

UNIVERSITY OF TWENTE.

BACHELOROPDRACHT TECHNISCHE WISKUNDE

# Simulatie van zeldzame gebeurtenissen

*Kans op grote benodigde voorraadcapaciteit van systeem met twee  
wachtrijen achter elkaar*

Simone van den Heuvel

S1558048

Begeleider: Anne Buijsrogge

30 juni 2017

## Samenvatting

In dit onderzoek is gekeken naar de kans dat een bepaalde grote voorraadcapaciteit van wachtrijen nodig is in een systeem met meerdere wachtrijen achter elkaar. Hierbij is niet alleen gekeken naar het aantal halffabricaten in de wachtrijen, maar ook naar de hoeveelheid plek die deze innemen als de halffabricaten in de verschillende wachtrijen niet even groot zijn. Voor het asymptotische gedrag van de hoeveelheid eenheden aan halffabricaten in het systeem is met behulp van een simulatie van één wachtrij en van twee wachtrijen een vermoeden ontstaan. Er is bewezen dat de ondergrens van dit asymptotische gedrag gelijk is aan het vermoeden. Verder is er een bovengrens bewezen die groter is dan het vermoeden. In het geval dat de halffabricaten in alle wachtrijen even groot zijn, geldt dat de bewezen bovengrens gelijk is aan de bewezen ondergrens en het asymptotisch gedrag dus gelijk is aan het vermoeden.

## 1 Inleiding

In een fabriekshal met meerdere servers achter elkaar kan het zijn dat de ene server veel langzamer werkt dan de ander en eigenlijk de aanvoer niet aan kan. Daarom heeft een fabriekshal een bepaalde voorraadcapaciteit nodig om de halffabricaten die staan te wachten om de volgende server in te gaan, zelf op te kunnen slaan. De vraag hierbij is hoeveel capaciteit hiervoor nodig is, zodanig dat er ook tijdens piekbelasting genoeg ruimte is om de halffabricaten op te slaan. En met welke kans deze capaciteit niet voldoende is, waardoor halffabricaten extern moeten worden opgeslagen.

Nu kan het ook nog zijn dat de halffabricaten in de verschillende wachtrijen niet even groot zijn, maar bijvoorbeeld in een van de servers wordt gehalveerd of verdubbeld. Het is dan niet voldoende om alleen te kijken hoeveel halffabricaten er in de gehele hal aanwezig zijn, want ook de ruimte die deze innemen moet worden meegenomen. De onderzoeksvraag van dit artikel is daarom:

Wat is de kans op een grote benodigde voorraadcapaciteit voor een fabriekshal met twee servers achter elkaar wanneer de halffabricaten in de wachtrijen niet even groot zijn?

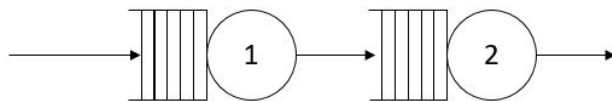
Vaak komen halffabricaten in een fabriekshal aan met stochastische tussenaankomsttijden en ook de servers zullen stochastische bedieningstijden

hebben. Wanneer in de literatuur wordt gekeken, wordt voor dit soort probleemstellingen vaak met simulatie een antwoord gezocht. Vaak wordt daarbij ook nog gebruik gemaakt van importance sampling zodat de simulatie sneller grote aantallen halffabricaten in het systeem heeft, zie [2], [3] en [4]. In deze artikelen wordt voor  $M|M|1$  systemen ook een afleiding gevonden voor het asymptotische gedrag van het aantal halffabricaten in het systeem. In [1] samen met [6] wordt er voor dit asymptotische gedrag ook een afleiding gevonden voor een systeem met een eindig aantal  $G|G|1$  servers achter elkaar. In al deze literatuur wordt alleen gekeken naar een groot aantal halffabricaten in de wachtrijen en wordt de grootte van deze halffabricaten niet meegenomen.

In dit artikel wordt eerst met behulp van de literatuur gekeken naar het asymptotische gedrag van het aantal halffabricaten in het gehele systeem. Samen met de resultaten van de gemaakte simulatie is er dan een vermoeden ontstaan voor het asymptotische gedrag van het aantal eenheden aan halffabricaten in het systeem. Er is vervolgens bewezen dat de ondergrens van dit asymptotische gedrag gelijk is aan het vermoeden. Ook is er voor dit gedrag een bovengrens bewezen. In het geval dat de halffabricaten in alle wachtrijen even groot zijn, zijn deze boven- en ondergrens gelijk aan elkaar.

## 2 Achtergrond

De onderzoeksvraag gaat over twee servers in serie, waarvan een schematische weergave is weergegeven in Figuur 1. Er zal echter ook gekeken worden naar een systeem met slechts één server en naar systemen met een eindig aantal servers achter elkaar. Het aantal halffabricaten in het gehele systeem is dus het aantal halffabricaten in de wachtrijen voor iedere server plus de halffabricaten die in de servers aanwezig zijn. Na de laatste server verlaat de halffabricaat gelijk de hal.



Figuur 1: Schematische weergave fabriekshal.

De probleemstelling gaat over de kans dat het aantal halffabricaten in het systeem een bepaalde grote waarde overstijgt. Hiervoor zou gekeken kunnen worden naar de stationaire kans dat het systeem zich in een bepaalde toestand bevindt. Dit wordt geschreven als  $P(L \geq M) = \lim_{t \rightarrow \infty} P(L(t) \geq M)$ , waarbij  $L(t)$  staat voor het aantal halffabricaten in het systeem op tijdstip  $t$ . Naast deze stationaire kansen kan er ook gekeken worden naar de kans dat het systeem eerder een bepaald aantal halffabricaten bereikt, voordat het systeem weer leeg is. Dit wordt weergegeven als  $P(t_M < t_0)$  met  $t_M$  het tijdstip dat er voor het eerst  $M$  halffabricaten in het systeem aanwezig zijn en  $t_0$  het tijdstip waarop het systeem weer voor het eerst leeg is.

Er zal echter niet alleen worden gekeken naar de kans op een gegeven aantal halffabricaten in het systeem, maar ook naar de hoeveelheid ruimte die deze innemen. Hierbij wordt  $g_i$  gebruikt voor de grootte van de halffabricaten in de wachtrij bij server  $i$ . De stationaire kans verandert nu naar  $P(W \geq N) = \lim_{t \rightarrow \infty} P(W(t) \geq N)$ , waarbij  $W(t)$  staat voor het aantal eenheden aan halffabricaten in het systeem op tijdstip  $t$ . Bij de hoeveelheid eenheden die in een actieve periode, de tijd tussen twee opeenvolgende lege systemen, bereikt wordt, staat  $\hat{t}_N$  voor het tijdstip dat er voor het eerst minstens  $N$  eenheden aan halffabricaten in het systeem aanwezig zijn. De kans wordt nu genoteerd als  $P(\hat{t}_N < t_0)$ .

## 2.1 Stationaire kansen

Wanneer de aankomst- en bedieningsverdelingen exponentieel zijn met respectievelijk variabelen  $\lambda$  en  $\mu_i$ , kan het volgende worden gezegd over de stationaire kansen.

