscanBezet[k]=CTscanBezet[k]+Bediening
        #de tijd dat de CT scan bezet is , wordt verhoogd
        EVENT[9]=TIME+Bediening#de patient verlaat de CT scan
        TypeInBediening[4]=TypeOnderweg[3]#sla het type op
    else:#wanneer de server bezet is
        WachtrijCTscan.insert(nVier,TypeOnderweg[3])#sluit de patient ...
        #achteraan in de wachtrij
        nVier=nVier+1#er staat nu 1 patient meer in de wachtrij
if len(TypeOnderwegDrie)==0:#hier stond TypeOnderwegVier
    TypeOnderweg[4]=0
    EVENT[8]=10e100#kan er geen volgende aankomst gepland worden

```

```

else:#wanneer er nog wel patienten onderweg zijn naar de dagbehandeling
    TypeOnderweg[4]=TypeOnderwegDrie[len(TypeOnderwegDrie)-1]
    #de aankomst van de eerstvolgende patient plannen
    del TypeOnderwegDrie[len(TypeOnderwegDrie)-1]
    #verwijder deze patient uit de lijst
    EVENT[8]=AankomstCTscan[len(AankomstCTscan)-1]
    #de volgende aankomst is de tijd dat deze patient bij
    del AankomstCTscan[len(AankomstCTscan)-1]#verwijder deze tijd uit..
    #meerdere patienten onderweg
elif TIME==EVENT[9]: #Vertrek CTscan
    geslotenct[k]=TIME#deze tijd wordt telkens opnieuw opgeslagen.
    #De laatste tijd is de sluitingstijd van de CT scan
    Type=TypeInBediening[4]#het type patient dat gaat vertrekken
    TypeInBediening[4]=0#nu geen patient in bediening
    Type=[Type[0],Type[1]+1,Type[2],Type[3]]#de patient is een fase verder
    Index=TypeIndexL(Type)#bepaald de index die bij dit type hoort
    NuL[Index]=NuL[Index]+1#het aantal patienten dat volgende...
    #behandelingsweek aankomt van dit type wordt met 1 verhoogd
    VerblijftijdMetCTscan.insert(0,TIME-Type[2])#sla de verblijftijd op
    DagenMetCTscan.insert(0,dag-Type[3]+1)#ook het aantal dagen dat de...
    #patient erover gedaan heeft
    if nVier!=0:#wanneer er iemand in de wachtrij staat
        Type=WachtrijCTscan[0]#gaat de eerst volgende patient in bediening
        del WachtrijCTscan[0]#deze patient verlaat de wachtrij
        nVier=nVier-1#het aantal patienten in de wachtrij is nu 1 lager
        Bediening=det(1/30)#bepaal de bedieningsduur
        CTscanBezet[k]=CTscanBezet[k]+Bediening# de tijd dat de CT scan...
        bezet is, wordt verhoogd met deze bedieningsduur
        EVENT[9]=TIME+Bediening#na deze bedieningsduur verlaat de patient..
        #de CT scan
        TypeInBediening[4]=Type#sla het type dat in bediening gaat op
    else:#wanneer er niemand in de wachtrij staat
        EVENT[9]=10e100#kan er ook niemand in bediening gaan
        Server[4]=0#dus is de server leeg
    TIME=min(EVENT)#bepaal de eerstvolgende gebeurtenis
LambdaL=NuL
#aantal patienten die volgende behandelingsweek opgeroepen worden bij longarts
LambdaL[68]=NieuweAankomst#nieuwe patienten
LambdaV=NuV
#aantal patienten die volgende behandelingsweek opgeroepen worden bij..
#de verpleegkundig specialist
gemgeslotenlab[j]=sum(geslotenlab)/(M-26)
#bepaal de gemiddelde sluitingstijd van het priklab
gemgeslotendag[j]=sum(geslotendag)/(M-26)
#bepaal de gemiddelde sluitingstijd van de dagbehandeling
gemgeslotenct[j]=sum(geslotenct)/(M-26)
#bepaal de gemiddelde sluitingstijd van de CT scan
GemVerblijftijdMetCTscan[j]=sum(VerblijftijdMetCTscan)/len(VerblijftijdMetCTscan)

```

```

#bepaal de gemiddelde verblijftijd van een behandelingsweek met CT scan
GemVerblijftijdMetLongarts[j]=sum(VerblijftijdMetLongarts)/len(VerblijftijdMetLongarts)
#bepaal de gemiddelde verblijftijd van een behandelingsweek met controle bij de longarts
GemVerblijftijdMetVerpleegkundig[j]=sum(VerblijftijdMetVerpleegkundig)/len(VerblijftijdMetVerpleegkundig)
#bepaal de gemiddelde verblijftijd van een behandelingsweek met controle bij de verpleegkundig specialist
GemVerblijftijdLongarts[j]=sum(VerblijftijdLongarts)/len(VerblijftijdLongarts)
#bepaal de gemiddelde verblijftijd van een behandelingsweek met alleen een controle bij de longarts
GemVerblijftijdMetLongartsNieuw[j]=sum(VerblijftijdMetLongartsNieuw)/len(VerblijftijdMetLongartsNieuw)
#bepaal de gemiddelde verblijftijd van een behandelingsweek met een intake bij de longarts
GemDagenMetCTscan[j]=sum(DagenMetCTscan)/len(DagenMetCTscan)
#bepaal het gemiddeld aantal dagen van een behandelingsweek met CT scan
GemDagenMetLongarts[j]=sum(DagenMetLongarts)/len(DagenMetLongarts)
#bepaal het gemiddeld aantal dagen van een behandelingsweek met controle bij de longarts
GemDagenMetVerpleegkundig[j]=sum(DagenMetVerpleegkundig)/len(DagenMetVerpleegkundig)
#bepaal het gemiddeld aantal dagen van een behandelingsweek met controle bij de verpleegkundig specialist
GemDagenLongarts[j]=sum(DagenLongarts)/len(DagenLongarts)
#bepaal het gemiddeld aantal dagen van een behandelingsweek met alleen een controle bij de longarts
GemDagenMetLongartsNieuw[j]=sum(DagenMetLongartsNieuw)/len(DagenMetLongartsNieuw)
#bepaal het gemiddeld aantal dagen van een behandelingsweek met een intake bij de longarts
GemLongartsBezetting[j]=sum(LongartsBezetting)/(M-26)
#bepaal de gemiddelde bezetting van de longarts
GemLabprikBezetting[j]=sum(LabprikBezetting)/(M-26)
#bepaal de gemiddelde bezetting van het priklab
GemDagbehandelingBezetting[j]=sum(DagbehandelingBezetting)/(M-26)
#bepaal de gemiddelde bezetting van de dagbehandeling
GemVerpleegkundigBezetting[j]=sum(VerpleegkundigBezetting)/(M-26)
#bepaal de gemiddelde bezetting van de verpleegkundig specialist
GemCTscanBezetting[j]=sum(CTscanBezetting)/(M-26)
#bepaal de gemiddelde bezetting van de CT scan
gemAankomstLab[j]=sum(AankomstLab)/(M-26)
#bepaal de gemiddelde aankomstintensiteit bij het priklab
gemAankomstLong[j]=sum(AankomstLong)/(M-26)
#bepaal de gemiddelde aankomstintensiteit bij de longarts
gemAankomstVerpleeg[j]=sum(AankomstVerpleeg)/(M-26)
#bepaal de gemiddelde aankomstintensiteit bij de verpleegkundig specialist
gemAankomstdag[j]=sum(Aankomstdag)/(M-26)
#bepaal de gemiddelde aankomstintensiteit bij de dagbehandeling
gemAankomstCT[j]=sum(AankomstCT)/(M-26)
#bepaal de gemiddelde aankomstintensiteit bij de CT scan
LongartsB=sum(GemLongartsBezetting)/len(GemLongartsBezetting)
#betrouwbaarheidsinterval voor de gemiddelde bezetting van de longarts
LB=[0 for i in range(N)]
for i in range(N):
    LB[i]=(GemLongartsBezetting[i]-LongartsB)**2
Slb=sum(LB)/(N-1)
BTIlb=[0,0]
BTIlb[0]=LongartsB-2.021*(math.sqrt(Slb/N))
BTIlb[1]=LongartsB+2.021*(math.sqrt(Slb/N))

```

```
print('Bezetting_Longarts', BTilb)
LabprikB=sum(GemLabprikBezet)/len(GemLabprikBezet)
#betrouwbaarheidsinterval voor de gemiddelde bezetting van het priklab
LaB=[0 for i in range(N)]
for i in range(N):
    LaB[i]=(GemLabprikBezet[i]-LabprikB)**2
Slab=sum(LaB)/(N-1)
BTilab=[0,0]
BTilab[0]=LabprikB-2.021*(math.sqrt(Slab/N))
BTilab[1]=LabprikB+2.021*(math.sqrt(Slab/N))
print('Bezetting_Labprik', BTilab)
DagbehandelingB=sum(GemDagbehandelingBezet)/len(GemDagbehandelingBezet)
#betrouwbaarheidsinterval voor de gemiddelde bezetting van de dagbehandeling
DB=[0 for i in range(N)]
for i in range(N):
    DB[i]=(GemDagbehandelingBezet[i]-DagbehandelingB)**2
Sdb=sum(DB)/(N-1)
BTIdb=[0,0]
BTIdb[0]=DagbehandelingB-2.021*(math.sqrt(Sdb/N))
BTIdb[1]=DagbehandelingB+2.021*(math.sqrt(Sdb/N))
print('Bezetting_Dagebehandeling', BTIdb)
VerpleegkundigB=sum(GemVerpleegkundigBezet)/len(GemVerpleegkundigBezet)
#betrouwbaarheidsinterval voor de gemiddelde bezetting van de verpleegkundig specialist
VB=[0 for i in range(N)]
for i in range(N):
    VB[i]=(GemVerpleegkundigBezet[i]-VerpleegkundigB)**2
Svb=sum(VB)/(N-1)
BTIvb=[0,0]
BTIvb[0]=VerpleegkundigB-2.021*(math.sqrt(Svb/N))
BTIvb[1]=VerpleegkundigB+2.021*(math.sqrt(Svb/N))
print('Bezetting_Verpleegkundig_specialist', BTIvb)
CTscanB=sum(GemCTscanBezet)/len(GemCTscanBezet)
#betrouwbaarheidsinterval voor de gemiddelde bezetting van de CT scan
CTB=[0 for i in range(N)]
for i in range(N):
    CTB[i]=(GemCTscanBezet[i]-CTscanB)**2
Sctb=sum(CTB)/(N-1)
BTIctb=[0,0]
BTIctb[0]=CTscanB-2.021*(math.sqrt(Sctb/N))
BTIctb[1]=CTscanB+2.021*(math.sqrt(Sctb/N))
print('Bezetting_CTscan', BTIctb)
Gem1=sum(GemVerblijftijdMetCTscan)/len(GemVerblijftijdMetCTscan)
#betrouwbaarheidsinterval voor de gemiddelde verblijftijd van een behandelingsweek met.
#CT scan
ctscan=[0 for i in range(N)]
for i in range(N):
    ctscan[i]=(GemVerblijftijdMetCTscan[i]-Gem1)**2
Sct=sum(ctscan)/(N-1) #steekproefvariantie
```

```

BTIctscan=[0,0]
BTIctscan[0]=Gem1-2.021*(math.sqrt(Sct/N))
BTIctscan[1]=Gem1+2.021*(math.sqrt(Sct/N))
Gem2=sum(GemVerblijftijdMetLongarts)/len(GemVerblijftijdMetLongarts)
#betrouwbaarheidsinterval voor de gemiddelde verblijftijd van een behandelingsweek met.
#een controle bij de longarts
metlongarts=[0 for i in range(N)]
for i in range(N):
    metlongarts[i]=(GemVerblijftijdMetLongarts[i]-Gem2)**2
Smetlong=sum(metlongarts)/(N-1)
BTImetlong=[0,0]
BTImetlong[0]=Gem2-2.021*(math.sqrt(Smetlong/N))
BTImetlong[1]=Gem2+2.021*(math.sqrt(Smetlong/N))
Gem3=sum(GemVerblijftijdMetVerpleegkundig)/len(GemVerblijftijdMetVerpleegkundig)
#betrouwbaarheidsinterval voor de gemiddelde verblijftijd van een behandelingsweek met.
#een controle bij de verpleegkundig specialist
metverpleeg=[0 for i in range(N)]
for i in range(N):
    metverpleeg[i]=(GemVerblijftijdMetVerpleegkundig[i]-Gem3)**2
Smetverpleeg=sum(metverpleeg)/(N-1)
BTImetverpleeg=[0,0]
BTImetverpleeg[0]=Gem3-2.021*(math.sqrt(Smetverpleeg/N))
BTImetverpleeg[1]=Gem3+2.021*(math.sqrt(Smetverpleeg/N))
Gem4=sum(GemVerblijftijdLongarts)/len(GemVerblijftijdLongarts)
#betrouwbaarheidsinterval voor de gemiddelde verblijftijd van een behandelingsweek met.
#alleen een controle bij de longarts
long=[0 for i in range(N)]
for i in range(N):
    long[i]=(GemVerblijftijdLongarts[i]-Gem4)**2
Slong=sum(long)/(N-1)
BTIlong=[0,0]
BTIlong[0]=Gem4-2.021*(math.sqrt(Slong/N))
BTIlong[1]=Gem4+2.021*(math.sqrt(Slong/N))
Gem5=sum(GemVerblijftijdMetLongartsNieuw)/len(GemVerblijftijdMetLongartsNieuw)
#betrouwbaarheidsinterval voor de gemiddelde verblijftijd van een behandelingsweek met.
#een intake
longnieuw=[0 for i in range(N)]
for i in range(N):
    longnieuw[i]=(GemVerblijftijdMetLongartsNieuw[i]-Gem5)**2
Slongnieuw=sum(longnieuw)/(N-1)
BTIlongnieuw=[0,0]
BTIlongnieuw[0]=Gem5-2.021*(math.sqrt(Slongnieuw/N))
BTIlongnieuw[1]=Gem5+2.021*(math.sqrt(Slongnieuw/N))
Gemdag=sum(GemDagenMetCTscan)/len(GemDagenMetCTscan)
#betrouwbaarheidsinterval voor de gemiddeld aantal dagen van een behandelingsweek met..
#CT scan
ctscandag=[0 for i in range(N)]
for i in range(N):

```

```

    ctscandag[i]=(GemDagenMetCTscan[i]-Gemdag)**2
Sctdag=sum(ctscandag)/(N-1) #steekproefvariantie
BTIctscandag=[0,0]
BTIctscandag[0]=Gemdag-2.021*(math.sqrt(Sctdag/N))
BTIctscandag[1]=Gemdag+2.021*(math.sqrt(Sctdag/N))
Gemdag2=sum(GemDagenMetLongarts)/len(GemDagenMetLongarts)
#betrouwbaarheidsinterval voor de gemiddeld aantal dagen van een behandelingsweek met..
#een controle bij de longarts
metlongartsdag=[0 for i in range(N)]
for i in range(N):
    metlongartsdag[i]=(GemDagenMetLongarts[i]-Gemdag2)**2
Smetlongdag=sum(metlongartsdag)/(N-1)
BTImetlongdag=[0,0]
BTImetlongdag[0]=Gemdag2-2.021*(math.sqrt(Smetlongdag/N))
BTImetlongdag[1]=Gemdag2+2.021*(math.sqrt(Smetlongdag/N))
Gemdag3=sum(GemDagenMetVerpleegkundig)/len(GemDagenMetVerpleegkundig)
#betrouwbaarheidsinterval voor de gemiddeld aantal dagen van een...
#behandelingsweek met een controle bij de verpleegkundig specialist
metverpleegdag=[0 for i in range(N)]
for i in range(N):
    metverpleegdag[i]=(GemDagenMetVerpleegkundig[i]-Gemdag3)**2
Smetverpleegdag=sum(metverpleegdag)/(N-1)
BTImetverpleegdag=[0,0]
BTImetverpleegdag[0]=Gemdag3-2.021*(math.sqrt(Smetverpleegdag/N))
BTImetverpleegdag[1]=Gemdag3+2.021*(math.sqrt(Smetverpleegdag/N))
Gemdag4=sum(GemDagenLongarts)/len(GemDagenLongarts)
#betrouwbaarheidsinterval voor de gemiddeld aantal dagen van een...
#behandelingsweek met alleen een controle bij de longarts
longdag=[0 for i in range(N)]
for i in range(N):
    longdag[i]=(GemDagenLongarts[i]-Gemdag4)**2
Slongdag=sum(longdag)/(N-1)
BTIlongdag=[0,0]
BTIlongdag[0]=Gemdag4-2.021*(math.sqrt(Slongdag/N))
BTIlongdag[1]=Gemdag4+2.021*(math.sqrt(Slongdag/N))
Gemdag5=sum(GemDagenMetLongartsNieuw)/len(GemDagenMetLongartsNieuw)
#betrouwbaarheidsinterval voor de gemiddeld aantal dagen...
#van een behandelingsweek met een intake
longnieuwdag=[0 for i in range(N)]
for i in range(N):
    longnieuwdag[i]=(GemDagenMetLongartsNieuw[i]-Gemdag5)**2
Slongnieuwdag=sum(longnieuwdag)/(N-1)
BTIlongnieuwdag=[0,0]
BTIlongnieuwdag[0]=Gemdag5-2.021*(math.sqrt(Slongnieuwdag/N))
BTIlongnieuwdag[1]=Gemdag5+2.021*(math.sqrt(Slongnieuwdag/N))
Gem6=sum(gemgeslotenlab)/len(gemgeslotenlab)
#betrouwbaarheidsinterval van de gemiddelde sluitingstijd van het priklab
gesloten1=[0 for i in range(N)]

```

```

for i in range(N):
    gesloten1[i]=(gemgeslotenlab[i]-Gem6)**2
Sgesloten1=sum(gesloten1)/(N-1)
BTIgesloten1=[0,0]
BTIgesloten1[0]=Gem6-2.021*(math.sqrt(Sgesloten1/N))
BTIgesloten1[1]=Gem6+2.021*(math.sqrt(Sgesloten1/N))
Gem7=sum(gemgeslotendag)/len(gemgeslotendag)
#betrouwbaarheidsinterval van de gemiddelde sluitingstijd van de dagbehandeling
gesloten2=[0 for i in range(N)]
for i in range(N):
    gesloten2[i]=(gemgeslotendag[i]-Gem7)**2
Sgesloten2=sum(gesloten2)/(N-1)
BTIgesloten2=[0,0]
BTIgesloten2[0]=Gem7-2.021*(math.sqrt(Sgesloten2/N))
BTIgesloten2[1]=Gem7+2.021*(math.sqrt(Sgesloten2/N))
Gem8=sum(gemgeslotenct)/len(gemgeslotenct)
#betrouwbaarheidsinterval van de gemiddelde sluitingstijd van de CT scan
gesloten3=[0 for i in range(N)]
for i in range(N):
    gesloten3[i]=(gemgeslotenct[i]-Gem8)**2
Sgesloten3=sum(gesloten3)/(N-1)
BTIgesloten3=[0,0]
BTIgesloten3[0]=Gem8-2.021*(math.sqrt(Sgesloten3/N))
BTIgesloten3[1]=Gem8+2.021*(math.sqrt(Sgesloten3/N))

Gem9=sum(gemAankomstLab)/len(gemAankomstLab)
#betrouwbaarheidsinterval van de gemiddelde aankomstintensiteit bij het priklab
aankomst1=[0 for i in range(N)]
for i in range(N):
    aankomst1[i]=(gemAankomstLab[i]-Gem9)**2
Saankomst1=sum(aankomst1)/(N-1)
BTIaankomst1=[0,0]
BTIaankomst1[0]=Gem9-2.021*(math.sqrt(Saankomst1/N))
BTIaankomst1[1]=Gem9+2.021*(math.sqrt(Saankomst1/N))
Gem10=sum(gemAankomstLong)/len(gemAankomstLong)
#betrouwbaarheidsinterval van de gemiddelde aankomstintensiteit bij de longarts
aankomst2=[0 for i in range(N)]
for i in range(N):
    aankomst2[i]=(gemAankomstLong[i]-Gem10)**2
Saankomst2=sum(aankomst2)/(N-1)
BTIaankomst2=[0,0]
BTIaankomst2[0]=Gem10-2.021*(math.sqrt(Saankomst2/N))
BTIaankomst2[1]=Gem10+2.021*(math.sqrt(Saankomst2/N))
Gem11=sum(gemAankomstVerpleeg)/len(gemAankomstVerpleeg)
#betrouwbaarheidsinterval van de gemiddelde aankomstintensiteit ...
#bij de verpleegkundig specialist
aankomst3=[0 for i in range(N)]

```

```

for i in range(N):
    aankomst3[i]=(gemAankomstVerpleeg[i]-Gem11)**2
Saankomst3=sum(aankomst3)/(N-1)
BTlaankomst3=[0,0]
BTlaankomst3[0]=Gem11-2.021*(math.sqrt(Saankomst3/N))
BTlaankomst3[1]=Gem11+2.021*(math.sqrt(Saankomst3/N))
Gem12=sum(gemAankomstdag)/len(gemAankomstdag)
#betrouwbaarheidsinterval van de gemiddelde aankomstintensiteit bij de dagbehandeling
aankomst4=[0 for i in range(N)]
for i in range(N):
    aankomst4[i]=(gemAankomstdag[i]-Gem12)**2
Saankomst4=sum(aankomst4)/(N-1)
BTlaankomst4=[0,0]
BTlaankomst4[0]=Gem12-2.021*(math.sqrt(Saankomst4/N))
BTlaankomst4[1]=Gem12+2.021*(math.sqrt(Saankomst4/N))
Gem13=sum(gemAankomstCT)/len(gemAankomstCT)
#betrouwbaarheidsinterval van de gemiddelde aankomstintensiteit bij de CT scan
aankomst5=[0 for i in range(N)]
for i in range(N):
    aankomst5[i]=(gemAankomstCT[i]-Gem13)**2
Saankomst5=sum(aankomst5)/(N-1)
BTlaankomst5=[0,0]
BTlaankomst5[0]=Gem13-2.021*(math.sqrt(Saankomst5/N))
BTlaankomst5[1]=Gem13+2.021*(math.sqrt(Saankomst5/N))
print('Verblijftijd_met_CTscan:',BTIctscan)
print('Verblijftijd_met_Longarts:',BTImetlong)
print('Verblijftijd_met_Verpleegkundige',BTImetverpleeg)
print('Verblijftijd_Longarts',BTIlong)
print('Verblijftijd_Longarts_Nieuw',BTIlongnieuw)
print('Dagen_met_CTscan:',BTIctscandag)
print('Dagen_met_Longarts:',BTImetlongdag)
print('Dagen_met_Verpleegkundige',BTImetverpleegdag)
print('Dagen_Longarts',BTIlongdag)
print('Dagen_Longarts_Nieuw',BTIlongnieuwdag)
print('sluitingstijd_Lab',BTIgesloten1)
print('sluitingstijd_Dagbehandeling',BTIgesloten2)
print('sluitingstijd_CT_scan',BTIgesloten3)
print('aankomstintensiteit_longarts',BTlaankomst2)
print('aankomstintensiteit_priklab',BTlaankomst1)
print('aankomstintensiteit_dagbehandeling',BTlaankomst4)
print('aankomstintensiteit_verpleegkundig_specialist',BTlaankomst3)
print('aankomstintensiteit_CT_scan',BTlaankomst5)

```

H.2. DE GEBRUIKTE FUNCTIES