Neem  $P_i = \lim_{t \rightarrow \infty} P(L(t) = i)$ . Voor een enkele server is dan bekend dat  $P_i = \rho_1^i (1 - \rho_1) = \rho_1^i - \rho_1^{i+1}$ , met  $\rho_j = \frac{\lambda}{\mu_j}$  [5]. Hieruit volgt dat:

$$\begin{aligned}
 P(L \geq M) &= 1 - P(L \leq M - 1) \\
 &= 1 - \sum_{k=0}^{M-1} P_k \\
 &= 1 - \sum_{k=0}^{M-1} (\rho_1^k - \rho_1^{k+1}) \\
 &= 1 - (\rho_1^0 - \rho_1^M) \\
 &= \rho_1^M.
 \end{aligned}$$

Bij twee servers achter elkaar geldt voor de eerste server nog steeds dat  $P(L_1 = m_1) = \rho_1^{m_1}(1 - \rho_1)$  met  $L_i$  het aantal halffabricaten in de wachtrij van server  $i$ , inclusief die in service, op tijdstip  $t \rightarrow \infty$ . Voor de tweede server geldt bij deze verdelingen van de eerste server, dat de aankomstverdeling ook exponentieel met parameter  $\lambda$  is. Voor de tweede server geldt dus ook dat  $P(L_2 = m_2) = \rho_2^{m_2}(1 - \rho_2)$ . Voor de gezamenlijke kans geldt dat  $P(L_1 = m_1, L_2 = m_2) = \rho_1^{m_1}(1 - \rho_1)\rho_2^{m_2}(1 - \rho_2)$ .

Omdat  $L = L_1 + L_2$ , is stationaire kans af te leiden als:

$$\begin{aligned}
P(L \geq M) &= 1 - P(L \leq M - 1) \\
&= 1 - \sum_{k=0}^{M-1} \sum_{i=0}^k P(L_1(t) = i, L_2(t) = k - i) \\
&= 1 - \sum_{k=0}^{M-1} (1 - \rho_1)(1 - \rho_2) \sum_{i=0}^k \rho_1^i \rho_2^{k-i} \\
&= 1 - \sum_{k=0}^{M-1} (1 - \rho_1)(1 - \rho_2) \frac{\rho_1^{k+1} - \rho_2^{k+1}}{\rho_1 - \rho_2} \\
&= 1 - \frac{(1 - \rho_1)(1 - \rho_2)}{\rho_1 - \rho_2} \left( \sum_{j=1}^M \rho_1^j - \sum_{l=1}^M \rho_2^l \right) \\
&= 1 - \frac{(1 - \rho_1)(1 - \rho_2)}{\rho_1 - \rho_2} \left( \frac{1 - \rho_1^{M+1}}{1 - \rho_1} - 1 - \left( \frac{1 - \rho_2^{M+1}}{1 - \rho_2} - 1 \right) \right) \\
&= 1 + \rho_1^{M+1} \left( \frac{1 - \rho_2}{\rho_1 - \rho_2} \right) - \rho_2^{M+1} \left( \frac{1 - \rho_1}{\rho_1 - \rho_2} \right) - 1.
\end{aligned}$$

Als nu  $c_i = \frac{1 - \rho_i}{\rho_1 - \rho_2}$ , kan de kans worden geschreven als  $c_2 \rho_1^{M+1} - c_1 \rho_2^{M+1}$ . Verder impliceert  $\rho_1 < \rho_2$  dat  $c_1, c_2 < 0$  en vice versa. Als nu  $M$  naar oneindig wordt genomen, domineert de grootste term dus gaat de kans richting:

$$\lim_{M \rightarrow \infty} P(L \geq M) \approx \begin{cases} c_1 \rho_2^{M+1} & \text{als } \rho_1 < \rho_2 \\ c_2 \rho_1^{M+1} & \text{als } \rho_1 > \rho_2. \end{cases}$$

## 2.2 Aantal halffabricaten in actieve periode

Wanneer er gekeken wordt naar het aantal halffabricaten dat in een actieve periode bereikt wordt, is er voor een stabiel G|G|1 systeem met een eindig aantal servers achter elkaar bekend dat  $\lim_{M \rightarrow \infty} \frac{1}{M} \log P(K_M < K_0) = \Lambda_A(-\theta_{min})$  [1][6]. Hierbij is  $K_M$  de index van de eerste halffabricaat die ervoor zorgt dat er  $M$  halffabricaten in het systeem zijn en  $K_0$  de index van de eerste halffabricaat die in een leeg systeem arriveert. Daarnaast is  $\Lambda_X(\theta) = \log \mathbb{E}[e^{\theta X}]$  en  $\theta_{min} = \min_j(\theta_j)$  waarbij geldt dat  $\theta_j = \sup_{\theta} \{ \Lambda_A(-\theta) + \Lambda_{B_j}(\theta) \leq 0 \}$ . Hierbij is  $A$  de verdeling van de tussenaankomsttijden en  $B_j$  de verdeling van de bedieningstijden bij server  $j$ . Nu geldt er het volgende:

**Theorie 1.** *Voor een eindig aantal servers achter elkaar geldt dat  $\lim_{M \rightarrow \infty} \frac{1}{M} \log P(t_M < t_0) = \Lambda_A(-\theta_{min})$ .*

*Bewijs.* Uit [1] samen met [6] blijkt dat  $\lim_{M \rightarrow \infty} \frac{1}{M} \log P(K_M < K_0) = \Lambda_A(-\theta_{min})$ . Ook geldt dat  $t_M = \sum_{k=1}^{K_M-1} A_k$  met  $A_k$  de tijd tussen de aankomst van halffabricaat  $k$  en  $k+1$  en  $t_0 \in \left[ \sum_{k=1}^{K_0-2} A_k, \sum_{k=1}^{K_0-1} A_k \right]$ , omdat dit tijdstip zit tussen de aankomst van de halffabricaat die een leeg systeem voor zich ziet en de halffabricaat daarvoor.

We nemen aan dat  $M > 0$  dus nu zijn er twee gevallen mogelijk:

- $K_M < K_0$ . Omdat  $K_i$  gehele getallen zijn, is dit gelijk aan  $K_M \leq K_0 - 1$ . Nu geldt dat  $t_0 \geq \sum_{k=1}^{K_0-2} A_k = \sum_{k=1}^{(K_0-1)-1} A_k \geq \sum_{k=1}^{K_M-1} A_k = t_M$ .  $t_0$  en  $t_M$  kunnen echter nooit gelijk zijn aan elkaar, dus is  $t_0 > t_M$ .
- $K_M > K_0$ . Nu geldt dat  $t_0 \leq \sum_{k=1}^{K_0-1} A_k < \sum_{k=1}^{K_M-1} A_k = t_M$ .