```

import random
import math
def exp(mu): #trekken uit exponentiele verdeling

```

```
u=random.random()
A=(-math.log(1-u))/mu
return A

def det(mu): #trekken uit deterministische verdeling
    A=1/(mu)
    return A

def pois(mu): #trekken uit poisson verdeling
    u=random.random()
    n=0
    F=0
    while u>F:
        F=F+(math.e**(-mu)*(mu**n)/(math.factorial(n)))
        n=n+1
    return n-1

def ber(mu): #trekken uit bernouilli verdeling
    u=random.random()
    F=1-mu
    if u>F:
        return 1
    else:
        return 0

def TypeIndexL(Type):
    if [Type[0],Type[1]]==[1,8]:
        Index=0
    elif [Type[0],Type[1]]==[2,7]:
        Index=1
    elif [Type[0],Type[1]]==[2,14]:
        Index=2
    elif [Type[0],Type[1]]==[3,7]:
        Index=3
    elif [Type[0],Type[1]]==[3,14]:
        Index=4
    elif [Type[0],Type[1]]==[3,21]:
        Index=5
    elif [Type[0],Type[1]]==[4,7]:
        Index=6
    elif [Type[0],Type[1]]==[4,14]:
        Index=7
    elif [Type[0],Type[1]]==[4,20]:
        Index=8
    elif [Type[0],Type[1]]==[4,27]:
        Index=9
```

```
elif [Type[0],Type[1]]==[5,7]:
    Index=10
elif [Type[0],Type[1]]==[5,14]:
    Index=11
elif [Type[0],Type[1]]==[5,20]:
    Index=12
elif [Type[0],Type[1]]==[5,27]:
    Index=13
elif [Type[0],Type[1]]==[5,34]:
    Index=14
elif [Type[0],Type[1]]==[6,7]:
    Index=15
elif [Type[0],Type[1]]==[6,14]:
    Index=16
elif [Type[0],Type[1]]==[6,20]:
    Index=17
elif [Type[0],Type[1]]==[6,27]:
    Index=18
elif [Type[0],Type[1]]==[6,33]:
    Index=19
elif [Type[0],Type[1]]==[6,40]:
    Index=20
elif [Type[0],Type[1]]==[7,7]:
    Index=21
elif [Type[0],Type[1]]==[7,14]:
    Index=22
elif [Type[0],Type[1]]==[7,20]:
    Index=23
elif [Type[0],Type[1]]==[7,27]:
    Index=24
elif [Type[0],Type[1]]==[7,33]:
    Index=25
elif [Type[0],Type[1]]==[7,40]:
    Index=26
elif [Type[0],Type[1]]==[7,47]:
    Index=27
elif [Type[0],Type[1]]==[8,7]:
    Index=28
elif [Type[0],Type[1]]==[8,14]:
    Index=29
elif [Type[0],Type[1]]==[8,20]:
    Index=30
elif [Type[0],Type[1]]==[8,27]:
    Index=31
elif [Type[0],Type[1]]==[8,33]:
    Index=32
elif [Type[0],Type[1]]==[8,40]:
    Index=33
```

```
elif [Type[0],Type[1]]==[8,46]:
    Index=34
elif [Type[0],Type[1]]==[8,53]:
    Index=35
elif [Type[0],Type[1]]==[9,7]:
    Index=36
elif [Type[0],Type[1]]==[9,14]:
    Index=37
elif [Type[0],Type[1]]==[9,20]:
    Index=38
elif [Type[0],Type[1]]==[9,27]:
    Index=39
elif [Type[0],Type[1]]==[9,33]:
    Index=40
elif [Type[0],Type[1]]==[9,40]:
    Index=41
elif [Type[0],Type[1]]==[9,46]:
    Index=42
elif [Type[0],Type[1]]==[9,53]:
    Index=43
elif [Type[0],Type[1]]==[9,60]:
    Index=44
elif [Type[0],Type[1]]==[10,7]:
    Index=45
elif [Type[0],Type[1]]==[10,14]:
    Index=46
elif [Type[0],Type[1]]==[10,20]:
    Index=47
elif [Type[0],Type[1]]==[10,27]:
    Index=48
elif [Type[0],Type[1]]==[10,33]:
    Index=49
elif [Type[0],Type[1]]==[10,40]:
    Index=50
elif [Type[0],Type[1]]==[10,46]:
    Index=51
elif [Type[0],Type[1]]==[10,53]:
    Index=52
elif [Type[0],Type[1]]==[10,59]:
    Index=53
elif [Type[0],Type[1]]==[10,66]:
    Index=54
elif [Type[0],Type[1]]==[11,7]:
    Index=55
elif [Type[0],Type[1]]==[11,14]:
    Index=56
elif [Type[0],Type[1]]==[11,20]:
    Index=57
```

```
elif [Type[0],Type[1]]==[11,27]:
    Index=58
elif [Type[0],Type[1]]==[11,33]:
    Index=59
elif [Type[0],Type[1]]==[11,40]:
    Index=60
elif [Type[0],Type[1]]==[11,46]:
    Index=61
elif [Type[0],Type[1]]==[11,53]:
    Index=62
elif [Type[0],Type[1]]==[11,59]:
    Index=63
elif [Type[0],Type[1]]==[11,66]:
    Index=64
elif [Type[0],Type[1]]==[11,72]:
    Index=65
elif [Type[0],Type[1]]==[11,78]:
    Index=66
elif [Type[0],Type[1]]==[11,84]:
    Index=67
elif [Type[0],Type[1]]==[0,0]: #nieuw
    Index=68
    print('Nieuw')
elif [Type[0],Type[1]]==[1,1]:
    Index=69
elif [Type[0],Type[1]]==[2,1]:
    Index=70
elif [Type[0],Type[1]]==[3,1]:
    Index=71
elif [Type[0],Type[1]]==[4,1]:
    Index=72
elif [Type[0],Type[1]]==[5,1]:
    Index=73
elif [Type[0],Type[1]]==[6,1]:
    Index=74
elif [Type[0],Type[1]]==[7,1]:
    Index=75
elif [Type[0],Type[1]]==[8,1]:
    Index=76
elif [Type[0],Type[1]]==[9,1]:
    Index=77
elif [Type[0],Type[1]]==[10,1]:
    Index=78
elif [Type[0],Type[1]]==[11,1]:
    Index=79
else:
    print('Geen_Index_L')
return Index
```

```
def IndexTypeL (Index):  
    if Index==0:  
        Type=[1,8]  
    elif Index==1:  
        Type=[2,7]  
    elif Index==2:  
        Type=[2,14]  
    elif Index==3:  
        Type=[3,7]  
    elif Index==4:  
        Type=[3,14]  
    elif Index==5:  
        Type=[3,21]  
    elif Index==6:  
        Type=[4,7]  
    elif Index==7:  
        Type=[4,14]  
    elif Index==8:  
        Type=[4,20]  
    elif Index==9:  
        Type=[4,27]  
    elif Index==10:  
        Type=[5,7]  
    elif Index==11:  
        Type=[5,14]  
    elif Index==12:  
        Type=[5,20]  
    elif Index==13:  
        Type=[5,27]  
    elif Index==14:  
        Type=[5,34]  
    elif Index==15:  
        Type=[6,7]  
    elif Index==16:  
        Type=[6,14]  
    elif Index==17:  
        Type=[6,20]  
    elif Index==18:  
        Type=[6,27]  
    elif Index==19:  
        Type=[6,33]  
    elif Index==20:  
        Type=[6,40]  
    elif Index==21:  
        Type=[7,7]  
    elif Index==22:
```

```

        Type=[7,14]
    elif Index==23:
        Type=[7,20]
    elif Index==24:
        Type=[7,27]
    elif Index==25:
        Type=[7,33]
    elif Index==26:
        Type=[7,40]
    elif Index==27:
        Type=[7,47]
    elif Index==28:
        Type=[8,7]
    elif Index==29:
        Type=[8,14]
    elif Index==30:
        Type=[8,20]
    elif Index==31:
        Type=[8,27]
    elif Index==32:
        Type=[8,33]
    elif Index==33:
        Type=[8,40]
    elif Index==34:
        Type=[8,46]
    elif Index==35:
        Type=[8,53]
    elif Index==36:
        Type=[9,7]
    elif Index==37:
        Type=[9,14]
    elif Index==38:
        Type=[9,20]
    elif Index==39:
        Type=[9,27]
    elif Index==40:
        Type=[9,33]
    elif Index==41:
        Type=[9,40]
    elif Index==42:
        Type=[9,46]
    elif Index==43:
        Type=[9,53]
    elif Index==44:
        Type=[9,60]
    elif Index==45:
        Type=[10,7]
    elif Index==46:

```

```
        Type=[10,14]
    elif Index==47:
        Type=[10,20]
    elif Index==48:
        Type=[10,27]
    elif Index==49:
        Type=[10,33]
    elif Index==50:
        Type=[10,40]
    elif Index==51:
        Type=[10,46]
    elif Index==52:
        Type=[10,53]
    elif Index==53:
        Type=[10,59]
    elif Index==54:
        Type=[10,66]
    elif Index==55:
        Type=[11,7]
    elif Index==56:
        Type=[11,14]
    elif Index==57:
        Type=[11,20]
    elif Index==58:
        Type=[11,27]
    elif Index==59:
        Type=[11,33]
    elif Index==60:
        Type=[11,40]
    elif Index==61:
        Type=[11,46]
    elif Index==62:
        Type=[11,53]
    elif Index==63:
        Type=[11,59]
    elif Index==64:
        Type=[11,66]
    elif Index==65:
        Type=[11,72]
    elif Index==66:
        Type=[11,78]
    elif Index==67:
        Type=[11,84]
    elif Index==68:
        Type=[0,0] #Nieuw
    elif Index==69:
        Type=[1,1]
    elif Index==70:
```

```
        Type=[2,1]
    elif Index==71:
        Type=[3,1]
    elif Index==72:
        Type=[4,1]
    elif Index==73:
        Type=[5,1]
    elif Index==74:
        Type=[6,1]
    elif Index==75:
        Type=[7,1]
    elif Index==76:
        Type=[8,1]
    elif Index==77:
        Type=[9,1]
    elif Index==78:
        Type=[10,1]
    elif Index==79:
        Type=[11,1]
    else:
        print('Geen_Type_L')
    return Type

def TypeIndexV(Type):
    if [Type[0],Type[1]]==[1,4]:
        Index=0
    elif [Type[0],Type[1]]==[2,4]:
        Index=1
    elif [Type[0],Type[1]]==[2,10]:
        Index=2
    elif [Type[0],Type[1]]==[3,4]:
        Index=3
    elif [Type[0],Type[1]]==[3,10]:
        Index=4
    elif [Type[0],Type[1]]==[3,17]:
        Index=5
    elif [Type[0],Type[1]]==[4,4]:
        Index=6
    elif [Type[0],Type[1]]==[4,10]:
        Index=7
    elif [Type[0],Type[1]]==[4,17]:
        Index=8
    elif [Type[0],Type[1]]==[4,23]:
        Index=9
    elif [Type[0],Type[1]]==[5,4]:
        Index=10
    elif [Type[0],Type[1]]==[5,10]:
        Index=11
```

```
elif [Type[0],Type[1]]==[5,17]:
    Index=12
elif [Type[0],Type[1]]==[5,23]:
    Index=13
elif [Type[0],Type[1]]==[5,30]:
    Index=14
elif [Type[0],Type[1]]==[6,4]:
    Index=15
elif [Type[0],Type[1]]==[6,10]:
    Index=16
elif [Type[0],Type[1]]==[6,17]:
    Index=17
elif [Type[0],Type[1]]==[6,23]:
    Index=18
elif [Type[0],Type[1]]==[6,30]:
    Index=19
elif [Type[0],Type[1]]==[6,36]:
    Index=20
elif [Type[0],Type[1]]==[7,4]:
    Index=21
elif [Type[0],Type[1]]==[7,10]:
    Index=22
elif [Type[0],Type[1]]==[7,17]:
    Index=23
elif [Type[0],Type[1]]==[7,23]:
    Index=24
elif [Type[0],Type[1]]==[7,30]:
    Index=25
elif [Type[0],Type[1]]==[7,36]:
    Index=26
elif [Type[0],Type[1]]==[7,43]:
    Index=27
elif [Type[0],Type[1]]==[8,4]:
    Index=28
elif [Type[0],Type[1]]==[8,10]:
    Index=29
elif [Type[0],Type[1]]==[8,17]:
    Index=30
elif [Type[0],Type[1]]==[8,23]:
    Index=31
elif [Type[0],Type[1]]==[8,30]:
    Index=32
elif [Type[0],Type[1]]==[8,36]:
    Index=33
elif [Type[0],Type[1]]==[8,43]:
    Index=34
elif [Type[0],Type[1]]==[8,49]:
    Index=35
```

```
elif [Type[0],Type[1]]==[9,4]:
    Index=36
elif [Type[0],Type[1]]==[9,10]:
    Index=37
elif [Type[0],Type[1]]==[9,17]:
    Index=38
elif [Type[0],Type[1]]==[9,23]:
    Index=39
elif [Type[0],Type[1]]==[9,30]:
    Index=40
elif [Type[0],Type[1]]==[9,36]:
    Index=41
elif [Type[0],Type[1]]==[9,43]:
    Index=42
elif [Type[0],Type[1]]==[9,49]:
    Index=43
elif [Type[0],Type[1]]==[9,56]:
    Index=44
elif [Type[0],Type[1]]==[10,4]:
    Index=45
elif [Type[0],Type[1]]==[10,10]:
    Index=46
elif [Type[0],Type[1]]==[10,17]:
    Index=47
elif [Type[0],Type[1]]==[10,23]:
    Index=48
elif [Type[0],Type[1]]==[10,30]:
    Index=49
elif [Type[0],Type[1]]==[10,36]:
    Index=50
elif [Type[0],Type[1]]==[10,43]:
    Index=51
elif [Type[0],Type[1]]==[10,49]:
    Index=52
elif [Type[0],Type[1]]==[10,56]:
    Index=53
elif [Type[0],Type[1]]==[10,62]:
    Index=54
elif [Type[0],Type[1]]==[11,4]:
    Index=55
elif [Type[0],Type[1]]==[11,10]:
    Index=56
elif [Type[0],Type[1]]==[11,17]:
    Index=57
elif [Type[0],Type[1]]==[11,23]:
    Index=58
elif [Type[0],Type[1]]==[11,30]:
    Index=59
```

```
elif [Type[0],Type[1]]==[11,36]:
    Index=60
elif [Type[0],Type[1]]==[11,43]:
    Index=61
elif [Type[0],Type[1]]==[11,49]:
    Index=62
elif [Type[0],Type[1]]==[11,56]:
    Index=63
elif [Type[0],Type[1]]==[11,62]:
    Index=64
elif [Type[0],Type[1]]==[11,69]:
    Index=65
elif [Type[0],Type[1]]==[11,75]:
    Index=66
elif [Type[0],Type[1]]==[11,81]:
    Index=67
else:
    print('Geen_index_V')
return Index

def IndexTypeV(Index):
    if Index==0:
        Type=[1,4]
    elif Index==1:
        Type=[2,4]
    elif Index==2:
        Type=[2,10]
    elif Index==3:
        Type=[3,4]
    elif Index==4:
        Type=[3,10]
    elif Index==5:
        Type=[3,17]
    elif Index==6:
        Type=[4,4]
    elif Index==7:
        Type=[4,10]
    elif Index==8:
        Type=[4,17]
    elif Index==9:
        Type=[4,23]
    elif Index==10:
        Type=[5,4]
    elif Index==11:
        Type=[5,10]
    elif Index==12:
        Type=[5,17]
```

```
elif Index==13:
    Type=[5,23]
elif Index==14:
    Type=[5,30]
elif Index==15:
    Type=[6,4]
elif Index==16:
    Type=[6,10]
elif Index==17:
    Type=[6,17]
elif Index==18:
    Type=[6,23]
elif Index==19:
    Type=[6,30]
elif Index==20:
    Type=[6,36]
elif Index==21:
    Type=[7,4]
elif Index==22:
    Type=[7,10]
elif Index==23:
    Type=[7,17]
elif Index==24:
    Type=[7,23]
elif Index==25:
    Type=[7,30]
elif Index==26:
    Type=[7,36]
elif Index==27:
    Type=[7,43]
elif Index==28:
    Type=[8,4]
elif Index==29:
    Type=[8,10]
elif Index==30:
    Type=[8,17]
elif Index==31:
    Type=[8,23]
elif Index==32:
    Type=[8,30]
elif Index==33:
    Type=[8,36]
elif Index==34:
    Type=[8,43]
elif Index==35:
    Type=[8,49]
elif Index==36:
    Type=[9,4]
```

```

elif Index==37:
    Type=[9,10]
elif Index==38:
    Type=[9,17]
elif Index==39:
    Type=[9,23]
elif Index==40:
    Type=[9,30]
elif Index==41:
    Type=[9,36]
elif Index==42:
    Type=[9,43]
elif Index==43:
    Type=[9,49]
elif Index==44:
    Type=[9,56]
elif Index==45:
    Type=[10,4]
elif Index==46:
    Type=[10,10]
elif Index==47:
    Type=[10,17]
elif Index==48:
    Type=[10,23]
elif Index==49:
    Type=[10,30]
elif Index==50:
    Type=[10,36]
elif Index==51:
    Type=[10,43]
elif Index==52:
    Type=[10,49]
elif Index==53:
    Type=[10,56]
elif Index==54:
    Type=[10,62]
elif Index==55:
    Type=[11,4]
elif Index==56:
    Type=[11,10]
elif Index==57:
    Type=[11,17]
elif Index==58:
    Type=[11,23]
elif Index==59:
    Type=[11,30]
elif Index==60:
    Type=[11,36]

```

```
elif Index==61:
    Type=[11,43]
elif Index==62:
    Type=[11,49]
elif Index==63:
    Type=[11,56]
elif Index==64:
    Type=[11,62]
elif Index==65:
    Type=[11,69]
elif Index==66:
    Type=[11,75]
elif Index==67:
    Type=[11,81]
else:
    print('Geen_Type_V')
return Type

def Systeemverlaten(Type):
    FaseTypeEen=[1,2,3,4,2,3,5,1] #1: Longarts , 2: Labprik , 3: Dagbehandeling
    FaseTypeTwee=[1,2,3,4,2,3,1,2,3,4,2,3,5,1] #4: Verpleegkundig specialist
    FaseTypeDrie=FaseTypeTwee+[2,3,4,2,3,5,1] #5: CT scan
    FaseTypeVier=FaseTypeTwee+[2,3,4,2,3,1,2,3,4,2,3,5,1]
    FaseTypeVijf=FaseTypeVier+[2,3,4,2,3,5,1]
    FaseTypeZes=FaseTypeVier+[2,3,4,2,3,1,2,3,4,2,3,5,1]
    FaseTypeZeven=FaseTypeZes+[2,3,4,2,3,5,1]
    FaseTypeAcht=FaseTypeZes+[2,3,4,2,3,1,2,3,4,2,3,5,1]
    FaseTypeNegen=FaseTypeAcht+[2,3,4,2,3,5,1]
    FaseTypeTien=FaseTypeAcht+[2,3,4,2,3,1,2,3,4,2,3,5,1]
    FaseTypeElf=FaseTypeTien+[2,3,4,2,3,1,2,3,4,2,3,1,2,3,4,2,3,1]
    if Type[0]==1:
        if Type[1]==len(FaseTypeEen):
            verlaten=1 #verlaat systeem
        else:
            verlaten=0 #blijft in systeem
    elif Type[0]==2:
        if Type[1]==len(FaseTypeTwee):
            verlaten=1
        else:
            verlaten=0
    elif Type[0]==3:
        if Type[1]==len(FaseTypeDrie):
            verlaten=1
        else:
            verlaten=0
    elif Type[0]==4:
        if Type[1]==len(FaseTypeVier):
            verlaten=1
```

```

        else:
            verlaten=0
    elif Type[0]==5:
        if Type[1]==len(FaseTypeVijf):
            verlaten=1
        else:
            verlaten=0
    elif Type[0]==6:
        if Type[1]==len(FaseTypeZes):
            verlaten=1
        else:
            verlaten=0
    elif Type[0]==7:
        if Type[1]==len(FaseTypeZeven):
            verlaten=1
        else:
            verlaten=0
    elif Type[0]==8:
        if Type[1]==len(FaseTypeAcht):
            verlaten=1
        else:
            verlaten=0
    elif Type[0]==9:
        if Type[1]==len(FaseTypeNegen):
            verlaten=1
        else:
            verlaten=0
    elif Type[0]==10:
        if Type[1]==len(FaseTypeTien):
            verlaten=1
        else:
            verlaten=0
    elif Type[0]==11:
        if Type[1]==len(FaseTypeElf):
            verlaten=1
        else:
            verlaten=0
    return verlaten

def Typebepalen():
    KansEen=0.2
    KansTwee=0.16
    KansDrie=0.128
    KansVier=0.1024
    KansVijf=0.0819
    KansZes=0.0655
    KansZeven=0.0524
    KansAcht=0.0419

```

```

KansNegen=0.0336
KansTien=0.0268
KansElf=0.1074
if ber(KansEen)==1:
    Type=[1,1]
elif ber(KansTwee/(1-KansEen))==1:
    Type=[2,1]
elif ber(KansDrie/(1-(KansEen+KansTwee)))==1:
    Type=[3,1]
elif ber(KansVier/(1-(KansEen+KansTwee+KansDrie)))==1:
    Type=[4,1]
elif ber(KansVijf/(1-(KansEen+KansTwee+KansDrie+KansVier)))==1:
    Type=[5,1]
elif ber(KansZes/(1-(KansEen+KansTwee+KansDrie+KansVier+KansVijf)))==1:
    Type=[6,1]
elif ber(KansZeven/(1-(KansEen+KansTwee+KansDrie+KansVier+KansVijf+KansZes)))==1:
    Type=[7,1]
elif ber(KansAcht/(1-(KansEen+KansTwee+KansDrie+KansVier+KansVijf+KansZes+KansZeven)))==1:
    Type=[8,1]
elif ber(KansNegen/(1-(KansEen+KansTwee+KansDrie+KansVier+KansVijf+KansZes+KansZeven)))==1:
    Type=[9,1]
elif ber(KansTien/(1-(KansEen+KansTwee+KansDrie+KansVier+KansVijf+KansZes+KansZeven)))==1:
    Type=[10,1]
else:
    Type=[11,1]
return Type

def Fasebepalen(Type):
    FaseTypeEen=[1,2,3,4,2,3,5,1] #1: Longarts , 2: Labprik , 3: Dagbehandeling
    FaseTypeTwee=[1,2,3,4,2,3,1,2,3,4,2,3,5,1] #4: Verpleegkundig specialist
    FaseTypeDrie=FaseTypeTwee+[2,3,4,2,3,5,1] #5: CT scan
    FaseTypeVier=FaseTypeTwee+[2,3,4,2,3,1,2,3,4,2,3,5,1]
    FaseTypeVijf=FaseTypeVier+[2,3,4,2,3,5,1]
    FaseTypeZes=FaseTypeVier+[2,3,4,2,3,1,2,3,4,2,3,5,1]
    FaseTypeZeven=FaseTypeZes+[2,3,4,2,3,5,1]
    FaseTypeAcht=FaseTypeZes+[2,3,4,2,3,1,2,3,4,2,3,5,1]
    FaseTypeNegen=FaseTypeAcht+[2,3,4,2,3,5,1]
    FaseTypeTien=FaseTypeAcht+[2,3,4,2,3,1,2,3,4,2,3,5,1]
    FaseTypeElf=FaseTypeTien+[2,3,4,2,3,1,2,3,4,2,3,1,2,3,4,2,3,1]
    if Type[0]==1:
        Fase=FaseTypeEen[Type[1]-1]
    elif Type[0]==2:
        Fase=FaseTypeTwee[Type[1]-1]
    elif Type[0]==3:
        Fase=FaseTypeDrie[Type[1]-1]
    elif Type[0]==4:
        Fase=FaseTypeVier[Type[1]-1]
    elif Type[0]==5:

```