Hieruit volgt dus dat  $K_M < K_0 \Rightarrow t_M < t_0$  en  $K_M > K_0 \Rightarrow t_M > t_0$ . Wat weer impliceert dat de kans op de gebeurtenissen gelijk is, dus  $P(K_M < K_0) = P(t_M < t_0)$ . Dit geeft dat  $\lim_{M \rightarrow \infty} \frac{1}{M} \log P(t_M < t_0) = \lim_{M \rightarrow \infty} \frac{1}{M} \log P(K_M < K_0) = \Lambda_A(-\theta_{min})$ .  $\square$

Een gevolg van deze theorie is:

**Gevolg 1.1.** *Wanneer de aankomsten exponentieel verdeeld zijn met parameter  $\lambda$  en server  $i$  bedieningstijden heeft die exponentieel verdeeld zijn met parameter  $\mu_i$ , geldt dat  $\lim_{M \rightarrow \infty} \frac{1}{M} \log P(t_M < t_0) = \log \left( \frac{\lambda}{\mu_{min}} \right)$  met  $\mu_{min} = \min_i \mu_i$ .*

*Bewijs.* Uit Theorie 1 volgt dat  $\lim_{M \rightarrow \infty} \frac{1}{M} \log P(t_M < t_0) = \Lambda_A(-\theta_{min})$ .  
Voor de exponentiële verdelingen geldt dat:

$$\begin{aligned}
\theta_i &= \sup_{\theta} \{ \Lambda_A(-\theta) + \Lambda_{B_i}(\theta) \leq 0 \} \\
&= \sup_{\theta} \left\{ \log \frac{\lambda}{\lambda + \theta} + \log \frac{\mu_i}{\mu_i - \theta} \leq 0 \right\} \\
&= \sup_{\theta} \left\{ \log \frac{\lambda \mu_i}{(\lambda + \theta)(\mu_i - \theta)} \leq 0 \right\} \\
&= \sup_{\theta} \left\{ \frac{\lambda \mu_i}{\lambda \mu_i + (\mu_i - \lambda)\theta - \theta^2} \leq 1 \right\} \\
&= \sup_{\theta} \{ \lambda \mu_i \leq \lambda \mu_i + (\mu_i - \lambda)\theta - \theta^2 \} \\
&= \sup_{\theta} \{ \theta^2 \leq (\mu_i - \lambda)\theta \} \\
&= \mu_i - \lambda.
\end{aligned}$$

Dus  $\theta_{min} = \mu_{min} - \lambda$  met  $\mu_{min} = \min_i \mu_i$ .

Voor de kans geldt nu:

$$\begin{aligned}
\lim_{M \rightarrow \infty} \frac{1}{M} \log P(t_M < t_0) &= \Lambda_A(-\theta_{min}) \\
&= \log \left( \frac{\lambda}{\lambda + (\mu_{min} - \lambda)} \right) \\
&= \log \left( \frac{\lambda}{\mu_{min}} \right).
\end{aligned}$$

□

### 2.2.1 Verschillende groottes van halffabricaten

Theorie 1 en Gevolg 1.1 gaan over het aantal halffabricaten in de fabriekshal, maar de onderzoeksvraag gaat ook over de hoeveelheid ruimte die deze innemen. Wanneer er wordt aangenomen dat de halffabricaten in alle wachtrijen nog steeds even groot zijn, maar nu  $g$  eenheden, blijven de kansen op een bepaald aantal halffabricaten gelijk. Er geldt dus dat  $P(\hat{t}_N < t_0) = P(t_M < t_0)$ , met  $M = \frac{N}{g}$ . De kans op  $K$  eenheden is dus  $g$  keer zo groot als de kans op  $K$  halffabricaten. Daardoor ontstaat het vermoeden dat  $\lim_{N \rightarrow \infty} \frac{1}{N} \log P(\hat{t}_N < t_0) = \frac{1}{g} \Lambda_A(-\theta_{min}) = \max_i \frac{1}{g} \Lambda_A(-\theta_i)$ .

Het asymptotische gedrag hangt dus af van  $\theta_{min}$  wat gelijk is aan de  $\theta$  behorende bij de langzaamste server. Wanneer de halffabricaten in de wachtrijen niet dezelfde grootte hebben, is het vermoeden dat dit gedrag niet altijd meer afhangt van de langzaamste server. Wanneer de halffabricaten in de snellere server namelijk veel groter zijn dan bij de langzame, zal de ruimte die nodig is meer afhangen van de snelle server. Daarom is het vermoeden ontstaan dat het asymptotische gedrag voor het aantal eenheden aan halffabricaten in het systeem in de buurt ligt van  $\lim_{N \rightarrow \infty} \frac{1}{N} \log P(\hat{t}_N < t_0) = \max_i \frac{1}{g_i} \Lambda_A(-\theta_i)$ .

### 3 Simulatie

Om te kijken of het vermoeden kan kloppen, is er van de probleemstelling een simulatie gemaakt. Dit is gedaan voor een fabriekshal met één server en met twee servers.

#### 3.1 Enkele server

Voor de simulatie met één server zijn de volgende toestandsvariabelen gebruikt:

- $T$  Het tijdstip waarop de simulatie zich bevindt.
- $T_{vorig}$  Het tijdstip van de vorige gebeurtenis.
- $A$  Het tijdstip waarop de volgende halffabricaat arriveert.
- $B_1$  Het tijdstip waarop de eerste server klaar is met zijn huidige halffabricaat.
- $W_1$  Het aantal halffabricaten in de wachtrij voor de eerste server, inclusief de halffabricaat in de server, op tijdstip  $T$ .
- $W_{max}$  Het grootste aantal halffabricaten in de server bereikt in de huidige actieve periode.

De stationaire kansen zullen worden bijgehouden in  $L_{tijd}$ , deze houdt bij hoeveel tijd de simulatie een bepaald aantal halffabricaten in het systeem heeft. De maximale hoeveelheid halffabricaten die per actieve periode in de fabriekshal aanwezig was, wordt bijgehouden in  $L_{max}$ .

Om de simulatie iets sneller te laten lopen, is er aangenomen dat de simulatie begint op  $T = 0$  wanneer de eerste halffabricaat de eerste server in gaat.



De simulatie ziet er als volgt uit:

```
genereer eerste  $A$  en  $B_1$ 
 $W_1 \leftarrow 1$ 
 $T \leftarrow 0$ 
while  $T <$  gewilde tijdsperiode do
   $T_{vorig} \leftarrow T$ 
   $T \leftarrow \min(A, B_1)$ 
   $L_{tijd}[W_1] \leftarrow L_{tijd}[W_1] + (T - T_{vorig})$ 
  if  $W_1 > W_{max}$  then
     $W_{max} \leftarrow W_1$ 
  else if  $W_1 = 0$  then
     $L_{max}[W_{max}] \leftarrow L_{max}[W_{max}] + 1$ 
     $W_{max} \leftarrow 0$ 
  end if
  if  $T = A$  then
     $W_1 \leftarrow W_1 + 1$ 
    genereer nieuwe  $A$ 
    if  $W_1 = 1$  then
      genereer nieuwe  $B_1$ 
    end if
  else if  $T = B_1$  then
     $W_1 \leftarrow W_1 - 1$ 
    if  $W_1 > 0$  then
      genereer nieuwe  $B_1$ 
    end if
  end if
end while.
```

Het genereren van de nieuwe aankomst- en bedieningstijden hangt van de verdeling af. Door de soort verdeling en bijbehorende variabelen in te voeren, worden er random getallen gegenereerd volgens die verdeling. Deze getallen worden dan telkens opgeteld bij de huidige simulatietijd  $T$ .

### 3.2 Twee servers

Voor de simulatie met twee servers is de simulatie van de enkele server uitgebreid met de volgende variabelen:

$B_2$  Het tijdstip waarop de tweede server klaar is met zijn huidige half-fabricaat.

- $W_2$  Het aantal halffabricaten in de wachtrij voor de tweede server, inclusief die in de server.
- $g_i$  Grootte van halffabricaat in wachtrij  $i$ , alleen als geheel getal.
- $W$  Het aantal eenheden aan halffabricaten in de eerste en tweede wachtrij bij elkaar opgeteld.

In deze simulatie gaan  $W_{max}$ ,  $L_{max}$  en  $L_{tijd}$  over het aantal halffabricaten in beide wachtrijen, dus over  $W$ . Deze wordt na het gelijkstellen van  $T_{vorig}$  aan  $T$  berekend door  $W = W_1g_1 + W_2g_2$ . In de simulatie verandert  $W_1$  in deze  $W$ , wordt er naast  $A$  en  $B_1$  ook  $B_2$  gegenereerd en wordt  $T = \min(A, B_1, B_2)$ .

Wanneer  $T = B_2$  gebeurt nu wat er bij een enkele server gebeurde op  $T = B_1$ , hierbij veranderen  $B_1$  en  $W_1$  in respectievelijk  $B_2$  en  $W_2$ . Als nu  $T = B_1$  gebeurt er:

```

 $W_1 \leftarrow W_1 - 1$ 
 $W_2 \leftarrow W_2 + 1$ 
if  $W_1 > 0$  then
    genereer  $B_1$ 
end if
if  $W_2 = 1$  then
    genereer  $B_2$ 
end if.

```

### 3.3 Betrouwbaarheidsinterval

Wanneer de simulatie is afgelopen, zijn  $L_{max}$  en  $L_{tijd}$  lijsten geworden met verschillende waarden voor het aantal eenheden aan halffabricaten in het systeem. Van deze waarden kan nu per aantal een gemiddelde kans met bijbehorend betrouwbaarheidsinterval worden berekend. Hiervoor zal er eerst een simulatie van  $Q$  actieve periodes worden uitgevoerd om te bepalen welke tijd dit inneemt, waarna de simulatie  $R$  keer over deze tijdspanne zal worden uitgevoerd.  $L_{max}$  wordt steeds weer meegegeven en blijft één lijst,  $L_{tijd}$  wordt een matrix met in elke rij de waarden van een simulatie.

Om het gemiddelde kansen van  $L_{tijd}$  te berekenen, worden de  $R$  rijen eerst allemaal omgezet van tijden naar kansen. De kans dat er in simulatie  $i$  minstens  $j$  eenheden aan halffabricaten in het systeem waren, is gelijk aan  $X_{i,j} = \frac{\sum_{k=j}^{\infty} L_{tijd}[i][k]}{\text{tijdspanne}}$ . Van al deze kansen wordt nu per aantal eenheden aan

halffabricaten een gemiddelde kans berekend:  $\bar{X}_j = \frac{\sum_{i=1}^R X_{i,j}}{R}$ . Hierna wordt voor ieder aantal de steekproefvariantie berekend door  $S_j^2 = \frac{\sum_{i=1}^R (X_{i,j} - \bar{X}_j)^2}{R-1}$ .

$L_{max}$  is één lijst over  $RQ$  actieve periodes. De gemiddelde kans dat er nu tijdens een actieve periode meer dan  $j$  eenheden aan halffabricaten in het systeem hebben gezeten, is gelijk aan  $\bar{X}_j = \frac{\sum_{k=j}^{\infty} L_{max}[k]}{RQ}$ . In de lijst van steekproefvarianties is de waarde bij minstens  $j$  eenheden aan halffabricaten gelijk aan  $S_j^2 = \frac{\sum_{i=1}^{RQ} (X_{i,j} - \bar{X}_j)^2}{RQ-1}$ , met  $X_{i,j} = 1$  als in actieve periode  $i$  meer dan  $j$  eenheden aanwezig zijn geweest. Anders is  $X_{i,j}$  gelijk aan 0.

Nadat de gemiddeldes en steekproefvarianties bekend zijn, kunnen de foutmarges berekend worden. Deze is voor  $L_{tijd}$  gelijk aan  $z_{1-\alpha/2} \sqrt{\frac{S^2}{R}}$  en voor  $L_{max}$  aan  $z_{1-\alpha/2} \sqrt{\frac{S^2}{RQ}}$ . Voor een 95% betrouwbaarheidsinterval is  $\alpha$  gelijk aan 0,05 en hoort bij  $z_{0,975} = 1,96$ . De hieruit komende lijsten samen met de lijsten van gemiddeldes geeft voor ieder aantal eenheden aan halffabricaten in het systeem aan in welk interval de kans ligt dat dit aantal wordt bereikt.

## 4 Resultaten

Uit de simulatie komen dus de gemiddeldes en bijbehorende betrouwbaarheidsintervallen van twee lijsten,  $L_{max}$  en  $L_{tijd}$ .

Uit Theorie 1 volgt dat voor het aantal halffabricaten in het systeem geldt dat:

$$\begin{aligned} \lim_{M \rightarrow \infty} \frac{1}{M} \log P(t_M < t_0) &= \Lambda_A(-\theta_{min}), \\ \lim_{M \rightarrow \infty} \log P(t_M < t_0)^{\frac{1}{M}} &= \log M_A(-\theta_{min}), \\ \lim_{M \rightarrow \infty} P(t_M < t_0) &\approx e^{M \log M_A(-\theta_{min})}, \\ \lim_{M \rightarrow \infty} P(t_M < t_0) &\approx M_A(-\theta_{min})^M. \end{aligned}$$

Met deze theoretische waarden zullen de resultaten uit de simulatie daarom worden vergeleken. Omdat deze kans alleen geldt in de limiet, worden er aankomst- en bedieningsverdelingen gekozen die in de buurt liggen van instabiliteit. In die gevallen worden er namelijk grotere waarden van  $M$  bereikt.

Alle onderstaande resultaten zijn berekend met  $R = 10000$  simulaties van een duur van  $Q = 1000$  actieve periodes en een 95 procent betrouwbaarheidsinterval.

#### 4.1 Enkele server

Voor een fabriekshal met slechts één server komen er uit de simulatie de volgende resultaten, zie Tabellen 1 tot en met 3. Ook de theoretische waarden  $M_A(-\theta_1)^M$  zijn gegeven.