```
    Fase=FaseTypeVijf[Type[1]-1]
elif Type[0]==6:
    Fase=FaseTypeZes[Type[1]-1]
elif Type[0]==7:
    Fase=FaseTypeZeven[Type[1]-1]
elif Type[0]==8:
    Fase=FaseTypeAcht[Type[1]-1]
elif Type[0]==9:
    Fase=FaseTypeNegen[Type[1]-1]
elif Type[0]==10:
    Fase=FaseTypeTien[Type[1]-1]
elif Type[0]==11:
    Fase=FaseTypeElf[Type[1]-1]
return Fase
```

I. MODELEERPROCES

Deze bijlage kan onderdelen bevatten met dezelfde informatie als in de hoofdttekst of andere bijlagen.

- Om inzicht te krijgen in het behandelingsproces heb ik eerst een afspraak gemaakt met Prof. Dr. Paul Baas, een longarts van het Nederlands kanker instituut in het Antoni van Leeuwenhoekziekenhuis. In het behandelingsproces komen de patiënten iedere twee weken naar het ziekenhuis. De ene keer komen ze op controle bij de longarts en de andere keer bij de verpleegkundig specialist. Dus iedere vier weken komen ze bij de longarts en de andere keren bij de verpleegkundig specialist. Iedere behandelweek begint met een controle, vervolgens gaat een patiënt naar de dagbehandeling voor het infuus. Daarna gaat de patiënt naar het priklab, zodat de toestand van de patiënt bekeken kan worden. Vijf keer per jaar gaat de patiënt ook nog naar de CT-scan, zodat er gekeken kan worden of de behandeling resultaten boekt. Wanneer een patiënt nieuw is, gaat deze patiënt naar de longarts voor een intake van gemiddeld 45 minuten. Een gewone controle bij de longarts of verpleegkundig specialist duurt gemiddeld 15 minuten. Bij de dagbehandeling is er gemiddeld 45 minuten een bed nodig, het priklab duurt gemiddeld 20 minuten en de CT-scan 30 minuten. De strategie van het AVL om patiënten toe te laten voor de behandeling is als volgt, iedere week worden er vijf nieuwe patiënten toegelaten. Dus in 2016 worden er 260 patiënten toegelaten voor de behandeling met Nivolumab. Nivolumab wordt gebruikt voor tweedelijs patiënten met niet kleincellige longkanker, dit zijn patiënten die voor de tweede keer niet kleincellige longkanker hebben. Jaarlijks zijn er ongeveer 12000 patiënten die voor de tweede keer niet kleincellige longkanker hebben. Niet iedere patiënt

wil of kan behandeld worden met Nivolumab. Het behandelingsproces is natuurlijk heel intensief en niet iedere patiënt heeft hier behoefte aan. Voor veel patiënten is deze behandeling wel wenselijk, omdat het de verwachte levensduur verhoogd en veel minder patiënten ervaren bijwerkingen dan bij chemotherapie. Bij een behandeling met Nivolumab heeft 1 op de 14 patiënten last van bijwerkingen terwijl bij chemotherapie 1 op de 10 patiënten geen last heeft van bijwerkingen. Er zijn ongeveer 2000 tot 3000 patiënten per jaar beschikbaar voor de behandeling. In Nederland zijn er 13 centra die deze behandeling aanbieden, waarvan het Antoni van Leeuwenhoekziekenhuis er dus een is. Volgens het Antoni van Leeuwenhoekziekenhuis is het knelpunt van deze behandeling de afspraken met de dagbehandeling

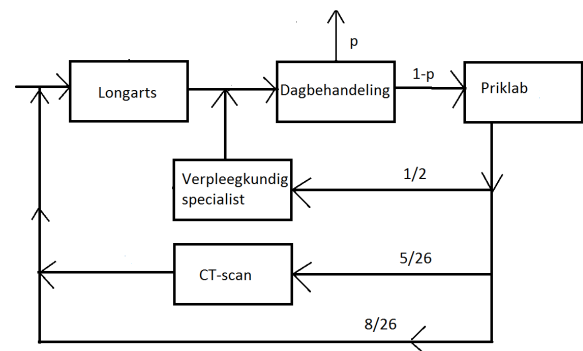
- Na deze afspraak heb ik een versimpeld model opgesteld, waarbij ik nog contact heb gehad met de betrokkene van het Antoni van Leeuwenhoekziekenhuis om nog meer inzicht te krijgen in het behandelingsproces. Dit versimpelde model had de volgende aannames:
 - Patiënten worden het gehele behandelingsproces niet bij dezelfde arts/ verpleegkundig specialist behandeld.
 - Een nieuwe patiënt krijgt in plaats van een intake een gewone controle, hierdoor is de gemiddelde bedieningsduur bij de longarts voor iedere patiënt gelijk.
 - De kans om het systeem te verlaten is constant, dus onafhankelijk van de tijd dat de patiënt al behandeld wordt.
 - Patiënten verlaten het systeem alleen na de dagbehandeling. Er wordt aangenomen dat er geen patiënten het systeem verlaten tussen de CT-scan, priklab, controle en dag-

behandeling en ook niet tussen de intake en de dagbehandeling. Aan gezien deze afspraken vaak op een dag plaats vinden en wanneer patiënten echt heel ziek zijn, zullen ze niet meer naar het ziekenhuis gaan voor een behandeling.

- Een patiënt gaat niet om en om op controle bij de longarts en de verpleegkundig specialist, maar beide met kans $\frac{1}{2}$.
- De bedieningsduren zijn exponentieel verdeeld, want deze zijn geheugenloos.
- De bedieningsduren zijn onderling onafhankelijk en identiek verdeeld.
- De patiënten zijn altijd beschikbaar voor de behandeling.
- De patiënten gaan met kans $\frac{5}{26}$ naar de CT-scan in plaats van 5 keer per jaar.
- De Bedieningsvolgorde is FIFO.
- De looptijden tussen de verschillende stations is nul.
- Het externe aankomstproces bij de longarts is Poisson.
- Ieder station beschikt over een oneindige wachtrij.

Verder had ik een tijdschaal van twee weken genomen, omdat het dan een continu proces is. Wat ik hierbij over het hoofd had gezien is dat een aankomst proces van tien patiënten per twee weken niet gelijk is aan een aankomstproces van vijf patiënten per week. Dus uiteindelijk heb ik gekozen voor een tijdschaal van een week. Iedere patiënt moet namelijk iedere twee weken behandeld worden, dus de patiënten die in week een behandeld worden zullen de patiënten die in week twee behandeld worden nooit tegenkomen in het behandelproces. Daarom kan er een tijdschaal van een week genomen

worden, waarbij de week tussendoor genegeerd wordt, dus er zijn in het systeem zesentwintig weken. Dit proces verloopt dus twee keer tegelijk onafhankelijk van elkaar, namelijk de patiënten van oneven weken en patiënten van de even weken. Een weergave van dit model is te zien in Figuur 26, waarbij p de kans is dat een patiënt het systeem verlaat. Dus $1 - p$ is de kans dat een patiënt in het systeem blijft.



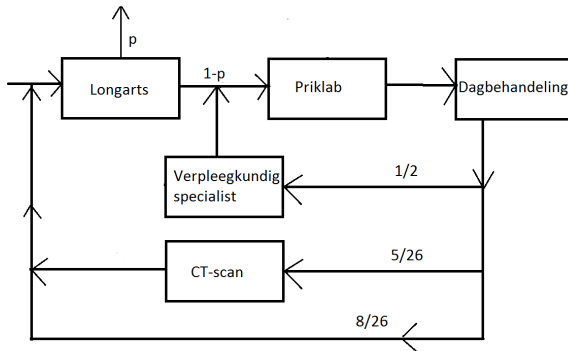
Figuur 26: Een weergave van het model.

- Vervolgens het ik weer contact opgenomen met de betrokkene van het Antoni van Leeuwenhoekziekenhuis en heb ik gevraagd om feedback op dit model. Hieruit bleek dat de dagbehandeling en het priklab omgewisseld moesten worden. Verder verlaten patiënten het systeem na een controle bij de longarts. Patiënten kunnen het systeem alleen verlaten nadat ze een CT-scan hebben gehad. Verder bleek dat de behandeling na een jaar stopt. Er zijn nu onderzoeken of in plaats van de behandeling na een jaar te stoppen, dit ook al na zes maanden kan. Dit heeft natuurlijk als effect dat er minder capaciteit nodig is om hetzelfde aantal patiënten te behandelen. Na 3 maanden heeft 80% het systeem verlaten. Om deze kans mee te nemen in het model is de volgende aanname gemaakt:

- Patiënten verlaten het systeem met

een constante kans. Na drie maanden zijn 80% van de patiënten gestopt met de behandeling. In deze drie maanden zijn de patiënten zes keer behandeld. Dus p is dan 0,24, want $(1 - p)^6 = 0.2$. Hieruit volgt dat $(1 - p) = 0.76$ en dus $p = 0.24$.

Wanneer deze aanpassingen aan het model worden gedaan, volgt hieruit het model zoals weergegeven in Figuur 27.



Figuur 27: Een weergave van het model met de aanpassingen.

Van dit model kunnen de stromingsvergelijkingen opgesteld worden. Een algemene stromingsvergelijking voor station i is weergegeven in (28), waarbij er in het totaal N stations zijn. Hierbij geldt dat λ_i de aankomstintensiteit bij station i is, γ_i de externe aankomstintensiteit is en r_{ji} de kans is dat een patiënt van station j naar station i gaat.

$$\lambda_i = \gamma_i + \sum_{j=1}^N \lambda_j \cdot r_{ji} \quad (28)$$

Voor het model zoals weergegeven in Figuur 27 zijn de stromingsvergelijkingen

te zien in (29), (30), (31), (32) en (33).

$$\lambda_l = \gamma + \frac{8}{26} \lambda_D + \lambda_{CT} \quad (29)$$

$$\lambda_{lab} = (1 - p) \lambda_l + \lambda_v \quad (30)$$

$$\lambda_D = \lambda_{lab} \quad (31)$$

$$\lambda_v = \frac{1}{2} \lambda_D \quad (32)$$

$$\lambda_{CT} = \frac{5}{26} \lambda_D \quad (33)$$

γ : de externe aankomstintensiteit bij de longarts. Dit is gelijk aan vijf per week.

λ_l : de aankomstintensiteit bij de longarts.

λ_{lab} : de aankomstintensiteit bij de prik-lab.

λ_D : de aankomstintensiteit bij de dagbehandeling.

λ_v : de aankomstintensiteit bij de verpleegkundig specialist.

λ_{CT} : de aankomstintensiteit bij de CT-scan.

Deze stromingsvergelijkingen 29, 30, 31, 32 en 33 zijn op te lossen en hieruit volgen de aankomstintensiteiten per week.

$$\lambda_l = 21$$

$$\lambda_{lab} = 32$$

$$\lambda_D = 32$$

$$\lambda_v = 16$$

$$\lambda_{CT} = 6$$

Vervolgens worden de gemiddelde bedieningsduren, die verkregen zijn in samenwerking met het Antoni van Leeuwenhoekziekenhuis, omgezet van minuten naar week. Hiervoor is wel de lengte van een werkweek per station nodig.

longarts: 30 uur per week

Dagbehandeling: 50 uur per week

Priklab: 45 uur per week

Verpleegkundig specialist: 40 uur per week

CT-scan: 48 uur per week

Nu kunnen de gemiddelde bedieningsduren ($\frac{1}{\mu}$) uitgedrukt worden per week

$$\frac{1}{\mu_l} = 15 \text{ minuten} = 0.008 \text{ werkweek} \quad (34)$$

$$\frac{1}{\mu_{lab}} = 20 \text{ minuten} = 0.007 \text{ werkweek} \quad (35)$$

$$\frac{1}{\mu_D} = 45 \text{ minuten} = 0.015 \text{ werkweek} \quad (36)$$

$$\frac{1}{\mu_v} = 15 \text{ minuten} = 0.006 \text{ werkweek} \quad (37)$$

$$\frac{1}{\mu_{CT}} = 30 \text{ minuten} = 0.01 \text{ werkweek} \quad (38)$$

Om te bepalen of het systeem stationair is, zijn eerst het aantal servers per station nodig.

longarts: 8 servers

Dagbehandeling: 48 servers

Priklab: 4 servers

Verpleegkundig specialist: 3 servers

CT-scan: 2 servers

Het systeem is stationair wanneer geldt $\frac{\lambda}{\mu} < s$ voor ieder station, wanneer dit station M/M/s is.

$$\text{longarts:} \quad \frac{\lambda}{\mu} = 0.175$$

$$\text{Dagbehandeling:} \quad \frac{\lambda}{\mu} = 0.48$$

$$\text{Priklab:} \quad \frac{\lambda}{\mu} = 0.22$$

$$\text{Verpleegkundig specialist:} \quad \frac{\lambda}{\mu} = 0.096$$

$$\text{CT-scan:} \quad \frac{\lambda}{\mu} = 0.06$$

Dus het systeem is in evenwicht op het moment dat er 1 longarts $0,175 \cdot 30 = 6$ uur per week beschikbaar is, 1 server bij het priklab 11 uur per week beschikbaar is, 1 bed bij de dagbehandeling 24 uur per week beschikbaar is, 1 verpleegkundig specialist 4 uur per week beschikbaar is en 1 CT-scan 3 uur per week beschikbaar is.

- Om het model vervolgens wat realistischer te maken, ga ik verschillende

type patiënten aan mijn model toevoegen. Hierdoor zijn de volgende aannames niet meer nodig.

- Patiënten verlaten het systeem met een constante kans. Na drie maanden zijn 80% van de patiënten gestopt met de behandeling. In deze drie maanden zijn de patiënten zes keer behandeld. Dus P is dan 0.24, want $(1 - P)^6 = 0.2$. Hieruit volgt dat $(1 - P) = 0.76$ en dus $P = 0.24$
- Een patiënt gaat niet om en om op controle bij de longarts en de verpleegkundig specialist, maar beide met kans $\frac{1}{2}$.
- De patiënten gaan met kans $\frac{5}{26}$ naar de CT-scan in plaats van vijf keer per jaar.

Nu kan ik ook in het model meenemen dat patiënten het systeem alleen verlaten nadat ze een CT-scan hebben gehad. Na drie maanden heeft 80% van de patiënten het systeem verlaten, net zoals in het vorige model. Ook kan er meegenomen worden dat de behandeling stopt na een jaar, dus patiënten kunnen niet langer dan een jaar behandeld worden.

Aangezien er vijf CT-scans plaatsvinden in een jaar, kan een patiënt dus op zes momenten het systeem verlaten, want na een jaar verlaat een patiënt het systeem sowieso. Hieruit volgt dus dat er zes type patiënten zijn, noem dit patiënt type a, b, c, d, e en f . Voor ieder patiënt type ligt de route door het systeem dus vast. Om dit te kunnen beschrijven wordt het behandelproces van een jaar verdeeld in 6 delen.

1. L-V-L-V+CT-L
2. V-L-V+CT-L
3. V-L-V+CT-L
4. V-L-V+CT-L
5. V-L-V+CT-L

6. V-L-V-L-V-longarts

Hierbij staat de V voor een behandelweek waarbij de controle bij de verpleegkunde specialist is, de L voor een behandelweek waarbij de controle bij de longarts is en V+CT een behandelweek waarbij de controle bij de verpleegkundig specialist is en de patiënt in deze week ook een CT-scan krijgt. De L na een CT-scan is alleen de longarts en de rest van de behandelingen in die week vallen onder het volgende pad, aangezien patiënten na de longarts het systeem kunnen verlaten. De longarts aan het einde van stuk 6 wil zeggen dat daar alleen nog maar een controle bij de longarts is.

Zoals te zien is, is hierbij aangenomen dat delen 1 en 6 wat langer duren dan delen 2, 3, 4 en 5. Dit komt omdat bij deel 1 het deel begint bij de longarts en het ook moet eindigen bij de longarts. De delen 2, 3, 4 en 5 beginnen bij de verpleegkundig specialist en eindigen bij de longarts. Verder zijn er 52 weken in een jaar dus 26 behandelweken en hieruit volgt de lengte van deel 6. De behandeling kan alleen beëindigd worden op het moment dat de patiënt bij de CT-scan is geweest. Dus er zijn de volgende type patiënten, namelijk de patiënten die de volgende paden door het systeem afleggen.

- Type *a*: pad 1
- Type *b*: pad 1 + 2
- Type *c*: pad 1 + 2 + 3
- Type *d*: pad 1 + 2 + 3 + 4
- Type *e*: pad 1 + 2 + 3 + 4 + 5
- Type *f*: pad 1 + 2 + 3 + 4 + 5 + 6

Voor deze verschillende type patiënten kan vastgelegd worden wat iedere fase is, zoals te zien is in Tabel 15 in Appendix E.

Hiervoor wordt wel de volgende aanname gemaakt.

- Iedere patiënt heeft een CT-scan na hetzelfde aantal behandelingsweken. Dit is in de praktijk natuurlijk niet het geval, want de CT-scan kan ook wat eerder af wat later gepland worden. Dit hangt ook af van de resultaten van het priklab.

Wanneer nu de kans bekend is of een patiënt die in het systeem opgenomen wordt een patiënt van type *a, b, c, d, e* of *f* is, kan de aankomstintensiteit van deze type patiënten bepaald worden. Hiervoor moet dus de kans op een bepaalde leeftijd in het systeem bekend zijn. Met de leeftijd in het systeem wordt bedoeld hoe lang een patiënt in het systeem is. Het is bekend dat de kans dat de patiënt in het systeem blijft bij nul weken gelijk is aan 1. Verder is er bekend dat na drie maanden al ongeveer 80% van de patiënten met de behandeling gestopt is en na een jaar stopt de behandeling. Het is dus de bedoeling om een functie voor de kans dat een patiënt in het systeem blijft afhankelijk van de verblijftijd van de patiënt in het systeem in weken te hebben. Deze functie kan een sprong hebben bij een verblijftijd van 52 weken, want hier verlaat de patiënt het systeem niet door progressie van de ziekte zoals bij de andere verblijftijden, maar doordat de behandeling dan stopt. Verder moet het een dalende functie door de punten (0, 1) en (12, 0.2) zijn, aangezien deze punten bekend zijn. Ook moet de functie positief blijven, want anders worden de kansen negatief. Er wordt gekozen voor een geometrische functie ($y = q(1 - q)^{n-1}$). Waarbij *n* het aantal keer is dat de patiënt het systeem heeft kunnen verlaten. Aangezien er bekend is dat na 3 maanden nog 20% van de patiënten in het systeem is en dit tijdstip in het tweede deel van de behandeling plaatsvindt volgt hieruit dat $q = 0.8$. Dit is het geval, omdat de

patiënt op dat moment nog maar een keer het systeem heeft kunnen verlaten. Hieruit volgt dus de functie:

$$y = \begin{cases} 0.8(0.2)^{n-1} & \text{voor } n < 6 \\ 0.2^n & \text{voor } n = 6 \end{cases}$$

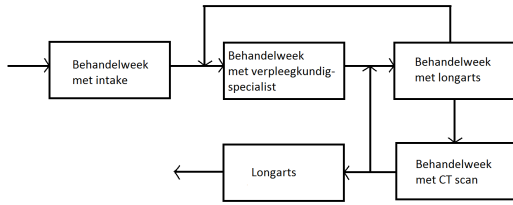
Hierbij is y de kans dat een patiënt het systeem verlaat bij mogelijkheid n . Hieruit kunnen dus de kansen op de verschillende type patiënten berekend worden.

- De kans dat een patiënt van type a is, is de kans dat de patiënt het systeem bij de eerste mogelijkheid verlaat. Hieruit volgt $y = 0.8(0.2)^0 = 0.8$. Dus de aankomstintensiteit van type a patiënten is gemiddeld $5 * 0.8 = 4$ per week.
- De kans dat een patiënt van type b is, is de kans dat de patiënt het systeem bij de tweede mogelijkheid verlaat. Hieruit volgt $y = 0.8(0.2)^1 = 0.16$. Dus de aankomstintensiteit van type b patiënten is gemiddeld $5 * 0.16 = 0.8$ per week.
- De kans dat een patiënt van type c is, is de kans dat de patiënt het systeem bij de derde mogelijkheid verlaat. Hieruit volgt $y = 0.8(0.2)^2 = 0.032$. Dus de aankomstintensiteit van type c patiënten is gemiddeld $5 * 0.032 = 0.16$ per week.
- De kans dat een patiënt van type d is, is de kans dat de patiënt het systeem bij de vierde mogelijkheid verlaat. Hieruit volgt $y = 0.8(0.2)^3 = 0.0064$. Dus de aankomstintensiteit van type d patiënten is gemiddeld $5 * 0.0064 = 0.032$ per week.
- De kans dat een patiënt van type e is, is de kans dat de patiënt het systeem bij de vijfde mogelijkheid verlaat. Hieruit volgt $y = 0.8(0.2)^4 = 0.0013$. Dus de aankomstintensiteit van type e patiënten is gemiddeld $5 * 0.0013 = 0.0064$ per week.

- De kans dat een patiënt van type f is, is de kans dat de patiënt het systeem bij de zesde mogelijkheid verlaat. Dit is dus de kans dat de patiënt de eerste vijf keer in het systeem bleef. Dit is dus gelijk aan $0.2^5 = 3.2 \cdot 10^{-4}$. Dus de aankomstintensiteit van type f patiënten is gemiddeld $5 * 3.2 \cdot 10^{-4} = 0.0016$ per week.

Hieruit volgt dat $v(a) = 4$, $v(b) = 0.8$, $v(c) = 0.16$, $v(d) = 0.032$, $v(e) = 0.0064$ en $v(f) = 0.0016$ per week. Hierbij is $v(i)$ de aankomstintensiteit van type i .

- Het is niet bekend hoeveel capaciteit het Antoni van Leeuwenhoekziekenhuis heeft voor de behandeling van Nivolumab, dus ga ik kijken hoeveel capaciteit er nodig is bij de strategie om vijf nieuwe patiënten per week op te roepen, zoals bij het Antoni van Leeuwenhoekziekenhuis. Hiervoor neem ik eerst aan dat er oneindig veel servers zijn bij ieder station. Hierdoor zou het netwerk gezien kunnen worden zoals in Figuur 28. De vraag is hoeveel capaciteit er nodig is per week. In Figuur 28 is het netwerk te zien, waarbij ieder element staat voor een behandelweek, omdat er oneindig veel servers zijn ontstaat er geen wachtrij in deze behandelweken. Er zijn natuurlijk verschillen in de behandelweek. De eerste week krijgt een patiënt namelijk een intake bij de longarts in plaats van een controle. Wanneer dit netwerk dus gebruikt wordt, is de aanname dat een intake 15 minuten duurt zoals een controle niet meer nodig. Verder zijn er behandelweken met een controle bij de longarts en andere met een controle bij de verpleegkundig specialist. Ook zijn er behandelweken met een CT-scan en andere zonder CT-scan. Tenslotte is er per patiënt één behandelweek die alleen bestaat uit de longarts, dit is namelijk de week dat de patiënt het systeem verlaat.