Tabel 1:  $A \sim \exp(\frac{1}{5})$ ,  $B_1 \sim \exp(\frac{1}{4.5})$

$M$	$L_{max}$	$L_{tijd}$	Theoretisch
20	0.015±0.0001	0.120±0.001	0.122
40	0.0016±0.00002	0.0143±0.0004	0.0148
60	0.00018±0.00001	0.00172±0.00014	0.00180
80	0.000020±0.0003	0.000236±0.000056	0.000218
100	0.0000032±0.0000010	0.0000365±0.0000195	0.0000266

Tabel 2:  $A \sim \exp(\frac{1}{5})$ ,  $B_1 \sim U(1, 8.8)$

$M$	$L_{max}$	$L_{tijd}$	Theoretisch
50	0.008±0.00005	0.185±0.002	0.189
100	0.0011±0.00002	0.0336±0.0009	0.0357
150	0.00019±0.00001	0.00611±0.00042	0.00676
200	0.00003±0.000003	0.00119±0.00020	0.00128
250	0.000006±0.000001	0.000250±0.000096	0.000241
300	0.0000010±0.0000005	0.0000522±0.0000432	0.0000456

Tabel 3:  $A \sim U(1, 9)$ ,  $B_1 \sim \exp(\frac{1}{4.5})$

$M$	$L_{max}$	$L_{tijd}$	Theoretisch
15	0.0094±0.0001	0.0760±0.0008	0.0726
30	0.00061±0.00001	0.00539±0.00025	0.00526
45	0.000042±0.000004	0.000347±0.000059	0.000382
60	0.0000023±0.0000009	0.0000190±0.0000111	0.0000277

Te zien is dat de waarden van  $L_{tijd}$  bijna allemaal in het betrouwbaarheidsinterval vallen, maar  $L_{max}$  veel kleiner is. Een reden hiervoor zou kunnen

zijn dat bij een enkele server de kans dat het systeem na 1 halffabricaat alweer leeg kan zijn, relatief groot is. In de resultaten is namelijk te zien dat  $L_{max}[1] = 1$  en  $L_{max}[2]$  afzakt naar 0.6 of zelfs 0.4, terwijl dit 0.8 of 0.9 hoort te zijn. Er zijn dus relatief veel actieve periodes die maximaal 1 halffabricaat in het systeem hadden, waardoor de kansen op grotere waarden kleiner worden.

## 4.2 Aantal halffabricaten bij dubbele server

Voor twee servers achter elkaar waarbij gekeken wordt hoeveel halffabricaten in de wachtrijen staan, zijn de resultaten te zien in Tabellen 4 tot en met 6. Hier zijn de theoretische waarden gelijk aan  $\max_i M_A(-\theta_i)^M$ .

Tabel 4:  $A \sim \exp(\frac{1}{5})$ ,  $B_1 \sim \exp(\frac{1}{4.5})$ ,  $B_2 \sim \exp(\frac{1}{4})$

$M$	$L_{max}$	$L_{tijd}$	Theoretisch
25	0.0633±0.0002	0.1255±0.0005	0.0718
50	0.00500±0.00004	0.00917±0.00016	0.00515
75	0.000359±0.000012	0.000662±0.000043	0.000370
100	0.0000253±0.0000032	0.0000523±0.0000128	0.0000266
125	0.00000176±0.00000084	0.00000413±0.00000383	0.00000191

Tabel 5:  $A \sim \exp(\frac{1}{5})$ ,  $B_1 \sim U(1, 8)$ ,  $B_2 \sim \exp(\frac{1}{4})$

$M$	$L_{max}$	$L_{tijd}$	Theoretisch
20	0.0527±0.0001	0.0784±0.0004	0.0308
40	0.002318±0.000030	0.002875±0.000076	0.000948
60	0.0000788±0.0000055	0.0000965±0.0000161	0.0000292
80	0.00000274±0.000000103	0.00000413±0.00000377	0.00000090

Tabel 6:  $A \sim U(1, 9)$ ,  $B_1 \sim U(1, 8.8)$ ,  $B_2 \sim \exp(\frac{1}{4})$

$M$	$L_{max}$	$L_{tijd}$	Theoretisch
25	0.0609±0.0001	0.1344±0.0006	0.0927
50	0.00511±0.00004	0.01234±0.00023	0.00858
75	0.000468±0.000013	0.001155±0.000071	0.000795
100	0.0000432±0.0000040	0.0001013±0.0000205	0.0000737
125	0.00000337±0.00000112	0.00000675±0.00000540	0.00000683

Hier is te zien dat waarden niet helemaal overeenkomen met de betrouwbaarheidsintervallen, maar wel in dezelfde orde van grootte vallen. De verschillen zouden kunnen zitten in het feit dat de theoretische waarden eigenlijk een benadering zijn die alleen in de limiet gelden.

### 4.3 Aantal eenheden aan halffabricaten bij dubbele server

Wanneer er wordt gekeken naar de hoeveelheid eenheden die de halffabricaten innemen, is het vermoeden dat de kans de vorm heeft van  $\max_i M_A(-\theta_i)^{\frac{N}{g_i}}$ . In Tabellen 7 tot en met 9 worden de resultaten daarom vergeleken met deze waarden.

Tabel 7:  $A \sim \exp(\frac{1}{5})$ ,  $B_1 \sim \exp(\frac{1}{4.5})$ ,  $B_2 \sim \exp(\frac{1}{4})$ , ,  $g_1 = 2, g_2 = 2$

$N$	$L_{max}$	$L_{tijd}$	Vermoeden
50	0.0633±0.0002	0.1257±0.0005	0.0718
100	0.00504±0.00004	0.00923±0.00015	0.00515
150	0.000359±0.000012	0.000642±0.000040	0.000370
200	0.0000238±0.0000030	0.0000388±0.0000095	0.0000266
225	0.00000622±0.00000459	0.00000972±0.00000855	0.00000712
250	0.00000111±0.00000065	0.00000348±0.00000337	0.00000191

Tabel 8:  $A \sim \exp(\frac{1}{5})$ ,  $B_1 \sim \exp(\frac{1}{4})$ ,  $B_2 \sim \exp(\frac{1}{2})$ , ,  $g_1 = 1, g_2 = 25$

$N$	$L_{max}$	$L_{tijd}$	Vermoeden
50	0.462±0.0003	0.161±0.0001	0.160
100	0.1257±0.0002	0.0257±0.0001	0.00256
150	0.02721±0.00010	0.00411±0.00002	0.00410
200	0.004779±0.000043	0.000656±0.000010	0.000655
250	0.000779±0.000017	0.000105±0.000004	0.000105
300	0.0001267±0.0000070	0.0000162±0.0000015	0.0000168
350	0.00001856±0.00000269	0.00000249±0.00000058	0.00000268

Net zoals bij de dubbele server waarbij gekeken werd naar het aantal halffabricaten, vallen de waarden van het vermoeden weer niet altijd in het betrouwbaarheidsinterval, maar zijn de kansen vaak wel in dezelfde orde van grootte. De waarden van  $L_{tijd}$  komen nog het beste overeen en de waarden voor  $L_{max}$  zijn bijna altijd groter dan het vermoeden. Het verschil zou weer kunnen zitten in het feit dat het vermoeden weer een benadering. Ook kan er

Tabel 9:  $A \sim U(1, 9)$ ,  $B_1 \sim U(1, 8.8)$ ,  $B_2 \sim \exp(\frac{1}{4})$ ,  $g_1 = 1, g_2 = 25$

$N$	$L_{max}$	$L_{tijd}$	Vermoeden
200	$0.2394 \pm 0.0003$	$0.0607 \pm 0.0001$	0.0505
400	$0.03330 \pm 0.00011$	$0.00299 \pm 0.00002$	0.00255
600	$0.001998 \pm 0.000026$	$0.000146 \pm 0.000004$	0.000129
800	$0.00010102 \pm 0.00000596$	$0.00000769 \pm 0.00000089$	0.00000649
900	$0.00002400 \pm 0.00000291$	$0.00000189 \pm 0.00000048$	0.00000146
1000	$0.000005129 \pm 0.000001343$	$0.000000359 \pm 0.000000247$	0.000000327

worden meegenomen dat het vermoeden gaat over het asymptotische gedrag en dus alleen geldt voor grote waarden van  $N$ . In Tabel 9 lijkt 1000 heel groot, maar gedeeld door  $g_2$  zijn er eigenlijk pas rond de 40 halffabricaten in het systeem aanwezig.

## 5 Bewijs

Met behulp van de simulaties is dus het volgende vermoeden ontstaan:  $\lim_{N \rightarrow \infty} \frac{1}{N} \log P(\hat{t}_N < t_0) = \max_i \frac{1}{g_i} \Lambda_A(-\theta_i)$ . Omdat de resultaten uit de simulaties bijna altijd groter waren dan dit vermoeden, zal eerst de ondergrens worden bewezen.

**Theorie 2.** *Voor een systeem met een eindig aantal servers achter elkaar, geldt dat  $\lim_{N \rightarrow \infty} \frac{1}{N} \log P(\hat{t}_N < t_0) \geq \max_i \frac{1}{g_i} \Lambda_A(-\theta_i)$ .*

*Bewijs.* Neem  $t_{i,j}$  voor het tijdstip dat er bij server  $i$  voor het eerst  $j$  halffabricaten aanwezig zijn en  $t_j$  voor het tijdstip dat er in het gehele systeem voor het eerst  $j$  halffabricaten aanwezig zijn. Volgens Theorie 1 geldt voor het aantal halffabricaten in een systeem met maar één server:

$$\lim_{M \rightarrow \infty} \frac{1}{M} \log P(t_{1,M} < t_{1,0}) = \Lambda_A(-\theta_1).$$

Wanneer de halffabricaten van deze server nu een grootte van  $g_1$  eenheden krijgen, is tijdstip  $t_{1,M}$  gelijk aan  $\hat{t}_{1,N}$  waarop voor het eerst  $N = Mg_1$

eenheden bij de eerste server aanwezig zijn. Nu geldt dat:

$$\begin{aligned} \lim_{N \rightarrow \infty} \frac{1}{N} \log P(\hat{t}_{1,N} < t_{1,0}) &= \lim_{M \rightarrow \infty} \frac{1}{M g_1} \log P(t_{1,M} < t_{1,0}) \\ &= \frac{1}{g_1} \lim_{M \rightarrow \infty} \frac{1}{M} \log P(t_{1,M} < t_{1,0}) \\ &= \frac{1}{g_1} \Lambda_A(-\theta_1). \end{aligned}$$

Doordat er geen verdere aannames over deze eerste server zijn gedaan, geldt voor alle servers in het systeem dat  $\lim_{N \rightarrow \infty} \frac{1}{N} \log P(\hat{t}_{i,N} < t_{i,0}) = \frac{1}{g_i} \Lambda_A(-\theta_i)$ , wanneer zij alleen in de serie zouden staan.

Om nu over te gaan naar  $P(\hat{t}_N < t_0)$  kan er worden beredeneerd dat deze kans altijd groter is dan  $P(\hat{t}_{i,N} < t_{i,0})$ . Er geldt namelijk altijd dat  $\hat{t}_N \leq \hat{t}_{i,N}$ , omdat het onmogelijk is dat er eerder  $N$  eenheden aan halffabricaten bij server  $i$  staan dan in het gehele systeem. Daarnaast geldt altijd dat  $t_0 \geq t_{i,0}$ , want het hele systeem kan pas leeg zijn als wachtrij  $i$  dit ook is. Hieruit volgt dat  $\hat{t}_{i,N} < t_{i,0} \Rightarrow \hat{t}_N \leq \hat{t}_{i,N} < t_{i,0} \leq t_0$ , dus  $\hat{t}_N < t_0$ . Dit geeft weer dat  $\lim_{N \rightarrow \infty} \frac{1}{N} \log P(\hat{t}_N < t_0) \geq \lim_{N \rightarrow \infty} \frac{1}{N} \log P(\hat{t}_{i,N} < t_{i,0})$  voor alle servers in de serie, dus in het bijzonder ook voor het maximum. Hieruit volgt dat:

$$\begin{aligned} \lim_{N \rightarrow \infty} \frac{1}{N} \log P(\hat{t}_N < t_0) &\geq \max_i \lim_{N \rightarrow \infty} \frac{1}{N} \log P(\hat{t}_{i,N} < t_{i,0}) \\ &= \max_i \frac{1}{g_i} \Lambda_A(-\theta_i). \end{aligned}$$

□

Deze theorie zegt alleen iets over het bereiken van een bepaald groot aantal eenheden in een actieve periode, maar ook over de stationaire kansen kan nu het volgende worden bewezen.

**Gevolg 2.1.**  $\lim_{N \rightarrow \infty} \frac{1}{N} \log P(W \geq N) \geq \max_i \frac{1}{g_i} \Lambda_A(-\theta_i)$ .

*Bewijs.* Uit [1] samen met Theorie 1 is bekend dat onder bepaalde voorwaarden geldt dat  $P(L \geq M) \geq P(t_M < t_0)$ . Door middel van soortgelijke argumenten kan worden bewezen dat ook geldt dat:

$$\begin{aligned} \lim_{N \rightarrow \infty} \frac{1}{N} \log P(W \geq N) &\geq \lim_{N \rightarrow \infty} \frac{1}{N} \log P(\hat{t}_N < t_0) \\ &\geq \max_i \frac{1}{g_i} \Lambda_A(-\theta_i). \end{aligned}$$

□



Nu de ondergrens gelijk is aan het vermoeden, is alleen nog nodig te bewijzen dat dit ook gelijk is aan de bovengrens. Voor de bovengrens kan het volgende worden bewezen:

**Theorie 3.** *Voor een systeem met een eindig aantal servers achter elkaar, geldt dat  $\lim_{N \rightarrow \infty} \frac{1}{N} \log P(\hat{t}_N < t_0) \leq \frac{1}{\max_i g_i} \max_j \Lambda_A(-\theta_j)$ .*