Figuur 28: Een weergave van het behandelingsproces in weken.

- Vervolgens wordt met de verschillende type patiënten en het netwerk zoals te zien in Figuur 28 een model opgesteld. Dit wordt gedaan met een theorie zoals beschreven in [2]. De fases $s = 1, 2, \dots, S(i)$ van de type a, b, c, d, e en f patiënten zijn weergegeven in Tabel 16 in Appendix E. Hierbij is $S(i)$ de laatste fase van patiënt type i . Dus een patiënt van type i gaat door het netwerk via $r(i, 1), r(i, 2), \dots, r(i, S(i))$. Het eerste component van r is het type en het tweede component is de fase waarin de patiënt zich bevindt. Er worden vervolgens de volgende aanname gemaakt:

- Het aankomstproces is Poisson voor ieder type en deze aankomstprocessen zijn onafhankelijk.
- Het aantal servers per wachtrij is oneindig, aangezien het vast staat in welke week de patiënten behandeld worden. Ze worden niet een week doorgeschoven, doordat ze op andere patiënten hoeven te wachten.

Om dit model te maken wordt er aangenomen dat de patiënten in wachtrij j ($j=1, 2, \dots, J$) gerangschikt zijn. Dit wil zeggen dat wachtrij j patiënten heeft op posities $1, 2, \dots, n_j$. Waarbij n_j het aantal patiënten in wachtrij j is. In wachtrij j heeft de volgende eigenschappen:

- Patiënten hebben een willekeurige

bedieningsduur die exponentieel verdeeld is met een gemiddelde ($\frac{1}{\mu_j}$).

- De totale service prestatie $\phi_j(n_j) = n_j$, want er zijn oneindig veel servers.
- Een deel $\gamma_j(l, n_j) = \frac{1}{n_j}$ van deze prestatie komt van de patiënt op positie l ($l=1, 2, \dots, n_j$). Aangezien er oneindig veel servers zijn is de bijdrage van iedere patiënt aan de prestatie van de wachtrij gelijk. In het algemeen geldt

$$\gamma_j(l, n) = \begin{cases} \frac{1}{n} & l = 1, 2, \dots, n; \\ & n = 1, 2, \dots, K \\ \frac{1}{K} & l = 1, 2, \dots, K; \\ & n = K + 1, K + 2, \dots \\ 0 & \text{anders} \end{cases}$$

Hierbij geldt dat er K servers per station zijn. Wanneer de patiënt op positie l de wachtrij verlaat, dit is het geval als zijn behandeling van wachtrij j volbracht is, verschuiven patiënten van positie $l + 1, l + 2, \dots, n_j$ naar de posities $l, l + 1, \dots, n_j$. Aangezien de bedieningsvolgorde FIFO is, geldt dat alleen de patiënt op positie 1 de wachtrij kan verlaten.

- Wanneer een patiënt arriveert in wachtrij j wordt zijn positie l met kans $\delta_j(l, n_j + 1)$. Aangezien een nieuwe patiënt in de wachtrij altijd de laatste positie krijgt geldt $\delta_j(n_j + 1, n_j + 1) = 1$ en voor alle andere posities l geldt $\delta_j(l, n_j + 1) = 0$.

Noem totale bediening die een patiënt bij een wachtrij nodig heeft zijn benodigde bediening. Vervolgens wordt er aangenomen dat de benodigde bediening onafhankelijk is van elkaar en van het tijdstip waarop de patiënt in het systeem komt. Neem nu $t_j(l)$ het type patiënt in positie l van wachtrij j en zij $s_j(l)$ de bijbehorende fase van de route van de patiënt. Noem $c_j(t_j(l), s_j(l))$ de

klasse van deze patiënt. Aangezien patiënten vaker dezelfde wachtrij bezoeken, geeft de klasse van een patiënt meer informatie dan alleen het type patiënt. De vector $\mathbf{c}_j = (c_j(1), c_j(2), \dots, c_j(n_j))$ beschrijft de toestand van wachtrij j en $\mathbf{C} = (\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_J)$ is een Markov proces representatie van de toestand van het systeem. Definieer $\alpha_j(i, s)$ en a_j als volgt

$$\alpha_j(i, s) = \begin{cases} v(i) & \text{als } r(i, s) = j \\ 0 & \text{anders} \end{cases}$$

en

$$a_j = \sum_{i=1}^I \sum_{s=1}^{S(i)} \alpha_j(i, s)$$

Wanneer het systeem in evenwicht is, is a_j het gemiddelde aantal patiënten dat arriveert bij wachtrij j per tijdseenheid. Zij

$$b_j^{-1} = \sum_{n=0}^{\infty} \frac{a_j^n}{\prod_{l=1}^n \phi_j(l)}$$

Vervolgens wordt er aangenomen dat $b_1, b_2, \dots, b_5$ niet gelijk aan nul zijn. Deze aanname is nodig om zeker te weten dat de evenwichtsverdeling bestaat. Definieer

$$\pi_j(\mathbf{c}_j) = b_j \prod_{l=1}^{n_j} \frac{\alpha_j(t_j(l), s_j(l))}{\phi_j(l)}$$

Dan is de evenwichtsverdeling van het netwerk gelijk aan

$$\pi(\mathbf{C}) = \prod_{j=1}^J \pi_j(\mathbf{c}_j)$$

Deze theorie moet dus toegepast worden op het netwerk zoals is weergegeven in Figuur 28 met de patiënt types a, b, c, d, e , en f . Dit netwerk heeft vijf verschillende wachtrijen, dus $J = 5$. Verder zijn er zes verschillende type patiënten, dus $I=6$. Ook zijn de aankomstintensiteiten per type bekend, namelijk $v(a) = 4$,

$v(b) = 0.8$, $v(c) = 0.16$, $v(d) = 0.032$, $v(e) = 0.0064$ en $v(f) = 0.0016$ per week. Dus hiermee kan $\alpha_j(i, s)$ bepaald worden.

Noem behandelweek met intake wachtrij 1, behandelweek met verpleegkundig specialist wachtrij 2, behandelweek met longarts wachtrij 3, behandelweek met CT-scan wachtrij 4 en longarts wachtrij 5. In Tabel 20 is weergegeven hoe vaak een patiënt van type a, b, c, d, e , of f wachtrij 1,2,3,4 en 5 bezoekt in het behandelingsproces. Hieruit volgen de waarden voor a_j , zoals hieronder te zien is.

$$\begin{aligned} a_1 &= v(a) + v(b) + v(c) + v(d) \\ &+ v(e) + v(f) = 5 \\ a_2 &= v(a) + 2v(b) + 3v(c) + 4v(d) \\ &+ 5v(e) + 8v(f) = 6.2528 \\ a_3 &= v(a) + 3v(b) + 5v(c) + 7v(d) \\ &+ 9v(e) + 12v(f) = 7.5008 \\ a_4 &= v(a) + 2v(b) + 3v(c) + 4v(d) \\ &+ 5v(e) + 5v(f) = 6.2480 \\ a_5 &= v(a) + v(b) + v(c) + v(d) \\ &+ v(e) + v(f) = 5 \end{aligned}$$

Met behulp van deze a_j met $j = 1, 2, \dots, 5$ kan b_j^{-1} bepaald worden. Hiervoor wordt het feit gebruikt dat $\frac{(\lambda)^k}{k!} e^{-\lambda}$ de poissonverdeling is. Hieruit volgt dat $\sum_{k=0}^{\infty} \frac{(\lambda)^k}{k!} e^{-\lambda} = 1$. Dus $\sum_{k=0}^{\infty} \frac{(\lambda)^k}{k!} = e^{\lambda}$. Hieronder is dat voor b_1^{-1} te zien. Dit zelfde wordt toegepast voor $b_2^{-1}, b_3^{-1}, b_4^{-1}$

Tabel 20: Het aantal keer dat een patiënt van een bepaald type een wachtrij bezoekt

Wachtrij	Type a	Type b	Type c	Type d	Type e	Type f
1	1	1	1	1	1	1
2	1	2	3	4	5	8
3	1	3	5	7	9	12
4	1	2	3	4	5	5
5	1	1	1	1	1	1

en b_5^{-1} .

$$\begin{aligned}
 b_1^{-1} &= \sum_{n=0}^{\infty} \frac{a_1^n}{\prod_{l=1}^n \phi_1(l)} \\
 &= \sum_{n=0}^{\infty} \frac{5^n}{n!} \\
 &= e^5 \\
 b_2^{-1} &= e^{6.2528} \\
 b_3^{-1} &= e^{7.5008} \\
 b_4^{-1} &= e^{6.2480} \\
 b_5^{-1} &= e^5
 \end{aligned}$$

Hieruit volgen dus de b_j zoals hieronder te zien is.

$$\begin{aligned}
 b_1 &= e^{-5} \\
 b_2 &= e^{-6.2528} \\
 b_3 &= e^{-7.5008} \\
 b_4 &= e^{-6.2480} \\
 b_5 &= e^{-5}
 \end{aligned}$$

Hiermee kunnen vervolgens de $\pi_j(\mathbf{c}_j)$ be-

paald worden.

$$\begin{aligned}
 \pi_1(\mathbf{c}_1) &= b_1 \prod_{l=1}^{n_1} \frac{\alpha_1(t_1(l), s_1(l))}{\phi_1(l)} \\
 &= e^{-5} \prod_{l=1}^{n_1} \frac{\alpha_1(t_1(l), s_1(l))}{l} \\
 &= e^{-5} n_1! \prod_{l=1}^{n_1} \alpha_1(t_1(l), s_1(l)) \\
 \pi_2(\mathbf{c}_2) &= e^{-6.2528} n_2! \prod_{l=1}^{n_2} \alpha_1(t_2(l), s_2(l)) \\
 \pi_3(\mathbf{c}_3) &= e^{-7.5008} n_3! \prod_{l=1}^{n_3} \alpha_1(t_3(l), s_3(l)) \\
 \pi_4(\mathbf{c}_4) &= e^{-6.2480} n_4! \prod_{l=1}^{n_4} \alpha_1(t_4(l), s_4(l)) \\
 \pi_5(\mathbf{c}_5) &= e^{-5} n_5! \prod_{l=1}^{n_5} \alpha_1(t_5(l), s_5(l))
 \end{aligned}$$

Met deze $\pi_j(\mathbf{c}_j)$ kan dus uiteindelijk de evenwichtsverdeling van het netwerk bepaald worden. Namelijk door deze $\pi_j(\mathbf{c}_j)$ met elkaar te vermenigvuldigen. Om de evenwichtsverdeling te bepalen is alleen wel $\alpha_j(t_j(l), s_j(l))$ nodig voor $j = 1, 2, \dots, 5$. Dus hiervoor moet bekend zijn hoeveel patiënten van ieder type in de verschillende wachtrijen staan.

- Nu wordt de benodigde capaciteit bij oneindig veel servers bepalen, zodat ik deze capaciteit in een M/M/1 model kan gebruiken om zo de wachttijden wel mee te kunnen nemen. Al zijn deze wachttijden wel begrensd, want een patiënt moet wel een behandelweek binnen een week volbrengen, want anders gaat deze patiënt van bijvoorbeeld de behandelweken

in de oneven weken naar de behandelweken van de even weken. Dit is niet mogelijk. Om dit te doen is het dus handiger om terug te gaan naar het model zoals te zien is in Figuur 27, want hier zijn de verschillende afdelingen van het behandelingsproces een aparte wachtrij. In het model zoals in Figuur 28 is bijvoorbeeld de dagbehandeling een onderdeel van bijna iedere wachtrij, waardoor het lastiger is om iets over de wachtrij van de dagbehandeling te zeggen. Eerst worden de verschillende wachtlijnen van het netwerk genummerd. Hiervoor wordt de volgende aanname dus weer gemaakt.

- Een nieuwe patiënt krijgt in plaats van een intake een gewone controle, hierdoor is de gemiddelde bedieningsduur bij de longarts voor iedere patiënt gelijk.

Noem de longarts wachtrij 1, de prikklab wachtrij 2, de dagbehandeling wachtrij 3, de verpleegkundig specialist wachtrij 4 en de CT-scan wachtrij 5. Vervolgens moeten de aankomstintensiteiten van ieder type patiënt bepaald worden, zoals bij het andere netwerk ook is gebeurd. De aankomstintensiteiten van de verschillende type patiënten zijn hetzelfde gebleven, namelijk $\nu(a) = 4$, $\nu(b) = 0.8$, $\nu(c) = 0.16$, $\nu(d) = 0.032$, $\nu(e) = 0.0064$ en $\nu(f) = 0.0016$ per week. Nu moet er dus alleen nog bepaald worden hoe vaak een bepaald type patiënt bij een bepaalde wachtrij komt.

Met behulp van Tabel 21 kunnen de aankomstintensiteiten van patiënten bij ie-

dere wachtrij per week bepaald worden.

$$\begin{aligned} a_1 &= 3\nu(a) + 5\nu(b) + 7\nu(c) + 9\nu(d) \\ &\quad + 11\nu(e) + 14\nu(f) = 17.5 \\ a_2 &= 4\nu(a) + 8\nu(b) + 12\nu(c) + 16\nu(d) \\ &\quad + 20\nu(e) + 26\nu(f) = 25 \\ a_3 &= 4\nu(a) + 8\nu(b) + 12\nu(c) + 16\nu(d) \\ &\quad + 20\nu(e) + 26\nu(f) = 25 \\ a_4 &= 2\nu(a) + 4\nu(b) + 6\nu(c) + 8\nu(d) \\ &\quad + 10\nu(e) + 13\nu(f) = 12.5 \\ a_5 &= \nu(a) + 2\nu(b) + 3\nu(c) + 4\nu(d) \\ &\quad + 5\nu(e) + 5\nu(f) = 6.2 \end{aligned}$$

De gemiddelde behandelingsduur bij iedere wachtrij is bekend, deze zijn hieronder weergegeven in minuten.

$$\begin{aligned} \frac{1}{\mu_1} &= 15 \\ \frac{1}{\mu_2} &= 20 \\ \frac{1}{\mu_3} &= 45 \\ \frac{1}{\mu_4} &= 15 \\ \frac{1}{\mu_5} &= 30 \end{aligned}$$

Met behulp van de a_j en de μ_j kan dus de gemiddelde capaciteit van wachtrij j bepaald worden.

Gemiddelde capaciteit wachtrij 1: $17.5 \cdot 15 = 262.5$ minuten per week.

Gemiddelde capaciteit wachtrij 2: $25 \cdot 20 = 500$ minuten per week.

Gemiddelde capaciteit wachtrij 3: $25 \cdot 45 = 1125$ minuten per week.

Gemiddelde capaciteit wachtrij 4: $12.5 \cdot 15 = 187.5$ minuten per week.

Gemiddelde capaciteit wachtrij 5: $6.2 \cdot 30 = 178.5$ minuten per week.

Nu worden deze capaciteiten gebruikt om een M/M/1 model te maken, zodat de wachtlijnen meegenomen kunnen worden. Dit M/M/1 model wordt op dezelfde manier gemaakt als het model met oneindig veel servers. Dus er

Tabel 21: Het aantal keer dat een patiënt van een bepaald type een wachtrij bezoekt bij het model zoals te zien is in Figuur 27

Wachtrij	Type a	Type b	Type c	Type d	Type e	Type f
1	3	5	7	9	11	14
2	4	8	12	16	20	26
3	4	8	12	16	20	26
4	2	4	6	8	10	13
5	1	2	3	4	5	5

wordt ook hier gebruik gemaakt van de verschillende type patiënten a, b, c, d, e en f . Ook hier geldt dat de fases van de type weergegeven zijn in Tabel 15. Het verschil is hier dat de totale service prestatie $\phi_j(n_j) = 1$, omdat er maar een server is per station. Ook kunnen er nu wel patiënten in de wachtrij staan, wat bij een model met oneindig veel servers natuurlijk niet mogelijk is. De aankomstintensiteiten van de verschillende stations blijven hetzelfde. Dit is het geval, omdat de stromingsvergelijkingen hetzelfde blijven als in het geval met oneindig veel servers. Nu kan er gekeken worden wanneer de stations stationair zijn. Aangezien er aangenomen is dat ieder station M/M/1 is, moet voor ieder station $j=1,2,\dots,5$ gelden dat $\frac{\lambda_j}{\mu_j} < 1$. De gemiddelde aankomstintensiteit ligt vast, maar de gemiddelde bedieningsduur kan gevarieerd worden door de capaciteit. Neem hiervoor aan

- Neem aan M/M/1 station j met een gemiddelde bedieningsduur $\frac{1}{s \cdot \mu_j}$ voor een station met s servers. Dit is natuurlijk niet hetzelfde, want de snelheid van de bediening wordt dus s keer zo snel. Op het moment dat er bij een station met $s > 1$ servers een server geen patiënt in behandeling heeft, wordt deze wel als werkend beschouwd in deze aanname. Wanneer $s < 1$ kan dit worden gezien als 1 server met een kortere werkweek. De werkweek is dan

$s \cdot$ normale werkweek.

Vervolgens wordt de formule van Little, (39), toegepast om de gemiddelde verblijftijd te bepalen. Hierbij is $E\{L\}$ het gemiddeld aantal klanten in systeem en $E\{F\}$ de gemiddelde verblijftijd per klant.

$$E\{L\} = \lambda \cdot E\{F\} \quad (39)$$

Aangezien het model uit M/M/1 stations bestaat, kan voor ieder M/M/1 station (40) gebruikt worden om de gemiddelde verblijftijd te bepalen. Hierbij geldt dat $\rho = \frac{\lambda}{\mu}$. Ieder station j kan opgevat worden als een onafhankelijk M/M/1 station, wanneer het stationair is. Dit komt omdat de stationaire verdeling van het systeem het product van de marginalen is. Dus geldt dat er op een vast moment, wanneer het systeem in evenwicht is, dat het aantal klanten bij de stations onafhankelijk is.

$$E\{L\} = \lambda \cdot E\{F\} = \frac{\rho}{1 - \rho} \quad (40)$$

Als alle stations stationair zijn, is het systeem stationair.

$$\rho_1 = \frac{\lambda_1}{s_1 \cdot \mu_1} = \frac{17.5 \cdot 0.008}{s_1} < 1 \quad (41)$$

$$\rho_2 = \frac{\lambda_2}{s_2 \cdot \mu_2} = \frac{25 \cdot 0.015}{s_2} < 1 \quad (42)$$

$$\rho_3 = \frac{\lambda_3}{s_3 \cdot \mu_3} = \frac{25 \cdot 0.007}{s_3} < 1 \quad (43)$$

$$\rho_4 = \frac{\lambda_4}{s_4 \cdot \mu_4} = \frac{12.5 \cdot 0.006}{s_4} < 1 \quad (44)$$

$$\rho_5 = \frac{\lambda_5}{s_5 \cdot \mu_5} = \frac{6.2 \cdot 0.01}{s_5} < 1 \quad (45)$$

Hieruit volgt dus het minimale aantal servers per station.

$$s_1 > 17.5 \cdot 0.008 = 0.14$$

$$s_2 > 25 \cdot 0.015 = 0.375$$

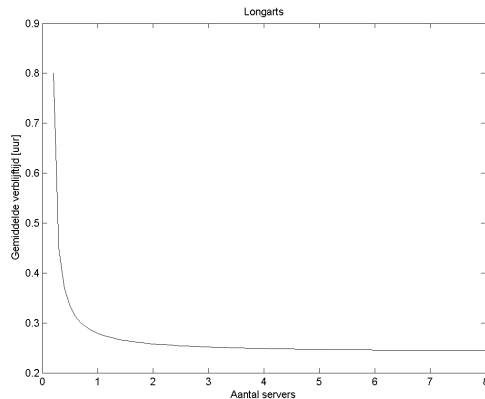
$$s_3 > 25 \cdot 0.007 = 0.175$$

$$s_4 > 12.5 \cdot 0.006 = 0.75$$

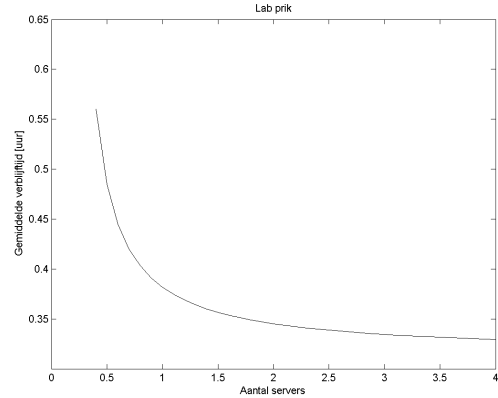
$$s_5 > 6.2 \cdot 0.01 = 0.062$$

Nu kan voor ieder station de gemiddelde verblijftijd bepaald worden. Eerst moet ρ bepaald worden. Hiervoor zijn de gemiddelde bedieningsduren zoals weergegeven in (34), (35), (36), (37) en (38) gebruikt. Vervolgens is ρ bepaald met behulp van (41), (42), (43), (44) en (45). De gemiddelde verblijftijd kan nu bepaald worden met behulp van (46). Dit kan gedaan worden met verschillende waarden van s , hiervoor moet gelden dat het station wel stationair is en het maximum van s is het aantal servers van het station die er in het Antoni van Leeuwenhoek-ziekenhuis aanwezig zijn. Wanneer (46) gebruikt wordt is de uitkomst $E\{F\}$ per week, dus om dit om te rekenen naar uren wordt $E\{F\}$ vermenigvuldigd met $s \cdot \text{lengte werkweek}$

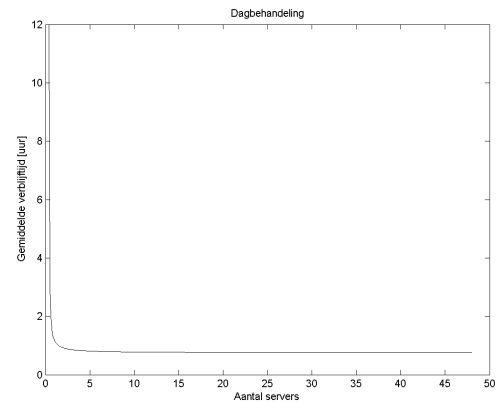
$$E\{F\} = \frac{\rho}{\lambda(1-\rho)} \quad (46)$$



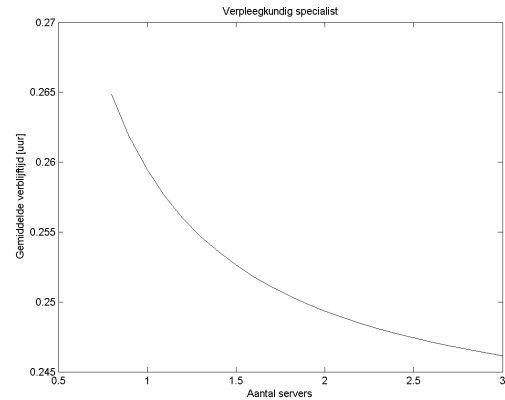
Figuur 29: De gemiddelde verblijftijd tegen het aantal servers bij de longarts



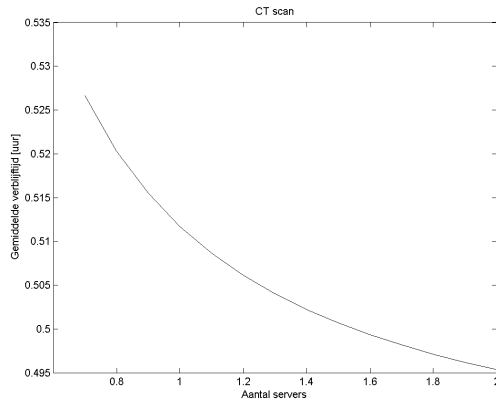
Figuur 30: De gemiddelde verblijftijd tegen het aantal servers bij het prik-lab



Figuur 31: De gemiddelde verblijftijd tegen het aantal servers bij de dagbehandeling



Figuur 32: De gemiddelde verblijftijd tegen het aantal servers bij de verpleegkundig specialist



Figuur 33: De gemiddelde verblijftijd tegen het aantal servers bij de CT-scan

In Figuur 29, 30, 31, 32 en 33 is de gemiddelde verblijftijd tegen het aantal servers te zien bij respectievelijk de longarts, priklab, dagbehandeling, verpleegkundig specialist en de CT-scan. Wanneer er 1 server per station de gehele werkweek beschikbaar is voor de behandeling, is de gemiddelde verblijftijd per station respectievelijk 16.7 minuten, 22.9 minuten, 72 minuten, 15.6 minuten en 30.7 minuten.

- Hetzelfde kan gedaan worden voor het behandelingsproces dat maximaal zes maanden duurt. Er zijn 13 behandelingsweken en in deze behandelingsweken wordt er twee keer een CT-scan gemaakt. Er zijn weer verschillende type patiënten, alleen dit keer zijn er maar drie verschillende types. Namelijk de patiënten die na de eerste CT-scan het systeem verlaten, patiënten die na de tweede CT-scan het systeem verlaten en patiënten die na zes maanden het systeem verlaten. Dus het behandelingsproces wordt verdeeld over drie delen.

1. L-V-L-V+CT-L

2. V-L-V+CT-L

3. V-L-V-longarts

Hieruit volgen de verschillende types, net zoals bij het behandelingsproces van een jaar. Type a en type b blijven hetzelfde, alleen type c is in dit geval iets anders dan bij het behandelingsproces van een jaar. In dit geval is type c het type patiënt dat het langste in het systeem blijft. Dus hoeft er aan het einde van type c geen CT-scan gemaakt te worden, wat wel het geval is bij het behandelingsproces van een jaar. Voor ieder type kan hiermee vastgelegd worden wat iedere fase is, dit is te zien in Tabel 17 in Appendix E. De kans op type a blijft hetzelfde, dus de aankomstintensiteit van type a patiënten is gemiddeld vier per week. Hetzelfde geldt voor de type b patiënten. Dus de aankomstintensiteit van type b patiënten is gemiddeld 0.8 per week. Voor type c patiënten geldt dat de kans hierop niet gelijk is aan die van het behandelingsproces van een jaar, want de types d , e en f van het behandelingsproces van een jaar horen nu bij type c . Dus de kans dat een patiënt een patiënt van type c is, is gelijk aan 0.04. Hieruit volgt dat de aankomstintensiteit van type c patiënten gemiddeld $0.04 \cdot 5 = 0.2$ per week is. Om nu de gemiddelde aankomstintensiteiten per station te bepalen, moet er eerst bekend zijn hoe vaak een patiënt van een bepaald type ieder station bezoekt. Dit is te zien in Tabel 22. Met behulp van Tabel 22 kunnen de gemiddelde aankomstintensiteiten van patiënten bij iedere wachtrij per week bepaald worden.

$$a_1 = 3v(a) + 5v(b) + 7v(c) = 17.4$$

$$a_2 = 4v(a) + 8v(b) + 12v(c) = 24.8$$

$$a_3 = 4v(a) + 8v(b) + 12v(c) = 24.8$$

$$a_4 = 2v(a) + 4v(b) + 6v(c) = 12.4$$

$$a_5 = v(a) + 2v(b) + 2v(c) = 6$$

Nu kan de formule van Little, (46), weer toegepast worden, om grafieken te verkrijgen voor de gemiddelde verblijftijd per station voor verschillende waardes

Tabel 22: Het aantal keer dat een patiënt van een bepaald type een wachtrij bezoekt bij het model zoals te zien is in Figuur 27 met een duur van maximaal zes maanden

Wachtrij	Type a	Type b	Type c
1	3	5	7
2	4	8	12
3	4	8	12
4	2	4	6
5	1	2	2

van s , hiervoor moet gelden dat het station wel stationair. Hiervoor moet wel eerst gecontroleerd worden of de stations stationair zijn. De s waarvoor het systeem stationair is wordt hieronder berekend.

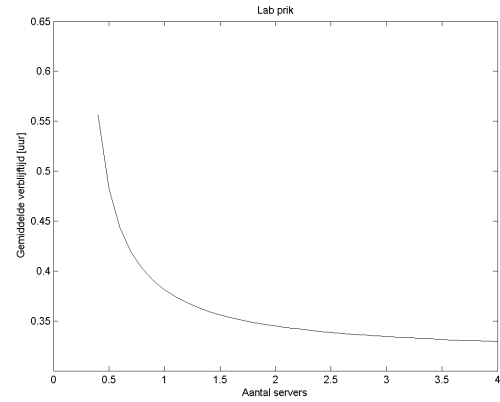
$$s_1 > 17.4 \cdot 0.008 = 0.14$$

$$s_2 > 24.8 \cdot 0.015 = 0.372$$

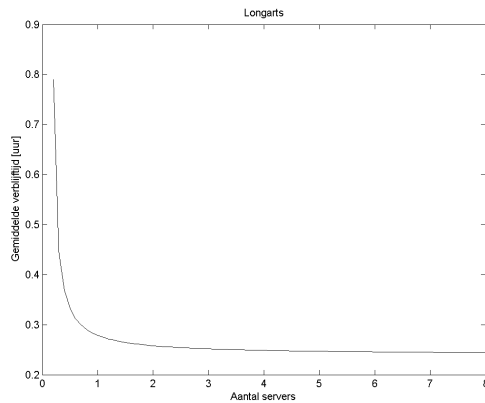
$$s_3 > 24.8 \cdot 0.007 = 0.174$$

$$s_4 > 12.4 \cdot 0.006 = 0.074$$

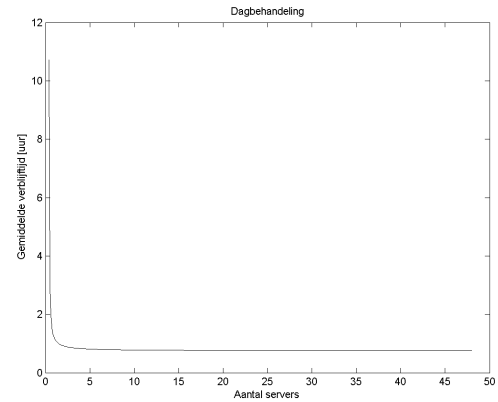
$$s_5 > 6 \cdot 0.01 = 0.06$$



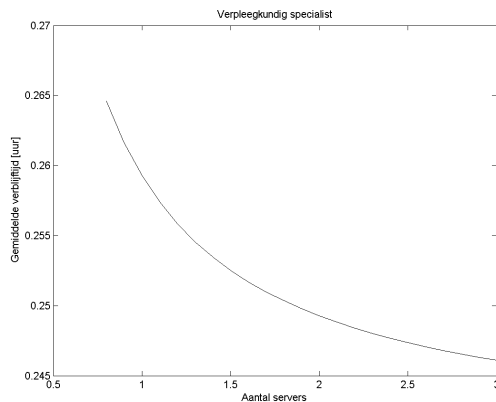
Figuur 35: De gemiddelde verblijftijd tegen het aantal servers bij het prik-lab met een behandelingsproces van maximaal een half jaar.



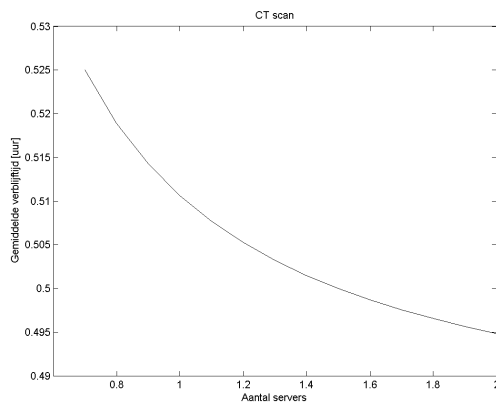
Figuur 34: De gemiddelde verblijftijd tegen het aantal servers bij de longarts met een behandelingsproces van maximaal een half jaar.



Figuur 36: De gemiddelde verblijftijd tegen het aantal servers bij de dagbehandeling met een behandelingsproces van maximaal een half jaar.



Figuur 37: De gemiddelde verblijftijd tegen het aantal servers bij de verpleegkundig specialist met een behandelingsproces van maximaal een half jaar.



Figuur 38: De gemiddelde verblijftijd tegen het aantal servers bij de CT-scan met een behandelingsproces van maximaal een half jaar.

In Figuur 34, 35, 36, 37 en 38 is de gemiddelde verblijftijd tegen het aantal servers te zien bij respectievelijk de longarts, priklab, dagbehandeling, verpleegkundig specialist en de CT-scan. Wanneer er 1 server per station de gehele werkweek beschikbaar is voor de behandeling, is de gemiddelde verblijftijd per station respectievelijk 16.7 minuten, 22.8 minuten, 71.7 minuten, 15.6 minuten en 30.6 minuten. Wanneer dit vergeleken wordt met de gemiddelde verblijftijden per station bij een

behandelingsproces van maximaal een jaar, is er te zien dat de verschillen heel klein zijn. Dit is te verklaren doordat er heel weinig patiënten van type *e* en *f* zijn, waardoor de gemiddelde aankomstintensiteiten bij een station met een behandelingsproces van maximaal zes maanden niet veel verschillen van de gemiddelde aankomstintensiteiten bij hetzelfde station met een behandelingsproces van maximaal een jaar.

- Het hierboven beschreven model met verschillende types, gaat ervan uit dat de CT-scans altijd gelijk verdeeld zijn over het jaar. Maar op het moment dat een patiënt al voor de eerste CT-scan heel slecht op Nivolumab reageert, wordt er natuurlijk al eerder een CT-scan gemaakt. Dus eigenlijk zou een CT-scan voor ieder bezoek van de longarts gepland kunnen worden. Hierdoor ontstaan er natuurlijk meer types, aangezien een patiënt iedere vier weken het systeem kan verlaten. Er wordt nu weer uitgegaan van een behandelingsproces van maximaal een jaar. Verder geldt wel dat de CT-scans verdeeld over het jaar gepland blijven, wanneer er geen aanleiding is om deze eerder te plannen. Er wordt verder aangenomen dat, wanneer de CT-scan vervroegd wordt, de patiënt ook het systeem verlaat. Er wordt niet meegenomen dat de CT-scan dan ook positief kan zijn en de patiënt dus in het systeem kan blijven. Hieruit komen de volgende type patiënten, waarbij alle patiënten die na de laatste CT-scan controle nog in het systeem zijn tot hetzelfde type behoren, omdat er dus geen CT-scans meer gepland hoeven te worden en het jaar nog niet vol is.

- Type 1:
L-V+CT-Longarts
- Type 2:
L-V-L-V+CT-Longarts
- Type 3:
L-V-L-V+CT-L-V-longarts

- Type 4:
L-V-L-V+CT-L-V-L-V+CT-longarts
- Type 5:
L-V-L-V+CT-L-V-L-V+CT-L-V+CT-longarts
- Type 6:
L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-longarts
- Type 7:
L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-L-V+CT-longarts
- Type 8:
L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-longarts
- Type 9:
L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-L-V+CT-longarts
- Type 10:
L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-longarts
- Type 11:
L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-L-V-L-V+CT-longarts

Er geldt nog steeds dat er na 3 maanden behandelen nog maar 20% van de patiënten in het systeem is. Dus 80% van de patiënten is type 1 of type 2, want dit zijn de enige types die minder dan 3 maanden behandeld worden. Nu kunnen de kansen op de verschillende type patiënten bepaald worden met behulp van een geometrische verdeling, waarbij p de kans is dat een patiënt het systeem verlaat. Dus de kans dat een patiënt van type 1 is, is de kans dat de patiënt de eerste mogelijkheid het systeem verlaat, dus P . De kans dat een patiënt van type 2 is, is de kans dat de patiënt de eerste keer nog geen CT-scan hoeft te hebben, neem aan dat deze kans gelijk is aan $1 - p$, en de tweede

keer het systeem verlaat. Dus de kans dat een patiënt van type 2 is, is $p(1 - p)$. Er geldt dus dat $p + (1 - p)p = 0.8$ en hieruit volgt $p = 0.55$ of $p = 1.45$, aangezien p een kans is moet p kleiner dan één zijn. Hieruit volgt dus dat $p = 0.55$ en de volgende functie voor de kans.

$$y = \begin{cases} 0.55(0.45)^{n-1} & \text{voor } n < 11 \\ 0.45^{n-1} & \text{voor } n = 11 \end{cases}$$

Met behulp van deze functie kunnen de aankomstintensiteiten van de verschillende type patiënten bepaald worden. De volgende berekeningen zijn gemaakt met behulp van MATLAB, zoals te zien is in Appendix F

- De kans dat een patiënt van type 1 is, is de kans dat de patiënt het systeem bij de eerste mogelijkheid verlaat. Hieruit volgt $y = 0.55(0.45)^0 = 0.55$. Dus de aankomstintensiteit van type 1 patiënten is gemiddeld $5 \cdot 0.55 = 2.75$ per week.
- De kans dat een patiënt van type 2 is, is de kans dat de patiënt het systeem bij de tweede mogelijkheid verlaat. Hieruit volgt $y = 0.55(0.45)^1 = 0.2475$. Dus de aankomstintensiteit van type 2 patiënten is gemiddeld $5 \cdot 0.2475 = 1.2375$ per week.
- De kans dat een patiënt van type 3 is, is de kans dat de patiënt het systeem bij de derde mogelijkheid verlaat. Hieruit volgt $y = 0.55(0.45)^2 = 0.1114$. Dus de aankomstintensiteit van type 3 patiënten is gemiddeld $5 \cdot 0.1111 = 0.5569$ per week.
- De kans dat een patiënt van type 4 is, is de kans dat de patiënt het systeem bij de vierde mogelijkheid verlaat. Hieruit volgt $y = 0.55(0.45)^3 = 0.0501$. Dus de aankomstintensiteit van type 4 patiënten is gemiddeld $5 \cdot 0.0501 = 0.2506$ per week.
- De kans dat een patiënt van type 5

is, is de kans dat de patiënt het systeem bij de vijfde mogelijkheid verlaat. Hieruit volgt $y = 0.55(0.45)^4 = 0.0226$. Dus de aankomstintensiteit van type 5 patiënten is gemiddeld $5 \cdot 0.0226 = 0.1128$ per week.

- De kans dat een patiënt van type 6 is, is de kans dat de patiënt het systeem bij de zesde mogelijkheid verlaat. Hieruit volgt $y = 0.55(0.45)^5 = 0.0101$. Dus de aankomstintensiteit van type 6 patiënten is gemiddeld $5 \cdot 0.0101 = 0.0507$ per week.
- De kans dat een patiënt van type 7 is, is de kans dat de patiënt het systeem bij de zevende mogelijkheid verlaat. Hieruit volgt $y = 0.55(0.45)^6 = 0.0046$. Dus de aankomstintensiteit van type 7 patiënten is gemiddeld $5 \cdot 0.0046 = 0.0228$ per week.
- De kans dat een patiënt van type 8 is, is de kans dat de patiënt het systeem bij de achtste mogelijkheid verlaat. Hieruit volgt $y = 0.55(0.45)^7 = 0.0021$. Dus de aankomstintensiteit van type 8 patiënten is gemiddeld $5 \cdot 0.0021 = 0.0103$ per week.
- De kans dat een patiënt van type 9 is, is de kans dat de patiënt het systeem bij de negende mogelijkheid verlaat. Hieruit volgt $y = 0.55(0.45)^8 = 9 \cdot 10^{-4}$. Dus de aankomstintensiteit van type 9 patiënten is gemiddeld $5 \cdot 9 \cdot 10^{-4} = 0.0046$ per week.
- De kans dat een patiënt van type 10 is, is de kans dat de patiënt het systeem bij de tiende mogelijkheid verlaat. Hieruit volgt $y = 0.55(0.45)^9 = 4 \cdot 10^{-4}$. Dus de aankomstintensiteit van type 10 patiënten is gemiddeld $5 \cdot 4 \cdot 10^{-4} = 0.0021$ per week.
- De kans dat een patiënt van type 11 is, is de kans dat de patiënt het systeem bij de elfde mogelijkheid verlaat. Dit is dus de kans dat de

patiënt de eerste tien keer in het systeem bleef. Dit is dus gelijk aan $0.45^{10} = 3 \cdot 10^{-4}$. Dus de aankomstintensiteit van type f patiënten is gemiddeld $5 \cdot 3 \cdot 10^{-4} = 0.0017$ per week.

Vervolgens wordt Tabel 23 opgesteld met daarin het aantal keer dat een type patiënt een bepaald station bezoekt. Met behulp van Tabel 23 kunnen de aankomstintensiteiten per station bepaald worden, zoals hieronder te zien is.

$$\begin{aligned}
 a_1 &= 2\nu(1) + 3\nu(2) + 4\nu(3) + 5\nu(4) \\
 &\quad + 6\nu(5) + 7\nu(6) + 8\nu(7) + 9\nu(8) \\
 &\quad + 10\nu(9) + 11\nu(10) + 14\nu(11) \\
 &= 14.1 \\
 a_2 &= 2\nu(1) + 4\nu(2) + 6\nu(3) + 8\nu(4) \\
 &\quad + 10\nu(5) + 12\nu(6) + 14\nu(7) + 16\nu(8) \\
 &\quad + 18\nu(9) + 20\nu(10) + 24\nu(11) \\
 &= 18.2 \\
 a_3 &= 2\nu(1) + 4\nu(2) + 6\nu(3) + 8\nu(4) \\
 &\quad + 10\nu(5) + 12\nu(6) + 14\nu(7) \\
 &\quad + 16\nu(8) + 18\nu(9) + 20\nu(10) \\
 &\quad + 24\nu(11) = 18.2 \\
 a_4 &= \nu(1) + 2\nu(2) + 3\nu(3) + 4\nu(4) \\
 &\quad + 5\nu(5) + 6\nu(6) + 7\nu(7) + 8\nu(8) \\
 &\quad + 9\nu(9) + 10\nu(10) + 13\nu(11) \\
 &= 9.1 \\
 a_5 &= \nu(1) + \nu(2) + 2\nu(3) + 2\nu(4) \\
 &\quad + 3\nu(5) + 3\nu(6) + 4\nu(7) + 4\nu(8) \\
 &\quad + 5\nu(9) + 5\nu(10) + 5\nu(11) \\
 &= 6.3
 \end{aligned}$$

Met deze aankomstintensiteiten kan bepaald worden wanneer ieder station in evenwicht is. Hieruit volgt een minimaal aantal servers per station, zoals hieronder

Tabel 23: Het aantal keer dat een patiënt van een bepaald type een wachtrij bezoekt bij het model zoals te zien is in Figuur 27 met type patiënten 1 tot en met 11

Wachtrij	Type 1	Type 2	Type 3	Type 4	Type 5	Type 6	Type 7	Type 8	Type 9	Type 10	Type 11
1	2	3	4	5	6	7	8	9	10	11	14
2	2	4	6	8	10	12	14	16	18	20	24
3	2	4	6	8	10	12	14	16	18	20	24
4	1	2	3	4	5	6	7	8	9	10	13
5	1	1	2	2	3	3	4	4	5	5	5

te zien is.

$$\rho_1 = \frac{\lambda_1}{s_1 \cdot \mu_1} = \frac{14.1 \cdot 0.008}{s_1} < 1$$

$$\rho_2 = \frac{\lambda_2}{s_2 \cdot \mu_2} = \frac{18.2 \cdot 0.015}{s_2} < 1$$

$$\rho_3 = \frac{\lambda_3}{s_3 \cdot \mu_3} = \frac{18.2 \cdot 0.007}{s_3} < 1$$

$$\rho_4 = \frac{\lambda_4}{s_4 \cdot \mu_4} = \frac{9.1 \cdot 0.006}{s_4} < 1$$

$$\rho_5 = \frac{\lambda_5}{s_5 \cdot \mu_5} = \frac{6.3 \cdot 0.01}{s_5} < 1$$

Het aantal servers per station is dus als volgt:

$$s_1 > 14.1 \cdot 0.008 = 0.11$$

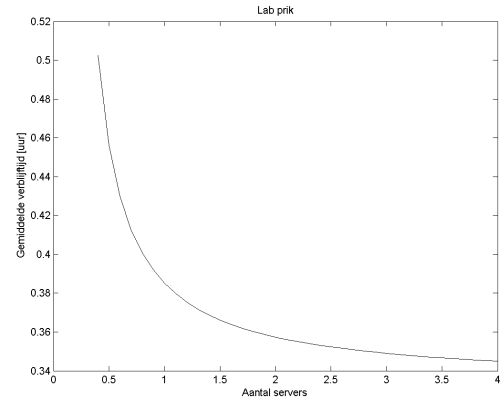
$$s_2 > 18.2 \cdot 0.015 = 0.13$$

$$s_3 > 18.2 \cdot 0.007 = 0.27$$

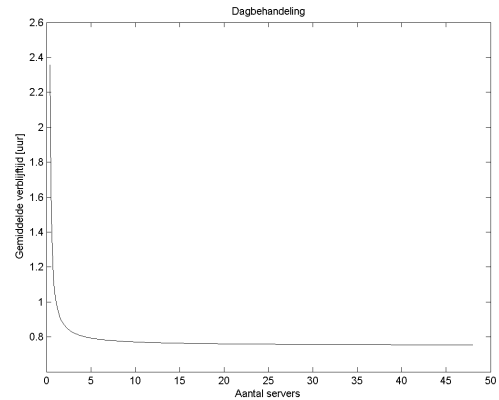
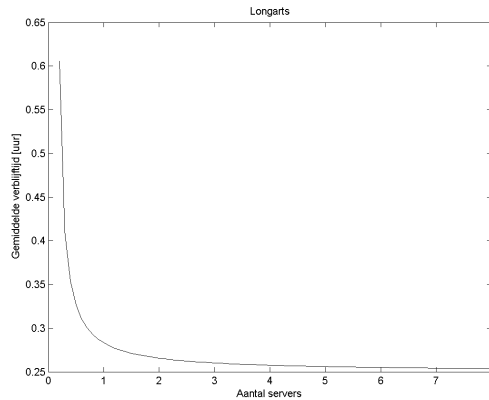
$$s_4 > 9.1 \cdot 0.006 = 0.05$$

$$s_5 > 6.3 \cdot 0.01 = 0.06$$

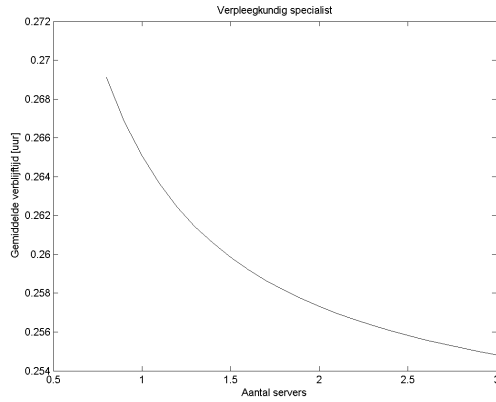
Figuur 39: De gemiddelde verblijftijd tegen het aantal servers bij de longarts met een behandelingsproces met elf verschillende types.



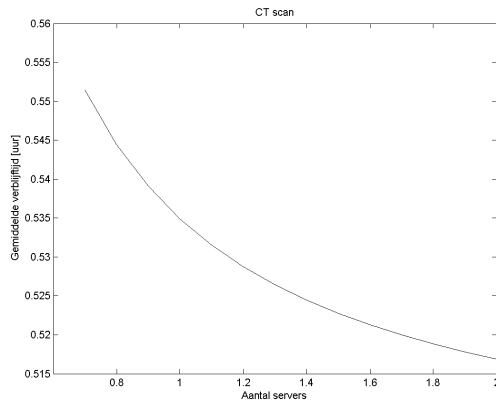
Figuur 40: De gemiddelde verblijftijd tegen het aantal servers bij het prik-lab met een behandelingsproces met elf verschillende types.



Figuur 41: De gemiddelde verblijftijd tegen het aantal servers bij de dagbehandeling met een behandelingsproces met elf verschillende types.



Figuur 42: De gemiddelde verblijftijd tegen het aantal servers bij de verpleegkundig specialist met een behandelingsproces met elf verschillende types.

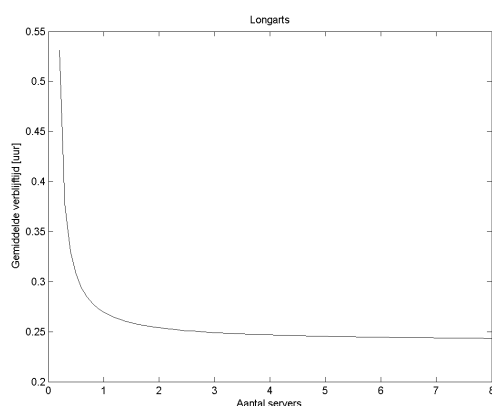


Figuur 43: De gemiddelde verblijftijd tegen het aantal servers bij de CT-scan met een behandelingsproces met elf verschillende types.

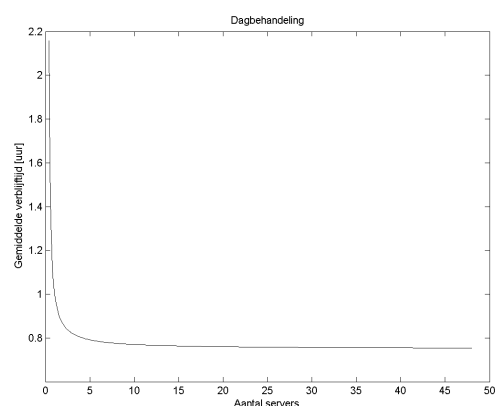
In Figuur 39, 40, 41, 42 en 43 is de gemiddelde verblijftijd tegen het aantal servers te zien bij respectievelijk de longarts, priklab, dagbehandeling, verpleegkundig specialist en de CT-scan. Wanneer er 1 server per station de gehele werkweek beschikbaar is voor de behandeling, is

de gemiddelde verblijftijd per station respectievelijk 17.0 minuten, 23.1 minuten, 61.8 minuten, 15.9 minuten en 32.1 minuten. Dit kan ook weer gedaan worden voor het behandelingsproces van maximaal zes maanden. Dan zijn er type patiënten 1 tot en met 6. Alleen is de aankomstintensiteit van type 6 patiënten groter, omdat type 7 tot en met 11 nu type 6 patiënten zijn. De kans dat een patiënt een type 6 patiënt is, is gelijk aan 0.0185. Hieruit volgt dat de aankomstintensiteit van type 6 patiënten gemiddeld $0.0185 \cdot 5 = 0.0923$ per week is. De gemiddelde aankomstintensiteiten zijn hieronder te zien.

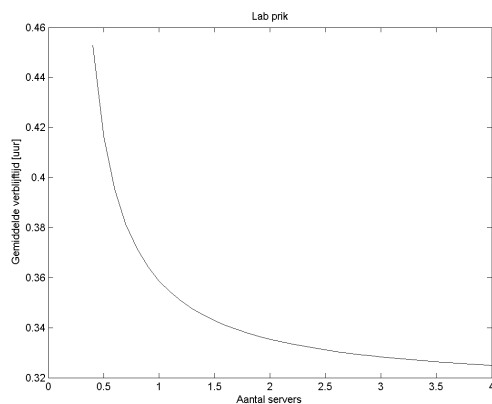
$$\begin{aligned}
 a_1 &= 2v(1) + 3v(2) + 4v(3) + 5v(4) \\
 &\quad + 6v(5) + 7v(6) \\
 &= 13.7 \\
 a_2 &= 2v(1) + 4v(2) + 6v(3) + 8v(4) \\
 &\quad + 10v(5) + 12v(6) \\
 &= 17.4 \\
 a_3 &= 2v(1) + 4v(2) + 6v(3) + 8v(4) \\
 &\quad + 10v(5) + 12v(6) \\
 &= 17.4 \\
 a_4 &= v(1) + 2v(2) + 3v(3) + 4v(4) \\
 &\quad + 5v(5) + 6v(6) \\
 &= 8.7 \\
 a_5 &= v(1) + v(2) + 2v(3) + 2v(4) \\
 &\quad + 3v(5) + 3v(6) \\
 &= 6
 \end{aligned}$$



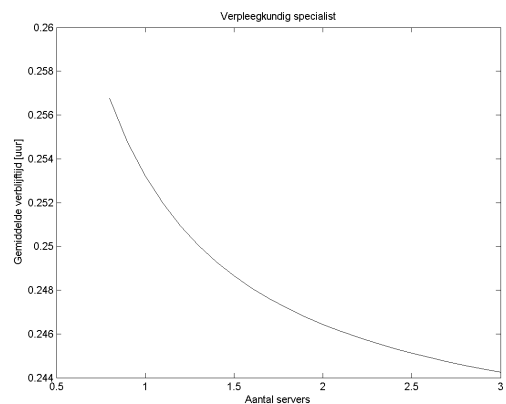
Figuur 44: De gemiddelde verblijftijd tegen het aantal servers bij de longarts met een behandelingsproces met zes verschillende types en maximaal een half jaar.



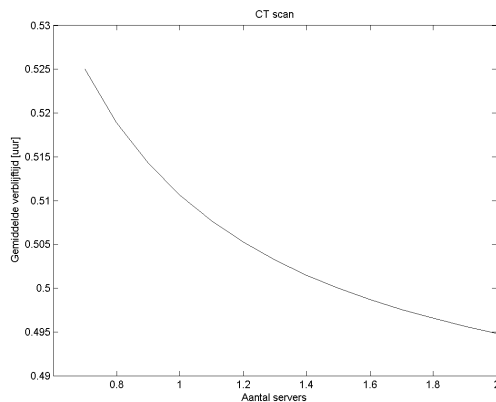
Figuur 46: De gemiddelde verblijftijd tegen het aantal servers bij de dagbehandeling met een behandelingsproces met zes verschillende types en maximaal een half jaar.



Figuur 45: De gemiddelde verblijftijd tegen het aantal servers bij het priklab met een behandelingsproces met zes verschillende types en maximaal een half jaar.



Figuur 47: De gemiddelde verblijftijd tegen het aantal servers bij de verpleegkundig specialist met een behandelingsproces met zes verschillende types en maximaal een half jaar.

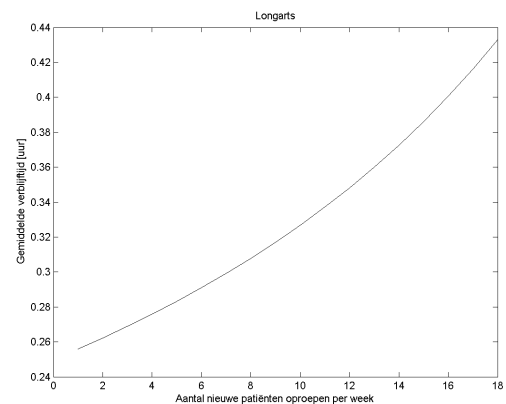


Figuur 48: De gemiddelde verblijftijd tegen het aantal servers bij de CT-scan met een behandlingsproces met zes verschillende types en maximaal een half jaar.

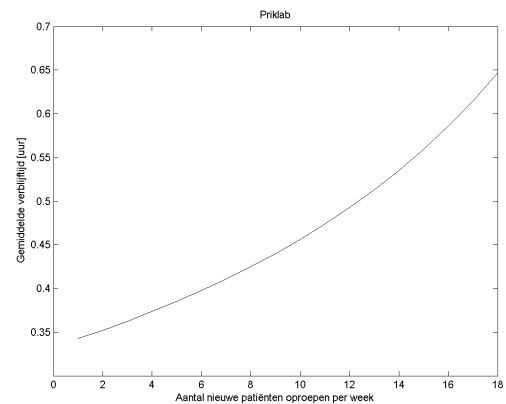
In Figuur 39, 40, 41, 42 en 43 is de gemiddelde verblijftijd tegen het aantal servers te zien bij respectievelijk de longarts, prikklab, dagbehandeling, verpleegkundig specialist en de CT-scan. Wanneer er 1 server per station de gehele werkweek beschikbaar is voor de behandeling, is de gemiddelde verblijftijd per station respectievelijk 16.8 minuten, 23.1 minuten, 61.6 minuten, 15.9 minuten en 32.1 minuten.

- Wanneer er nu een vaste capaciteit per station genomen wordt kan er bekeken worden wat voor invloed het verhogen van het aantal nieuwe patiënten per week is op de gemiddelde verblijftijd van een patiënt per station. Vervolgens wordt er bekeken wat voor invloed het verhogen van het aantal nieuwe patiënten per week is op de gemiddelde verblijftijd van een patiënt per station. Neem voor ieder station aan dat er een server een volledige werkweek beschikbaar is voor de behandeling van Nivolumab. Met behulp van MATLAB en Appendix G wordt er bepaald wat de maximale aankomstintensiteit van nieuwe patiënten is. Hieruit volgt dat λ maximaal 42 is voor de longarts, maximaal 37 is voor het prikklab, maximaal 18 is voor de dagbehandeling, maximaal 87 is voor de verpleegkundig

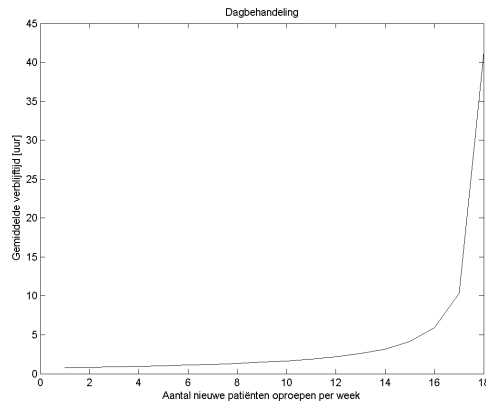
specialist en maximaal 76 is voor de CT-scan. Dus wanneer λ gevarieerd wordt van 1 tot en met 18 zijn alle stations stationair. Vervolgens wordt het aantal patiënten per week gevarieerd van 1 tot en met 19. Hieruit volgen de Figuren 49, 50, 51, 52 en 53 voor respectievelijk de longarts, het prikklab, de dagbehandeling, de verpleegkundig specialist en de CT-scan.



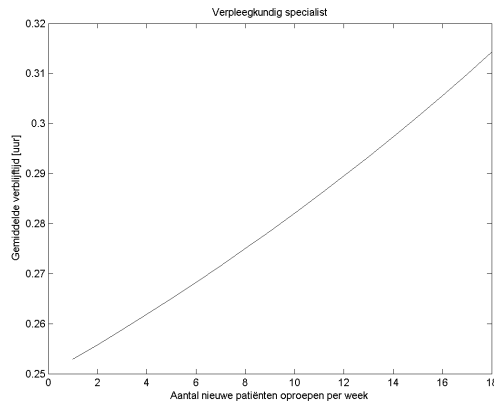
Figuur 49: De gemiddelde verblijftijd tegen het aantal nieuwe patiënten per week bij de longarts.



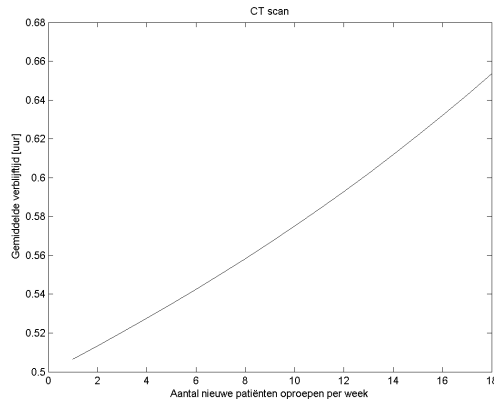
Figuur 50: De gemiddelde verblijftijd tegen het aantal nieuwe patiënten per week bij het prikklab.



Figuur 51: De gemiddelde verblijftijd tegen het aantal nieuwe patiënten per week bij de dagbehandeling.



Figuur 52: De gemiddelde verblijftijd tegen het aantal nieuwe patiënten per week bij de verpleegkundig specialist.



Figuur 53: De gemiddelde verblijftijd tegen het aantal nieuwe patiënten per week bij de CT-scan.

- Hierna kan een simulatie gemaakt worden. Aangezien een simulatie minder aannames nodig heeft, is dit realistischer. Verder kan dan de verdeling van de bedieningsduren ook deterministisch gemaakt worden in plaats van exponentieel. Als eerste wordt een trekking uit een exponentiële verdeling gedefinieerd. Om $X \sim \text{Exp}(\lambda)$ te trekken wordt de inverse methode gebruikt. Hiervoor wordt eerst U uit de uniforme verdeling $U[0, 1)$ getrokken. En vervolgens geldt $X = F^{-1}(U)$, waarbij:

$$F(x) = P(X \leq x) = 1 - e^{-\lambda x}$$

$$F^{-1}(y) = -\frac{\ln(1 - y)}{\lambda}$$

Vervolgens wordt ook trekken uit een Bernoulli verdeling gedefinieerd. Eerst wordt U uit de uniforme verdeling $U[0, 1)$ getrokken. Vervolgens geldt $F = 1 - \mu$. Wanneer $u > F$ is het resultaat 1 en anders 0. Deze verdeling wordt gebruikt om het type te bepalen van een nieuwe patiënt. Er geldt namelijk dat μ de kans is, dat de uitkomst 1 is. Om een simulatie te maken wordt er gebruik gemaakt van discrete event simulation. Eerst wordt er een simulatie gemaakt van een werkweek, waarbij ieder station 1 server heeft die 30 uur beschikbaar is. Op deze manier kan de simulatie geverifieerd worden met het analytische model. Eerst moet er natuurlijk een simulatie gemaakt worden. Er wordt gebruik gemaakt van de volgende gebeurtenissen:

1. Een aankomst bij de longarts
2. Een vertrek bij de longarts
3. Een aankomst bij het priklab
4. Een vertrek bij het priklab
5. Een aankomst bij de dagbehandeling

6. Een vertrek bij de dagbehandeling
7. Een aankomst bij de verpleegkundig specialist
8. Een vertrek bij de verpleegkundig specialist