*Bewijs.* Wanneer  $\hat{t}_N < t_0$  zijn er in die actieve periode dus meer dan  $N$  eenheden aan halffabricaten aanwezig geweest. Dit kan alleen als er minstens  $M_{min} = \left\lceil \frac{N}{g_{max}} \right\rceil$  halffabricaten in het systeem hebben gezeten, hierbij is  $g_{max} = \max_i g_i$ . Er geldt dus dat  $t_{M_{min}} \leq \hat{t}_N$  wat weer impliceert dat  $P(\hat{t}_N < t_0) \leq P(t_{M_{min}} < t_0)$ . Uit [1] samen met Theorie 1 is bekend dat onder bepaalde voorwaarden geldt dat  $P(L \geq M) \geq P(t_M < t_0)$ . Hieruit volgt dat:

$$\begin{aligned} \lim_{N \rightarrow \infty} \frac{1}{N} \log P(\hat{t}_N < t_0) &\leq \lim_{M_{min} \rightarrow \infty} \frac{1}{M_{min} g_{max}} \log P(t_{M_{min}} < t_0) \\ &\leq \frac{1}{g_{max}} \lim_{M_{min} \rightarrow \infty} \frac{1}{M_{min}} \log P(L \geq M_{min}). \end{aligned}$$

Uit [1] blijkt dit laatste limiet kleiner te zijn dan  $\Lambda_A(-\theta_{min})$ , waaruit volgt dat:

$$\begin{aligned} \lim_{N \rightarrow \infty} \frac{1}{N} \log P(\hat{t}_N < t_0) &\leq \frac{1}{g_{max}} \Lambda_A(-\theta_{min}) \\ &= \frac{1}{\max_i g_i} \max_j \Lambda_A(-\theta_j). \end{aligned}$$

□

Een gevolg van deze theorie is:

**Gevolg 3.1.** *Wanneer  $\forall i, j$  geldt dat  $g_i = g_j$  volgt  $\lim_{N \rightarrow \infty} \frac{1}{N} \log P(\hat{t}_N < t_0) = \max_i \frac{1}{g_i} \Lambda_A(-\theta_i)$ .*

*Bewijs.* Theorie 2 geeft al dat  $\lim_{N \rightarrow \infty} \frac{1}{N} \log P(\hat{t}_N < t_0) \geq \max_i \frac{1}{g_i} \Lambda_A(-\theta_i)$ . Als  $\forall i, j$  geldt dat  $g_i = g_j$ , geldt ook dat  $\frac{1}{\max_i g_i} = \frac{1}{g_i} = \max_i \frac{1}{g_i}$ . Met Theorie

3 geeft dit:

$$\begin{aligned} \lim_{N \rightarrow \infty} \frac{1}{N} \log P(\hat{t}_N < t_0) &\leq \frac{1}{\max_i g_i} \max_j \Lambda_A(-\theta_j) \\ &= \frac{1}{g_j} \max_j \Lambda_A(-\theta_j) \\ &= \max_j \frac{1}{g_j} \Lambda_A(-\theta_j). \end{aligned}$$

Deze boven- en ondergrens samen geeft dat  $\lim_{N \rightarrow \infty} \frac{1}{N} \log P(\hat{t}_N < t_0) = \max_i \frac{1}{g_i} \Lambda_A(-\theta_i)$ .  $\square$

Wanneer alle halffabricaten dus even groot zijn, is het asymptotische gedrag gelijk aan het vermoeden. Er wordt eigenlijk weer alleen gekeken naar het aantal halffabricaten, waardoor dit hetzelfde resultaat is als in [1]. Het verschil is alleen dat er nu kan worden meegenomen hoeveel ruimte deze halffabricaten in zullen nemen.

Als de resultaten van de simulatie worden vergeleken met de waarden van de bewezen bovengrens, is deze in de limiet vele malen groter en helemaal niet meer in dezelfde orde van grootte. Het is ook een vreemde intuïtie dat de kans zich gedraagt naar de langzaamste server en de grootste grootte, terwijl deze misschien helemaal niet bij elkaar horen. Daarom blijft het vermoeden dat het asymptotisch gedrag zich gedraagt als de bewezen ondergrens.

## 6 Conclusie

De onderzoeksvraag van dit artikel was: Wat is de kans op een grote benodigde voorraadcapaciteit voor een fabriekshal met twee servers achter elkaar wanneer de halffabricaten in de wachtrijen niet even groot zijn? In de literatuur is al te vinden welk asymptotisch gedrag het aantal halffabricaten in het systeem heeft. Niet alleen voor een systeem met twee servers, maar voor een eindig aantal servers achter elkaar.

Als na de eerste server de halffabricaten een andere grootte krijgen dan ervoor, is het interessanter om te weten welke ruimte deze grote aantallen halffabricaten innemen. Hiervoor is met behulp van simulatie een vermoeden ontstaan voor het asymptotische gedrag van het aantal eenheden aan halffabricaten in de fabriekshal. Er is bewezen dat dit gedrag altijd tussen

een bepaalde boven- en ondergrens ligt. Ook dit is niet alleen gedaan voor twee, maar voor een eindig aantal servers achter elkaar. Deze ondergrens is gelijk aan het vermoeden. De bovengrens ligt er nog boven, maar in het geval dat de halffabricaten in alle wachtrijen even groot zijn, is deze ook gelijk aan het vermoeden.

Met dit onderzoek kan dus bepaald worden tussen welke waarden het asymptotische gedrag ligt van het aantal eenheden aan halffabricaten in een fabriekshal met twee, maar ook voor meer servers. Voor verder onderzoek zou er nog gekeken kunnen worden of de bovengrens misschien nog kleiner kan worden bewezen. Ook zou de simulatie kunnen worden verbeterd, waardoor er sneller grotere aantallen halffabricaten in het systeem aanwezig zijn. Daarnaast kan de simulatie worden uitgebreid naar meer dan twee servers om na te gaan of de waarden dan nog steeds in de buurt van het vermoeden liggen.

## Referenties

- [1] Anne Buijsrogge, Pieter-Tjerk de Boer, Karol Rosen, and Werner Scheinhardt. Large deviations for the total queue size in non-markovian tandem queues. *Queueing Systems*, pages 1–8, 2017.
- [2] Pieter-Tjerk De Boer. Analysis of state-independent importance-sampling measures for the two-node tandem queue. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 16(3):225–250, 2006.
- [3] Philip Heidelberger. Fast simulation of rare events in queueing and reliability models. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 5(1):43–85, 1995.
- [4] Shyam Parekh and Jean Walrand. A quick simulation method for excessive backlogs in networks of queues. *IEEE Transactions on Automatic Control*, 34(1):54–66, 1989.
- [5] Sheldon M Ross. *Introduction to probability models*. Academic press, 2007.
- [6] John S Sadowsky. Large deviations theory and efficient simulation of excessive backlogs in a GI/GI/m queue. *IEEE Transactions on Automatic Control*, 36(12):1383–1394, 1991.

## A Python code voor dubbele server

```
import random
import math

#een variabele uit verschillende verdelingen halen
def uitverdeling(verd):
    if verd[0] == 'uniform':
        X = random.uniform(verd[1], verd[2])
    elif verd[0] == 'exp':
        X = -verd[1] * math.log(random.uniform(0, 1))
    elif verd[0] == 'normaal':
        X = random.normalvariate(verd[1], verd[2])
    elif verd[0] == 'deterministisch':
        X = verd[1]
    elif verd[0] == 'hyper':
        u = random.uniform(0, 1)
        kans = verd[1]
        for i in range(int((len(verd)-1)/2)):
            if u < kans:
                mu = verd[i * 2 + 2]
                break
            else:
                kans += verd[i * 2 + 3]
        X = -mu * math.log(random.uniform(0,1))
    return X

#uitvoeren bij aankomst
def aankomst(T, B1, W1, tijden, verdA, verdB1):
    if B1 < T:
        B1 = T + uitverdeling(verdB1)
        tijden.append(B1)
    W1 += 1
    A = T + uitverdeling(verdA)
    tijden.append(A)
    return A, B1, W1, tijden

#uitvoeren bij einde bediening server 1
def bediening1 (T, B2, W1, W2, tijden, verdB1, verdB2):
```

```

W1 -= 1
W2 += 1
if W1 > 0:
    B1 = T + uitverdeling(verdB1)
    tijden.append(B1)
else:
    B1 = -1
if B2 < T:
    B2 = T + uitverdeling(verdB2)
    tijden.append(B2)
return B1, B2, W1, W2, tijden

#uitvoeren bij einde bediening server 2
def bediening2 (T, W, tijden , verdB2):
    W -= 1
    if W > 0:
        B2 = T + uitverdeling(verdB2)
        tijden.append(B2)
    else:
        B2 = -1
    return B2, W, tijden

#lijst van tijden updaten
def updateL_tijd(L, W, T, T_vorig):
    while W > len(L)-1:
        L.append(0)
    L[W] += T - T_vorig
    return L

#lijst van maximale lengte wachtrij updaten
def updateL_max(L, W):
    while W > len(L)-1:
        L.append(0)
    L[W] += 1
    return L

#initiele simulatie om tijdspanne te berekenen
def sim_per\
    (aant_per, g1, g2, verdA, verdB1, verdB2):
    T = 0

```

```

A = uitverdeling(verdA)
B1 = uitverdeling(verdB1)
B2 = 0
tijden = [A, B1]
W1 = 1
W2 = 0
W_max = W1*g1 + W2*g2
L_tijd = []
L_max = []
periode = 0
while periode < aant_per:
    T_vorig = T
    tijden.sort()
    T = tijden.pop(0)
    W = W1*g1 + W2*g2
    L_tijd = updateL_tijd(L_tijd, W, T, T_vorig)
    if W > W_max:
        W_max = W
    elif W == 0:
        L_max = updateL_max(L_max, W_max)
        periode += 1
        W_max = 0
    if T == A:
        A, B1, W1, tijden = \
aankomst(T, B1, W1, tijden, verdA, verdB1)
    elif T == B1:
        B1, B2, W1, W2, tijden = \
bediening1 (T, B2, W1, W2, tijden, verdB1, verdB2)
    else:
        B2, W2, tijden = \
bediening2 (T, W2, tijden, verdB2)
    return L_tijd, L_max, T

#simulatie zelf
def sim_tijd\
(tijd, L_max, g1, g2, verdA, verdB1, verdB2):
    T = 0
    A = uitverdeling(verdA)
    B1 = uitverdeling(verdB1)
    B2 = 0

```

```

    tijden = [A, B1]
    W1 = 1
    W2 = 0
    W_max = W1*g1 + W2*g2
    L_tijd = []
    while T < tijd:
        T_vorig = T
        tijden.sort()
        T = tijden.pop(0)
        W = W1*g1 + W2*g2
        L_tijd = updateL_tijd(L_tijd, W, T, T_vorig)
        if W > W_max:
            W_max = W
        elif W == 0:
            L_max = updateL_max(L_max, W_max)
            W_max = 0
        if T == A:
            A, B1, W1, tijden = \
aankomst(T, B1, W1, tijden, verdA, verdB1)
        elif T == B1:
            B1, B2, W1, W2, tijden = \
bediening1 (T, B2, W1, W2, tijden, verdB1, verdB2)
        else:
            B2, W2, tijden = \
bediening2 (T, W2, tijden, verdB2)
    return L_tijd, L_max

```

*#betrouwbaarheidsinterval voor L\_tijd berekenen*

```

def bi_tijd(uitk):
    max = 0
    for i in range(len(uitk)):
        if len(uitk[i]) > max:
            max = len(uitk[i])
    gem = []
    for i in range(max):
        som = 0
        for j in range(len(uitk)):
            if len(uitk[j]) > i:
                som += uitk[j][i]
        gem.append(som / len(uitk))

```

```

var = []
for i in range(max):
    som = 0
    for j in range(len(uitk)):
        if len(uitk[j]) > i:
            verschil = uitk[j][i] - gem[i]
            som += verschil*verschil
        else:
            som += gem[i]*gem[i]
    var.append(som / (len(uitk) - 1))
fm = []
for i in range(max):
    fm.append(1.96 * math.sqrt(var[i] / len(uitk)))
return gem, fm, max

```

*#betrouwbaarheidsinterval voor L\_max berekenen*

```

def bi_max(L_max):
    max = len(L_max)
    per = sum(L_max)
    gem = []
    var = []
    som = per
    for i in range(max):
        gem.append(som / per)
        v = 1 - gem[i]
        var.append((som*v*v + \
                    (per-som)*gem[i]*gem[i])/(per-1))
        som -= L_max[i]
    fm = []
    for i in range(max):
        fm.append(1.96 * math.sqrt(var[i] / per))
    return gem, fm, max, per

```

*#experiment uitvoeren*

```

def uitvoeren\
(aant_sim, aant_per, g1, g2, verdA, verdB1, verdB2):
    uitk_tijd = []
    L_max = []
    T = 0
    for i in range(aant_sim):

```



```

        if i == 0:
            L_tijd, L_max, T = \
sim_per(aant_per, g1, g2, verdA, verdB1, verdB2)
        else:
            L_tijd, L_max = \
sim_tijd(T, L_max, g1, g2, verdA, verdB1, verdB2)
            L_tijd_kansen = []
            totaal_tijd = sum(L_tijd)
            som_tijd = totaal_tijd
            for j in range(len(L_tijd)):
                L_tijd_kansen.append(som_tijd/totaal_tijd)
                som_tijd -= L_tijd[j]
            uitk_tijd.append(L_tijd_kansen)
            gem_tijd, fm_tijd, max_tijd = bi_tijd(uitk_tijd)
            gem_max, fm_max, max_max, periodes = bi_max(L_max)
            print('Max_rijlengte', max_max - 1, max_tijd - 1)
            print("Maximale_lengte_periode")
            print('gemiddelde', gem_max)
            print('foutmarge', fm_max)
            print("Stationaire_kansen")
            print('gemiddelde', gem_tijd)
            print('foutmarge', fm_tijd)
            return T, periodes

print(uitvoeren(10000, 1000, 1, 25, \
['exp', 5], ['exp', 4], ['exp', 2]))

```