9. Een aankomst bij de CT-scan
10. Een vertrek bij de CT-scan

Nu moet er bij iedere gebeurtenis vastgelegd worden wat hier gebeurd, dit is hieronder weergegeven per gebeurtenis.

1. Eerst wordt er gekeken of de patiënt die bij de longarts komt een nieuwe patiënt is. Wanneer dit het geval is moet het type van de patiënt bepaald worden. Dit wordt gedaan met de Bernoulli verdeling zoals hierboven genoemd is. Wanneer de patiënt geen nieuwe patiënt is, heeft deze patiënt al een type. Dus hoeft deze niet meer bepaald te worden. Aangezien de behandelingsweek begint bij de longarts wordt het aankomsttijdstip van de patiënt opgeslagen, zodat de verblijftijd van de patiënt in het systeem bepaald kan worden. Vervolgens wordt de bedieningsduur bepaald door een trekking te doen uit de exponentiële verdeling met $\mu = \frac{1}{15}$, aangezien de gemiddelde bedieningsduur 15 minuten is. Vervolgens wordt er een nieuwe aankomst van hetzelfde type patiënt als die is aangekomen gepland. Dit wordt gedaan door een trekking te doen uit de exponentiële verdeling met $\mu = \lambda$, waarbij λ de aankomstintensiteit van dat type patiënten per minuut is. De volgende aankomst van dit type is dus de tijd plus de uitkomst van de trekking uit de exponentiële verdeling. En de volgende aankomst bij de longarts is het minimum van alle geplande aankomsten van de verschillende type patiënten.

Vervolgens wordt er gekeken of de longarts bezet is. Wanneer dit niet het geval is kan de patiënt die is aangekomen meteen in behandeling bij de longarts. Nu is de longarts dus wel bezet en wordt er opgeslagen dat de longarts gedurende de bedieningsduur aan het werk is. Hierdoor kan later de bezettingsgraad van de longarts bepaald worden. De gebeurtenis 2 (Een vertrek bij de longarts) kan nu ook gepland worden. Dit is namelijk de tijd plus de bedieningsduur. Wanneer de longarts wel bezet is, sluit de patiënt achteraan in de wachtrij. De lengte van de wachtrij is nu verhoogd met 1.

2. Er wordt gekeken of dit de laatste fase van deze patiënt is. Wanneer dit het geval is verlaat de patiënt het systeem en wordt de verblijftijd van deze patiënt in deze behandelingsweek opgeslagen. Wanneer dit niet het geval is gaat de patiënt door naar het prikklab, dus de patiënt is een fase verder. Er wordt gekeken of er al meer patiënten onderweg zijn naar het prikklab. Wanneer dit niet het geval is, is gebeurtenis 3 (aankomst prikab) nu, er wordt dus geen rekening gehouden met looptijden. Wanneer er al wel een patiënt onderweg is naar het prikklab, wordt deze patiënt met zijn type en huidige tijd toegevoegd aan de patiënten die onderweg zijn naar het prikklab. Vervolgens is gebeurtenis 3 het minimum van de aankomsttijdstippen van de verschillende patiënten bij het prikklab.

Tenslotte wordt er gekeken of de wachtrij bij de longarts leeg is. Wanneer dit niet het geval is, wordt de eerste patiënt in de wachtrij behandeld bij de longarts. Er staat nu dus 1 patiënt minder in de wachtrij. Vervolgens wordt er een trekking uit de exponentiële verdeling gedaan met

$\mu = \frac{1}{15}$. Hieruit volgt de bedieningsduur van de patiënt. De tijd dat de longarts bezet is, is nu dus de tijd die hij al bezet is geweest plus de bedieningsduur. De nieuwe gebeurtenis 2 (vertrek longarts) is de huidige tijd plus de bediening. Wanneer er niemand in de wachtrij staat, is de nieuwe gebeurtenis 2 (vertrek longarts) oneindig. Aangezien er geen patiënt kan vertrekken, wanneer er niemand in bediening is.

3. Er wordt eerst gekeken of het prik-lab bezet is. Wanneer dit niet het geval is, kan de patiënt meteen in bediening. Hierdoor is het prik-lab nu wel bezet. Vervolgens wordt er een trekking uit de exponentiële verdeling gedaan met $\mu = \frac{1}{20}$. De tijd dat het prik-lab bezet is, is nu de huidige tijd dat het prik-lab bezet was plus de bedieningsduur. Gebeurtenis 4 (vertrek prik-lab) is de huidige tijd plus de bedieningsduur. Wanneer het prik-lab wel bezet is, sluit deze patiënt achter aan in de wachtrij. Het aantal patiënten in de wachtrij is nu dus met 1 verhoogd. Vervolgens wordt er gekeken of er nog meer patiënten onderweg naar het prik-lab zijn. Wanneer dit niet het geval is, wordt gebeurtenis 3 (aankomst prik-lab) oneindig, omdat er nog geen patiënten zijn die een aankomst bij het prik-lab hebben. Wanneer er wel patiënten onderweg zijn naar het prik-lab, wordt het minimum van deze aankomsten genomen. Dit minimum is de nieuwe gebeurtenis 3 (aankomst prik-lab). Het aantal patiënten dat onderweg is, is nu dus verminderd met 1.
4. De patiënt gaat weer een fase verder. Eerst wordt er gekeken of er nog meer patiënten onderweg naar de dagbehandeling zijn. Wanneer dit niet het geval is, is gebeurtenis 5

(aankomst dagbehandeling) de huidige tijd. Wanneer er wel meer patiënten onderweg zijn naar de dagbehandeling, wordt deze patiënt hieraan toegevoegd. Vervolgens wordt het minimum genomen over de aankomsten van de patiënten bij de dagbehandeling. Dit is dan gebeurtenis 5 (aankomst dagbehandeling). Vervolgens wordt er gekeken of de wachtrij van het prik-lab leeg is. Wanneer dit niet het geval is, gaat de eerste patiënt in de wachtrij in bediening. Er staat nu dus 1 patiënt minder in de wachtrij. Vervolgens wordt een trekking gedaan uit de exponentiële verdeling met $\mu = \frac{1}{20}$ om de bedieningsduur van deze patiënt te bepalen. De tijd dat het prik-lab nu bezet is, is de tijd dat het prik-lab al bezet is geweest plus de bedieningsduur. De nieuwe gebeurtenis 4 (vertrek prik-lab) is de huidige tijd plus de bedieningsduur. Wanneer er niemand in de wachtrij staat voor het prik-lab, is de nieuwe gebeurtenis 4 (vertrek prik-lab) oneindig, aangezien er niemand kan vertrekken als er niemand in bediening is.

5. Eerst wordt er gekeken of de dagbehandeling bezet is. Wanneer dit niet het geval is, kan de patiënt meteen in bediening. De dagbehandeling is nu dus wel bezet. Er kan nu een trekking gedaan worden uit de exponentiële verdeling met $\mu = \frac{1}{45}$ om de bedieningsduur van deze patiënt te bepalen. De tijd dat de dagbehandeling bezet is, is de tijd dat de dagbehandeling tot nu toe bezet is geweest plus de bedieningsduur. Gebeurtenis 6 (vertrek dagbehandeling) is de huidige tijd plus de bedieningsduur. Wanneer de dagbehandeling wel bezet is, sluit de patiënt achteraan in de wachtrij. Het aantal patiënten in de wachtrij is nu dus met 1 verhoogd. Vervolgens wordt

er gekeken of er nog meer patiënten onderweg zijn naar de dagbehandeling. Wanneer dit het geval is, wordt het minimum van deze aankomsten genomen en is gebeurtenis 5 (aankomst dagbehandeling) dit minimum. Het aantal patiënten dat onderweg is, wordt nu dus verminderd met 1. Wanneer er niet meer patiënten onderweg zijn naar de dagbehandeling, wordt gebeurtenis 5 (aankomst dagbehandeling) oneindig.

6. De patiënt die nu gaat vertrekken bij de dagbehandeling, is weer een fase verder wanneer dit gebeurt. Er wordt nu bekeken wat de volgende fase van de patiënt is. Wanneer dit de longarts is, wordt er opgeslagen dat deze patiënt de volgende behandelingsweek weer bij de longarts komt. Vervolgens wordt zijn verblijftijd in het systeem deze behandelingsweek bepaald. Wanneer de volgende fase de verpleegkundig specialist is, wordt er opgeslagen dat deze patiënt de volgende behandelingsweek weer bij de verpleegkundig specialist begint. Vervolgens wordt de verblijftijd van de patiënt bepaald van afgelopen behandelingsweek. Wanneer de patiënt als volgende fase niet de longarts of verpleegkundig specialist heeft, moet de patiënt naar de CT-scan. Deze CT-scan wordt nog wel deze behandelingsweek gemaakt. Dus wordt er gekeken of er meer patiënten onderweg zijn naar de CT-scan. Wanneer dit niet het geval is, is gebeurtenis 9 (aankomst CT-scan) de huidige tijd. Wanneer er wel meer patiënten onderweg zijn naar de CT-scan, wordt deze patiënt daaraan toegevoegd en wordt hier het minimum over de aankomsttijdstippen genomen. Dit minimum is de gebeurtenis 9 (aankomst CT-scan).

Tenslotte wordt er gekeken of de wachtrij van de dagbehandeling leeg is. Wanneer dit niet het geval is, kan de eerste patiënt in de wachtrij geholpen worden. Het aantal patiënten in de wachtrij voor de dagbehandeling is dan met 1 vermindert. Er wordt weer een trekking gedaan uit de exponentiële verdeling met $\mu = \frac{1}{45}$ om de bedieningsduur van deze patiënt te bepalen. Vervolgens wordt gebeurtenis 6 (vertrek dagbehandeling) gepland, dit is de huidige tijd plus de bedieningsduur. Wanneer er niemand in de wachtrij staat, kan er geen gebeurtenis 6 (vertrek dagbehandeling) gepland worden. Dus is deze gebeurtenis oneindig.

7. Wanneer er een patiënt aankomt bij de verpleegkundig specialist, wordt er eerst een nieuwe aankomst van hetzelfde type gepland. Dit wordt gedaan door een trekking te doen uit de exponentiële verdeling met $\mu = \lambda$, waarbij λ de aankomstintensiteit van dit type patiënt is. Wanneer dit gebeurd is, kan gebeurtenis 7 (aankomst verpleegkundig specialist) gepland worden. Dit wordt gedaan door het minimum te nemen over de geplande aankomsten van de verschillende type patiënten. Nu kan er gekeken worden of de verpleegkundig specialist bezet is. Wanneer dit het geval is, sluit de patiënt achteraan in de wachtrij. Het aantal wachtende patiënten wordt dus verhoogd met 1. Wanneer de verpleegkundig specialist niet bezet is, kan de patiënt meteen in bediening. Er wordt een trekking gedaan uit de exponentiële verdeling met $\mu = \frac{1}{15}$ om de bedieningsduur te bepalen. De tijd dat de verpleegkundig specialist bezet is wordt verhoogd met de bedieningsduur. Tenslotte wordt een gebeurtenis 8 (vertrek verpleeg-

kundig specialist) gepland, dit is namelijk de huidige tijd plus de bedieningsduur.

8. Wanneer een patiënt vertrekt bij de verpleegkundig specialist, is de patiënt weer een fase verder. Vervolgens wordt er gekeken of er meer patiënten onderweg zijn naar het priklab. Wanneer dit niet het geval is, kan een gebeurtenis 2 (aankomst priklab) gepland worden, dit is namelijk de huidige tijd. Wanneer er meer patiënten onderweg zijn naar het priklab, wordt deze patiënt hierbij gevoegd. Hier wordt vervolgens het minimum over de geplande aankomsttijden genomen. De uitkomst hiervan is de nieuwe gebeurtenis 3 (aankomst priklab). Vervolgens wordt er gekeken of de wachtrij bij de verpleegkundig specialist leeg is. Wanneer dit niet het geval is, wordt de eerste patiënt van de wachtrij geholpen. Er staat nu dus 1 patiënt minder in de wachtrij. Om de bedieningsduur te bepalen, wordt er nu een trekking uit de exponentiële verdeling met $\mu = \frac{1}{15}$. De tijd dat de verpleegkundig specialist aan het werk is geweest, wordt verhoogd met de bedieningsduur. Verder wordt gebeurtenis 8 (vertrek verpleegkundig specialist) gepland, dit is de huidige tijd plus de bedieningsduur. Wanneer er niemand in de wachtrij staat, is gebeurtenis 8 oneindig.
9. Wanneer een patiënt aankomt bij de CT-scan, wordt ook hier eerst gekeken of de CT-scan bezet is. Wanneer dit niet het geval is, kan de patiënt meteen in bediening. Er wordt voor de bedieningsduur een trekking uit de exponentiële verdeling met $\mu = \frac{1}{30}$ gedaan. De tijd dat de CT-scan bezet is wordt nu verhoogd met de bedieningsduur.

De gebeurtenis 10 (vertrek CT-scan) kan gepland worden, dit is namelijk de huidige tijd plus de bedieningsduur. Wanneer de CT-scan wel bezet is, sluit de patiënt achteraan in de wachtrij. Het aantal patiënten in de wachtrij wordt nu met 1 verhoogd. Vervolgens wordt er gekeken of er nog meer patiënten onderweg waren naar de CT-scan. Wanneer dit het geval is, kan het minimum over de aankomsttijdstippen genomen worden. Dit minimum is gebeurtenis 9 (aankomst CT-scan). Wanneer er verder geen patiënten onderweg zijn naar de CT-scan, komt er geen nieuwe patiënt bij de CT-scan aan. Dus is gebeurtenis 9 (aankomst CT-scan) oneindig.

10. Wanneer een patiënt vertrekt bij de CT-scan, gaat de patiënt weer een fase verder. Deze patiënt wordt volgende week weer opgeroepen bij de longarts. Vervolgens kan de verblijftijd van de patiënt bepaald worden. Wanneer er nog patiënten in de wachtrij voor de CT-scan staan, kan de eerste in bediening gaan. Er staat dan dus 1 patiënt minder in de wachtrij. Er kan nu een trekking uit de exponentiële verdeling gedaan worden met $\mu = \frac{1}{30}$ om de bedieningsduur te bepalen. De tijd dat de CT-scan bezet is, wordt verhoogd met de bedieningsduur. Vervolgens is gebeurtenis 10 (vertrek CT-scan) de huidige tijd plus de bedieningsduur. Wanneer de wachtrij leeg is, kan er geen patiënt bediend worden. Hierdoor kan er ook geen patiënt vertrekken bij de CT-scan, dus gebeurtenis 10 (vertrek CT-scan) is oneindig.

De patiënten die de volgende behandelingsweek terug komen, zijn de startwaarden van de volgende behandelingsweek. Er komen dan natuurlijk ook weer

5 nieuwe patiënten bij. Dit wordt 41 weken achter elkaar gesimuleerd. In de eerste week komt van ieder type en fase 5 keer de kans op dit type. Dus de eerste week is het systeem nog niet in evenwicht, daarom wordt de eerste week niet meegenomen. Van deze 99 andere weken wordt een gemiddelde genomen van de bezetting van de verschillende stations per week. Ook wordt een gemiddelde genomen van de verblijftijd van een patiënt in een behandelingsweek. Hierdoor wordt natuurlijk wel onderscheidt gemaakt tussen de verschillende behandelingsweken. Er is namelijk een behandelingsweek met een intake bij de longarts, een behandelingsweek met een controle bij de longarts, een behandelingsweek met een controle bij de verpleegkundig specialist en tenslotte is er ook nog een behandelingsweek waarbij de patiënt een CT-scan moet laten maken. Er worden 100 replicaties gemaakt van deze simulatie van 100 weken. Hierdoor kunnen er 95% betrouwbaarheidsintervallen gemaakt worden voor de bezetting van de verschillende stations en de verblijftijd van de patiënten voor de verschillende behandelingsweken. Voor de verificatie van de simulatie wordt er gekeken of de analytische waarden in de betrouwbaarheidsintervallen van de simulatie liggen.

Tabel 24: Een tabel met de verblijftijden van patiënten in minuten voor de verificatie van de simulatie, waarbij voor ieder station een werkweek van 30 uur gebruikt is.

Station	Analytisch	Simulatie
Longarts	17.0	[16.6 17.0]
Priklab	25.0	[23.8 24.5]
Dagbehandeling	72	[71.5 72.6]
Verpleegkundig specialist	18.7	[15.7 16.2]
CT-scan	37.8	[32.7 33.9]

Tabel 25: Een tabel met de verblijftijden van patiënten in minuten voor de verificatie van de simulatie, waarbij voor ieder station een werkweek van 40 uur gebruikt is.

Station	Analytisch	Simulatie
Longarts	16.5	[16.3 16.7]
Priklab	23.6	[23.0 23.6]
Dagbehandeling	68.3	[65.5 66.4]
Verpleegkundig specialist	15.9	[15.7 16.1]
CT-scan	32.5	[32.1 32.9]

Wanneer er naar Tabel 24 en 25 gekeken wordt valt het op dat de meeste van de analytisch berekende waarden in de betrouwbaarheidsintervallen liggen. De stations waarbij dit niet altijd het geval is, zijn het priklab en de dagbehandeling. Hierdoor zou het mogelijk zijn dat er in de simulatie misschien patiënten tussen het vertrek bij de longarts of de verpleegkundig specialist en de aankomst bij het priklab verdwijnen. Wanneer dit echter gecontroleerd wordt blijkt dit niet het geval te zijn.

Nu kan dit verschil verklaard worden door het feit dat in de simulatie de aankomsten van de patiënten bij de longarts of verpleegkundig specialist zijn. Dus deze aankomsten komen overeen met de aankomsten van een losse wachtrij. Maar bij de dagbehandeling zitten er al de stations longarts of verpleegkundig specialist en priklab tussen, dus de aankomstintensiteit is wat lager bij de dagbehandeling. Dit is omdat er sommige patiënten halverwege de behandelingsweek zijn op het moment dat de werkweek van de stations voorbij is. Wanneer de gemiddelde aankomstintensiteit uitgerekend wordt bij een werkweek van 30 uur, komt hieruit dat de gemiddelde aankomstintensiteit 15 patiënten per week is. Wanneer dit aantal als input een los wachtrijmodel van de dagbehandeling wordt gebruikt, komt hieruit dat de gemiddelde verblijftijd van een patiënt 72 minuten is. Dit

getal ligt wel in het betrouwbaarheidsinterval van de simulatie.

Dit geldt ook in mindere mate voor het prikklab, daarom ligt deze waarde boven in het betrouwbaarheidsinterval van de simulatie. Verder heeft dit ook invloed op de CT-scan, maar bij de CT-scan zijn hele lage wachttijden waardoor dit niet opvalt.

De simulatie moet dus nog wel aangepast worden, zodat het niet mogelijk is dat patiënten de behandelingsweek niet volbrengen. Dit kan gedaan worden door de patiënten niet tot het laatste moment van de werkweek op te roepen bij de longarts of verpleegkundig specialist. Er moet namelijk nog minimaal een resterende tijd zijn van de som van de bedieningsduren van de resterende behandelingen van de behandelingsweek. De patiënten die halverwege de behandelingsweek moeten stoppen, verdwijnen ook uit het systeem. Dus komen de volgende week niet meer terug. Dit verklaard waardoor de analytische waarden allemaal aan de bovenkant van de betrouwbaarheidsintervallen liggen.

- Vervolgens ben ik weer naar het Antoni van Leeuwenhoek ziekenhuis geweest met mijn verkregen resultaten over de capaciteiten. Hieruit bleek dat de kans om na drie maanden nog in het systeem te blijven niet 20% is maar 65%. Hierdoor wordt natuurlijk de kans op iedere type patiënt anders. Dus dit moet eerst aangepast worden. Wanneer er weer gebruik wordt gemaakt van de geometrische verdeling geldt er nu $p + (1 - p)p = 0.35$ in plaats van $p + (1 - p)p = 0.8$. Hieruit volgt dat $p=0.2$ of $p=1.8$, aangezien p een kans is moet p kleiner dan één zijn. Dus dit geeft $p=0.2$ en de volgende formule voor de kans

$$y = \begin{cases} 0.2(0.8)^{n-1} & \text{voor } n < 11 \\ 0.8^{n-1} & \text{voor } n = 11 \end{cases}$$

Met behulp van deze functie kunnen de aankomstintensiteiten van de verschil-

lende type patiënten bepaald worden. De volgende berekeningen zijn gemaakt met behulp van MATLAB, zoals te zien is in Appendix F

- De kans dat een patiënt van type 1 is, is de kans dat de patiënt het systeem bij de eerste mogelijkheid verlaat. Hieruit volgt $y = 0.2(0.8)^0 = 0.2$. Dus de aankomstintensiteit van type 1 patiënten is gemiddeld $5 \cdot 0.2 = 1$ per week.
- De kans dat een patiënt van type 2 is, is de kans dat de patiënt het systeem bij de tweede mogelijkheid verlaat. Hieruit volgt $y = 0.2(0.8)^1 = 0.16$. Dus de aankomstintensiteit van type 2 patiënten is gemiddeld $5 \cdot 0.16 = 0.8$ per week.
- De kans dat een patiënt van type 3 is, is de kans dat de patiënt het systeem bij de derde mogelijkheid verlaat. Hieruit volgt $y = 0.2(0.8)^2 = 0.128$. Dus de aankomstintensiteit van type 3 patiënten is gemiddeld $5 \cdot 0.128 = 0.64$ per week.
- De kans dat een patiënt van type 4 is, is de kans dat de patiënt het systeem bij de vierde mogelijkheid verlaat. Hieruit volgt $y = 0.2(0.8)^3 = 0.1024$. Dus de aankomstintensiteit van type 4 patiënten is gemiddeld $5 \cdot 0.1024 = 0.512$ per week.
- De kans dat een patiënt van type 5 is, is de kans dat de patiënt het systeem bij de vijfde mogelijkheid verlaat. Hieruit volgt $y = 0.2(0.8)^4 = 0.0819$. Dus de aankomstintensiteit van type 5 patiënten is gemiddeld $5 \cdot 0.0819 = 0.4096$ per week.
- De kans dat een patiënt van type 6 is, is de kans dat de patiënt het systeem bij de zesde mogelijkheid verlaat. Hieruit volgt $y = 0.2(0.8)^5 = 0.0655$. Dus de aankomstintensiteit van type 6 patiënten is gemiddeld

$5 \cdot 0.0655 = 0.3277$ per week.

- De kans dat een patiënt van type 7 is, is de kans dat de patiënt het systeem bij de zevende mogelijkheid verlaat. Hieruit volgt $y = 0.2(0.8)^6 = 0.0524$. Dus de aankomstintensiteit van type 7 patiënten is gemiddeld $5 \cdot 0.0524 = 0.2621$ per week.
- De kans dat een patiënt van type 8 is, is de kans dat de patiënt het systeem bij de achtste mogelijkheid verlaat. Hieruit volgt $y = 0.2(0.8)^7 = 0.0419$. Dus de aankomstintensiteit van type 8 patiënten is gemiddeld $5 \cdot 0.0419 = 0.2097$ per week.
- De kans dat een patiënt van type 9 is, is de kans dat de patiënt het systeem bij de negende mogelijkheid verlaat. Hieruit volgt $y = 0.2(0.8)^8 = 0.0336$. Dus de aankomstintensiteit van type 9 patiënten is gemiddeld $5 \cdot 0.0336 = 0.1678$ per week.
- De kans dat een patiënt van type 10 is, is de kans dat de patiënt het systeem bij de tiende mogelijkheid verlaat. Hieruit volgt $y = 0.2(0.8)^9 = 0.0268$. Dus de aankomstintensiteit van type 10 patiënten is gemiddeld $5 \cdot 0.0268 = 0.1342$ per week.
- De kans dat een patiënt van type 11 is, is de kans dat de patiënt het systeem bij de elfde mogelijkheid verlaat. Dit is dus de kans dat de patiënt de eerste tien keer in het systeem bleef. Dit is dus gelijk aan $0.8^{10} = 0.1074$. Dus de aankomstintensiteit van type f patiënten is gemiddeld $5 \cdot 0.1074 = 0.5369$ per week.

Vervolgens wordt Tabel 23 gebruikt om de aankomstintensiteiten te bepalen per station. De stromingsvergelijkingen zijn namelijk niet veranderd, doordat de kans

op de types veranderd is.

$$\begin{aligned}
 a_1 &= 2\nu(1) + 3\nu(2) + 4\nu(3) + 5\nu(4) \\
 &\quad + 6\nu(5) + 7\nu(6) + 8\nu(7) + 9\nu(8) \\
 &\quad + 10\nu(9) + 11\nu(10) + 14\nu(11) \\
 &= 28.9 \\
 a_2 &= 2\nu(1) + 4\nu(2) + 6\nu(3) + 8\nu(4) \\
 &\quad + 10\nu(5) + 12\nu(6) + 14\nu(7) + 16\nu(8) \\
 &\quad + 18\nu(9) + 20\nu(10) + 24\nu(11) \\
 &= 46.8 \\
 a_3 &= 2\nu(1) + 4\nu(2) + 6\nu(3) + 8\nu(4) \\
 &\quad + 10\nu(5) + 12\nu(6) + 14\nu(7) \\
 &\quad + 16\nu(8) + 18\nu(9) + 20\nu(10) \\
 &\quad + 24\nu(11) = 46.8 \\
 a_4 &= \nu(1) + 2\nu(2) + 3\nu(3) + 4\nu(4) \\
 &\quad + 5\nu(5) + 6\nu(6) + 7\nu(7) + 8\nu(8) \\
 &\quad + 9\nu(9) + 10\nu(10) + 13\nu(11) \\
 &= 23.9 \\
 a_5 &= \nu(1) + \nu(2) + 2\nu(3) + 2\nu(4) \\
 &\quad + 3\nu(5) + 3\nu(6) + 4\nu(7) + 4\nu(8) \\
 &\quad + 5\nu(9) + 5\nu(10) + 5\nu(11) \\
 &= 12.4
 \end{aligned}$$

Met deze aankomstintensiteiten kan bepaald worden wanneer ieder station in evenwicht is. Hieruit volgt een minimaal aantal servers per station, zoals hieronder te zien is.

$$\begin{aligned}
 \rho_1 &= \frac{\lambda_1}{s_1 \cdot \mu_1} = \frac{28.9 \cdot 0.008}{s_1} < 1 \\
 \rho_2 &= \frac{\lambda_2}{s_2 \cdot \mu_2} = \frac{46.8 \cdot 0.015}{s_2} < 1 \\
 \rho_3 &= \frac{\lambda_3}{s_3 \cdot \mu_3} = \frac{46.8 \cdot 0.007}{s_3} < 1 \\
 \rho_4 &= \frac{\lambda_4}{s_4 \cdot \mu_4} = \frac{23.9 \cdot 0.006}{s_4} < 1 \\
 \rho_5 &= \frac{\lambda_5}{s_5 \cdot \mu_5} = \frac{12.4 \cdot 0.01}{s_5} < 1
 \end{aligned}$$

Het aantal servers per station is dus als

volgt:

$$s_1 > 28.9 \cdot 0.008 = 0.24$$

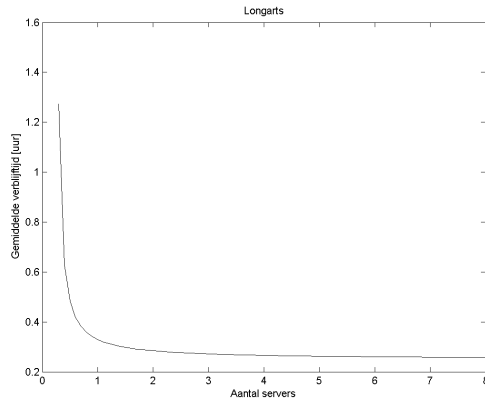
$$s_2 > 46.8 \cdot 0.015 = 0.35$$

$$s_3 > 46.8 \cdot 0.007 = 0.70$$

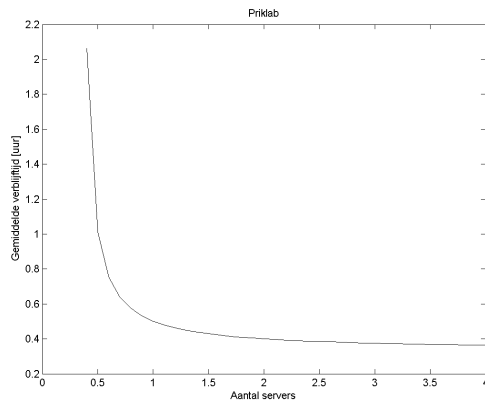
$$s_4 > 23.9 \cdot 0.006 = 0.15$$

$$s_5 > 12.4 \cdot 0.01 = 0.13$$

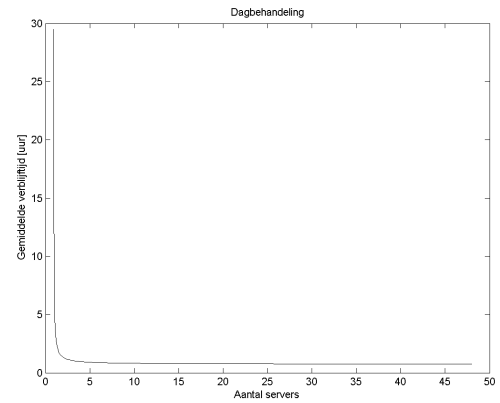
Hieruit volgt dus dat er een longarts minimaal 7.2 uur, het priklab 15.6 uur, de dagbehandeling 35.1 uur, de verpleegkundig specialist 6.0 uur en de CT-scan 6.2 uur een server beschikbaar moeten hebben.



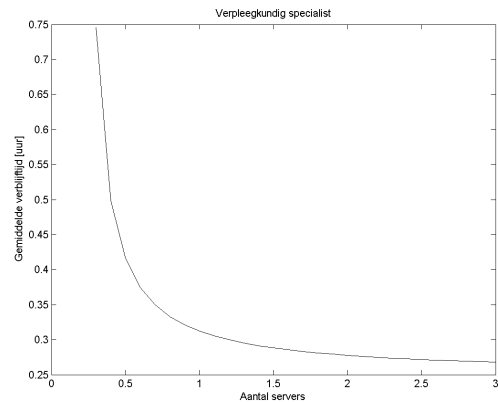
Figuur 54: De gemiddelde verblijftijd tegen het aantal servers bij de longarts met een behandelingsproces met elf verschillende types.



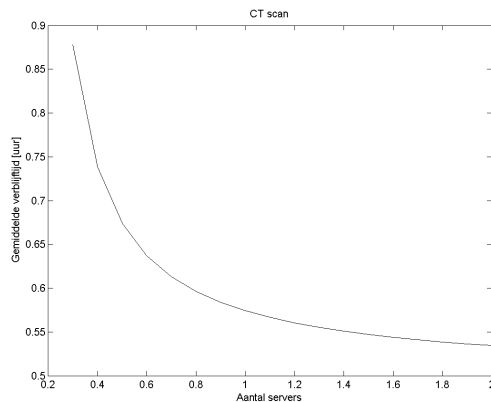
Figuur 55: De gemiddelde verblijftijd tegen het aantal servers bij het priklab met een behandelingsproces met elf verschillende types.



Figuur 56: De gemiddelde verblijftijd tegen het aantal servers bij de dagbehandeling met een behandelingsproces met elf verschillende types.



Figuur 57: De gemiddelde verblijftijd tegen het aantal servers bij de verpleegkundig specialist met een behandelingsproces met elf verschillende types.



Figuur 58: De gemiddelde verblijftijd tegen het aantal servers bij de CT-scan met een behandlingsproces met elf verschillende types.

In Figuur 54, 55, 56, 57 en 58 is de gemiddelde verblijftijd tegen het aantal servers te zien bij respectievelijk de longarts, prikklab, dagbehandeling, verpleegkundig specialist en de CT-scan. Wanneer er 1 server per station de gehele werkweek beschikbaar is voor de behandeling, is de gemiddelde verblijftijd per station respectievelijk 19.8 minuten, 30.6 minuten, 204.2 minuten, 18.0 minuten en 35 minuten.

- Verder heb ik de informatie gekregen dat er iedere week ongeveer 25 patiënten bij de verpleegkundig specialist komen. Wanneer er voor iedere patiënt 15 minuten nodig is, betekent dit dat er ongeveer 6.3 uur een verpleegkundig specialist aanwezig moet zijn. Uit analytisch berekende waarde is 6.0 uur. Hieruit kan geconcludeerd worden dat deze kans op verschillende types aannemelijk is. Het verschil is namelijk 1 patiënt per week.
- Nu moet ook in de simulatie de kans op de verschillende types aangepast worden. Hierna kan opnieuw de verificatie gedaan worden. Hiervoor moeten de stations natuurlijk wel stationair zijn, hierdoor wordt er gekozen voor een werkweek van 50 uur.

Tabel 26: Een tabel met de verblijftijden van patiënten in minuten voor de verificatie van de simulatie, waarbij voor ieder station een werkweek van 50 uur gebruikt is.

Station	Analytisch	Simulatie
Longarts	17.5	[16.4 16.7]
Priklab	29.1	[24.1 24.7]
Dagbehandeling	150.8	[77 78.3]
Verpleegkundig specialist	17.0	[15.9 16.2]
CT-scan	34.2	[31.5 32.6]

Wanneer de simulatie maar 2 weken na de opwarm periode gedaan wordt, zouden er minder patiënten wegvallen doordat ze de handlingsweek niet hebben afgemaakt. Hierdoor zouden de analytische waarden dicht bij de betrouwbaarheidsintervallen moeten liggen. Dit is te zien in Tabel 27

Tabel 27: Een tabel met de verblijftijden van patiënten in minuten voor de verificatie van de simulatie, waarbij voor ieder station een werkweek van 50 uur gebruikt is en replicaties van twee weken.

Station	Analytisch	Simulatie
Longarts	17.5	[16.2 18.1]
Priklab	29.1	[26 29.4]
Dagbehandeling	150.8	[117 121.4]
Verpleegkundig specialist	17.0	[15.9 17.8]
CT-scan	34.2	[32.0 37.1]

Wanneer nu de longarts en verpleegkundig specialist maar 30 uur beschikbaar zijn en de rest van de stations 50 uur, zou dit dus wel ongeveer overeen moeten komen met de analytische waardes. De rest van de stations is dan natuurlijk aan het einde van de week nog 20 uur langer open. De longarts en verpleegkundig specialist moeten vergeleken worden met de analytische waardes bij een werkweek van 30 uur. Het prikklab, dagbehandeling en de CT-scan met een werkweek van

50 uur. Dit is natuurlijk niet precies hetzelfde als de analytische waarden van het station dat 50 uur open is, aangezien er in het begin nog geen patiënten aan kunnen komen. De patiënten moeten namelijk eerst nog naar de longarts of verpleegkundig specialist. In dit geval zijn er nog wel patiënten die verdwijnen, maar het zijn er veel minder. Hierdoor zouden de waarden beter overeen moeten komen. Dit is te zien in Tabel 28. Zoals te zien is zijn de analytische waarden van de longarts en verpleegkundig specialist wat hoger dan de simulatie, dit komt omdat er wel wat patiënten wegvallen omdat ze de behandelweek niet afmaken. Verder is te zien dat de analytische waarden van het priklab en de dagbehandeling wat lager zijn. Dit komt omdat er bij de simulatie in de laatste uren natuurlijk minder patiënten komen dan in de uren dat de longarts en de verpleegkundig specialist ook open zijn. Bij de CT-scan komen de waarden wel overeen, dit komt omdat de aankomstintensiteit bij de CT-scan niet zo hoog is.

Tabel 28: Een tabel met de verblijftijden van patiënten in minuten voor de verificatie van de simulatie.

Station	Analytisch	Simulatie
Longarts	19.8	[17.8 18.1]
Priklab	29.1	[30.5 31.8]
Dagbehandeling	150.8	[155.6 160.8]
Verpleegkundig specialist	18.7	[17.0 17.5]
CT-scan	34.2	[33.9 35.1]

Vervolgens wordt ook het aantal patiënten die aankomen bij ieder station bekeken. Ook is ervoor gekozen om na 40 uur geen patiënten meer aan te laten komen bij de longarts en de verpleegkundig specialist, maar de andere stations open te houden totdat alle patiënten de behandelingsweek voltooid hebben. De resultaten zijn te zien in Tabel 29. Zoals hierin te

zien is zijn de waarden van de aankomstintensiteiten bij de simulatie lager dan de analytisch berekende waarden. Dit is wel opmerkelijk. Hierdoor zijn ook de verblijftijden per stations bij de simulatie lager dan de analytische waarden. Hieruit blijkt dat er een fout zit in de simulatie. Nadat deze fout is opgelost, moet de simulatie opnieuw geverifieerd worden. Dit is te zien in Tabel 30. Er is hiervoor een simulatie van 126 weken gemaakt, waarbij de eerste 26 weken verwijderd worden als opwarm periode. Verder zijn er 41 replicaties gemaakt, zodat er 40 vrijheidsgraden zijn. Zoals hier te zien is liggen alle analytisch berekende aankomstintensiteiten in de betrouwbaarheidsintervallen van de simulatie. De verblijftijd ligt alleen bij de dagbehandeling in het betrouwbaarheidsinterval. Bij de andere stations wijkt deze minder dan vijf minuten van het betrouwbaarheidsinterval af. Dit is gedeeltelijk te verklaren doordat de stations niet precies 40 uur geopend zijn, omdat er nog een uitloop meegenomen wordt. Zo bezoeken de patiënten wel alle stations van de behandelingsweek binnen een week.

- Uit deze verificatie kan geconcludeerd worden dat de simulatie goed werkt. Hierdoor kan het feit dat een intake gemiddeld 45 minuten duurt toegevoegd worden. Verder kunnen nu de werkelijke openingstijden van de verschillende stations in de simulatie toegevoegd worden. Dus nu moet er gekeken worden welke stations wanneer open zijn en wanneer de patiënten met de behandeling van Nivolumab hier terecht kunnen. Zo kunnen de patiënten maar tot 14:30 uur naar de verpleegkundig specialist of longarts komen, want zo is de kans groter dat ze nog meer stations van het behandelingsproces op deze dag kunnen bezoeken. Terwijl het voor andere patiënten wel mogelijk is om later bij de verpleegkundig specialist of longarts te komen.
Longarts: 8:30-14:30 uur

Priklab: 7:30-17:30 uur

Dagbehandeling: 8:00-16:00 uur

Verpleegkundig specialist: 8:30-14:30 uur

CT-scan: 8:00-16:00 uur

Verder zijn alle stations vijf dagen per week geopend, dus gewoon maandag tot en met vrijdag, behalve de CT-scan. De CT-scan is namelijk ook nog op zaterdag geopend.

De week begint op maandag en alle patiënten beginnen bij de longarts of bij de verpleegkundig specialist. Dus alle stations die maandag eerder open zijn krijgen geen patiënten van de behandeling van Nivolumab. Dus deze capaciteit wordt niet meegenomen. Hieruit volgt dat de week begint op maandag om 8:30 uur. Dit wordt dus uur 0 van de simulatie. Vervolgens is dus 9:30 uur het tweede uur van de simulatie enzovoorts. De openingstijden waarin patiënten van de behandeling met Nivolumab de stations kunnen bezoeken zijn te zien in Tabel 31. Aan het einde van de week blijven de stations open totdat alle patiënten de behandelingsweek afmaken. Het kan dus zijn dat het priklab en de dagbehandeling op vrijdag over moeten werken om dit voor elkaar te krijgen. Er wordt bijgehouden hoe lang deze verschillende stations over moeten werken, zodat er gekeken kan worden om dit te verminderen. Dit kan later gedaan worden door bijvoorbeeld patiënten op vrijdag niet tot 14:30 uur op te roepen bij de verpleegkundig specialist en longarts, maar deze stations op vrijdag eerder te sluiten voor patiënten met de behandeling van Nivolumab. Uit deze openingstijden van de verschillende stations blijkt dat de werkweken per station aangepast moeten worden. Hieronder staan de nieuwe werkweken weergegeven.

longarts: 30 uur per week

Priklab: 46.5 uur per week

Dagbehandeling: 40 uur per week

Verpleegkundig specialist: 30 uur per week

CT-scan: 48 uur per week

In het analytische model worden deze werkweken ook aangepast.

- Nu kan dus met behulp van het analytische model voor het huidige behandelingsproces en het behandelingsproces van maximaal een half jaar bekeken worden hoeveel servers er per station nodig zijn, zodat het systeem in evenwicht is. Dit blijkt voor beide behandelingsprocessen één server per station te zijn. Vervolgens kan er gekeken worden of er met deze capaciteit meer nieuwe patiënten per week opgeroepen kunnen worden. Ook dit wordt gedaan met het analytische model. Hiervoor moet natuurlijk nog wel gelden dat alle stations in evenwicht zijn. Hieruit blijkt dat bij het huidige behandelingsproces er niet meer patiënten per week opgeroepen kunnen worden. Maar bij het behandelingsproces van maximaal een half jaar kunnen er zeven patiënten per week opgeroepen worden.
- Deze resultaten worden gebruikt voor de simulatie. De simulatie is gemaakt in PYTHON en is te zien in Appendix H. Er is gebruik gemaakt van een deterministische verdeling voor de bedieningsduren in plaats van een exponentiële verdeling. Dit is namelijk realistischer. Verder worden patiënten niet volgens een Poisson proces opgeroepen, maar met een deterministisch tussentijd. Vervolgens wordt er ook nog gekeken naar een strategie om patiënten dynamisch op te roepen. Dit wordt gedaan door de maximale aankomstintensiteit bij de dagbehandeling bij één server te bepalen, aangezien de dagbehandeling het knelpunt is. Vervolgens is het aantal patiënten dat opgeroepen wordt: de maximale aankomstintensiteit bij de dagbehandeling minus de terugkomende patiënten plus het aantal patiënten die de maximale duur van het behandelingsproces erop hebben zitten. De resultaten zijn te zien in hoofdstuk 5.

Tabel 29: Een tabel met prestatiegegevens voor de verificatie van de simulatie, waarbij er nog een fout zit in de simulatie

Station	Prestatiemaat	Analytisch	Simulatie
Longarts	Verblijftijd [min]	18.3	[17.4 18.0]
Longarts	Aankomstintensiteit [per week]	28.9	[19.2 21.2]
Priklab	Verblijftijd [min]	32.8	[27.5 29.1]
Priklab	Werkweek [uur]	2400	[2347 2366]
Priklab	Aankomstintensiteit [per week]	46.8	[30.1 33.4]
Dagbehandeling	Verblijftijd [min]	366.2	[141 149]
Dagbehandeling	Werkweek [min]	2400	[2467 2513]
Dagbehandeling	Aankomstintensiteit [per week]	46.8	[30.1 33.4]
Verpleegkundig specialist	Verblijftijd [min]	17.6	[16.4 17.3]
Verpleegkundig specialist	Aankomstintensiteit [per week]	23.9	[14.0 15.6]
CT-scan	Verblijftijd [min]	35.5	[33.0 35.4]
CT-scan	Werkweek [min]	2400	[1950 2071]
CT-scan	Aankomstintensiteit [per week]	12.4	[6.6 7.4]

Tabel 30: Een tabel met prestatiegegevens voor de verificatie van de simulatie.

Station	Prestatiemaat	Analytisch	Simulatie
Longarts	Verblijftijd [min]	18.3	[19.1 20.1]
Longarts	Aankomstintensiteit [per week]	28.9	[27.8 30.3]
Priklab	Verblijftijd [min]	32.8	[37.5 43.7]
Priklab	Werkweek [uur]	2400	[2394 2403]
Priklab	Aankomstintensiteit [per week]	46.8	[45.7 50.2]
Dagbehandeling	Verblijftijd [min]	366.2	[354.4 376.2]
Dagbehandeling	Werkweek [min]	2400	[2766 2884]
Dagbehandeling	Aankomstintensiteit [per week]	46.8	[45.7 50.2]
Verpleegkundig specialist	Verblijftijd [min]	17.6	[18.4 19.1]
Verpleegkundig specialist	Aankomstintensiteit [per week]	23.9	[22.8 25.2]
CT-scan	Verblijftijd [min]	35.5	[36.4 37.4]
CT-scan	Werkweek [min]	2400	[2459 2581]
CT-scan	Aankomstintensiteit [per week]	12.4	[11.9 13.0]

Tabel 31: Een tabel met een overzicht van de opening van de verschillende stations in uren van de simulatie en tijdstippen in de week.

Dag	Klok uren	Uren simulatie	Geopende stations
Maandag	8:30-14:30	0-6	Alle stations
Maandag	14:30-16:00	6-7.5	Priklab, dagbehandeling en CT-scan
Maandag	16:00-17:30	7.5-9	Priklab
Dinsdag	7:30-8:00	9-9.5	Priklab
Dinsdag	8:00-8:30	9.5-10	Priklab, dagbehandeling en CT-scan
Dinsdag	8:30-14:30	10-16	Alle stations
Dinsdag	14:30-16:00	16-17.5	Priklab, dagbehandeling en CT-scan
Dinsdag	16:00-17:30	17.5-19	Priklab
Woensdag	7:30-8:00	19-19.5	Priklab
Woensdag	8:00-8:30	19.5-20	Priklab, dagbehandeling en CT-scan
Woensdag	8:30-14:30	20-26	Alle stations
Woensdag	14:30-16:00	26-27.5	Priklab, dagbehandeling en CT-scan
Woensdag	16:00-17:30	27.5-29	Priklab
Donderdag	7:30-8:00	29-29.5	Priklab
Donderdag	8:00-8:30	29.5-30	Priklab, dagbehandeling en CT-scan
Donderdag	8:30-14:30	30-36	Alle stations
Donderdag	14:30-16:00	36-37.5	Priklab, dagbehandeling en CT-scan
Donderdag	16:00-17:30	37.5-39	Priklab
Vrijdag	7:30-8:00	39-39.5	Priklab
Vrijdag	8:00-8:30	39.5-40	Priklab, dagbehandeling en CT-scan
Vrijdag	8:30-14:30	40-46	Alle stations
Vrijdag	14:30-16:00	46-47.5	Priklab, dagbehandeling en CT-scan
Zaterdag	8:00-16:00	47.5-54.5	CT-scan