BACHELOR

Analyse van een productielijn met buffers

Meyfroyt, P.H.A.

Award date:
2010

Analyse van een productielijn met buffers

P.H.A Meyfroyt

Begeleiders:
prof.dr.ir. I.J.B.F. Adan
dr. J.A.C. Resing

11 juni 2010

It is the mark of an educated mind to rest satisfied with the degree of precision which the nature of the subject admits and not to seek exactness where only an approximation is possible.

- Aristotle (384 BC - 322 BC)

Samenvatting

Dit verslag behandelt het probleem waarbij de output van een productielijn benaderd moet worden met behulp van een rekenprogramma. We gaan er hierbij van uit dat de lijn verbonden is aan een bron die constant voor input zorgt en dat het uiteinde nooit geblokkeerd is. Na iedere machine, behalve de laatste, bevindt zich ook een eindige buffer. Om de output van een dergelijke lijn te benaderen hebben we een rekenprogramma geconstrueerd dat de output van een lijn met twee machines kan berekenen. Om de output van een lijn met meer dan twee machines te berekenen kan dit programma op drie verschillende manieren herhaaldelijk op de lijn toegepast worden.

Methode 1: Machines worden van voor naar achter geaggregeerd.
Methode 2: Machines worden van achter naar voor geaggregeerd.
Methode 3: Een combinatie van methode 1 en 2 waarbij zowel van achter naar voor als van voor naar achter geaggregeerd wordt.

Met behulp van een simulatieprogramma hebben we kunnen bepalen welke van deze manieren het beste bij een bepaald type lijn past. Vervolgens hebben we de volgende punten kunnen concluderen:

Methode 1: werkt het beste bij een productielijn waarbij alle machines dezelfde snelheid hebben en de capaciteit van op één volgende buffers niet daalt.
Methode 2: werkt het beste bij een productielijn waarbij alle machines dezelfde snelheid hebben en de capaciteit van op één volgende buffers niet stijgt.
Methode 3: werkt het beste bij een productielijn waarbij iedere buffer dezelfde capaciteit heeft en lange productielijnen.
Alle methodes werken goed bij een productielijn met enkel grote buffers.
Alle methodes werken goed bij een productielijn die minstens één trage machine bevat.

Om de schatting van de output te verbeteren hebben we vervolgens een iteratie-methode ontwikkeld. Kort gezegd is deze methode niets anders dan het herhaaldelijk toepassen van Methode 1 terwijl de parameters van de geaggregeerde machines na iedere iteratie worden bijgesteld. We hebben tijdens het onderzoek ook geconstateerd dat deze iteratie-methode, methode 4, de schatting in veel gevallen beter maakt. Voornamelijk in het geval van lange productielijnen blijkt methode 4 het stukken beter te doen. Als alle machines in de lijn niet dezelfde snelheid hebben of als de lijn te grote of te kleine buffers bevat, dan zijn de andere methodes toch beter.

Inhoudsopgave

1 Inleiding

Het probleem dat in dit verslag behandeld wordt, komt oorspronkelijk van bierfabrikant Heineken. Na een onderzoek naar de efficiëntie van het productieproces bleek dat de buffers in de productielijn te groot waren. Heineken is hiermee naar de TU/e gestapt en heeft gevraagd of er een model gemaakt kon worden van de productielijn. In dit verslag wordt uitgelegd hoe zo'n model tot stand komt en op welke verschillende manieren de output ermee benaderd kan worden. Vervolgens wordt uiteengezet welke methode het beste werkt bij een bepaald type productielijn. Als laatste komt er een verbetering van het model aan bod.

2 2-machinesysteem

We beginnen het verslag met het onderzoeken van een productielijn van slechts twee machines. We gaan er hier van uit, en dit geldt voor het gehele verslag, dat een productielijn continu input krijgt en dat de laatste machine nooit geblokkeerd raakt. Iedere machine heeft een bepaalde productiesnelheid en is onderhevig aan een *up* en een *down* rate. Tussen de machines bevindt zich een buffer die gevuld wordt door de eerste machine en geleegd wordt door de tweede machine. In paragraaf 2.1 geven we eerst een algemene analyse die we later gaan specificeren tot twee verschillende situaties. Namelijk de situatie waarbij we te maken hebben met een tijdafhankelijke failurerate en de situatie waarbij we te maken hebben met een productieafhankelijke failurerate. Tijdafhankelijke failurerates zijn, zoals de benaming al doet vermoeden, enkel afhankelijk van de tijd. Productieafhankelijke failurerates zijn daar bovenop ook nog afhankelijk van het feit of de machine aan het produceren is. Dit betekent dat als de machine niet aan het produceren is deze ook niet *down* kan gaan. De tijdafhankelijke situatie zal aan bod komen in paragraaf 2.2, de productieafhankelijke situatie in paragraaf 2.3.

2.1 Analyse

Voor onze algemen analyse introduceren we eerst enkele variabelen.

K	buffercapaciteit
M_m	machine m
λ_m	rate waarmee machine m *down* gaat
μ_m	rate waarmee machine m *up* gaat
v_m	productiesnelheid van machine m
α_i	nettosnelheid in toestand i waarmee de buffer vult/leegloopt

Ten gevolge van de verschillende *up* en *down* rates kan het systeem zich in vier verschillende toestanden bevinden. Ieder van deze toestanden heeft een bepaalde α_i. In de volgende tabel staan de verschillende toestanden weergegeven met de bijbehorende α_i's. Met een plusteken in de kolom van M_m wordt bedoeld dat machine m *up* is. Een minteken betekent dat deze *down* is.

i	M_1	M_2	α_i
1	+	−	v_1
2	−	+	$-v_2$
3	−	−	0
4	+	+	$v_1 - v_2$

Vervolgens definïeren we de verdelingsfuncties $F_i(x,t)$. Deze stellen de kans voor dat op tijdstip t het systeem zich in toestand i bevindt en de bufferinhoud niet meer dan x is. Voor het komende gedeelte stelt λ_{ij} de overgangsrate voor van toestand i naar toestand j. Deze kunnen bepaald worden uit de *up* en *down* rates van de machines.

Voorbeeld:
Om van toestand 1, waar M_1 *up* en M_2 *down* is, naar toestand 3 te gaan waar zowel M_1 en M_2 *down* zijn, moet M_1 *down* gaan. λ_{13} is dus de rate waarmee M_1 *down* gaat, dus $\lambda_{13} = \lambda_1$

Om het verdere verloop van de analyse te vereenvoudigen definïeren we

$$\lambda_i = \sum_{j \neq i} \lambda_{ij}.$$

We kunnen nu de volgende vergelijking opstellen

$$F_i(x, t+\Delta) = F_i(x - \alpha_i\Delta, t)(1 - \lambda_i\Delta) + \sum_{j \neq i} F_j(x + O(\Delta), t)\lambda_{ji}\Delta + O(\Delta^2).$$

Als we dit vervolgens delen door Δ en herschrijven, krijgen we

$$\frac{F_i(x,t+\Delta)}{\Delta} - \frac{F_i(x-\alpha_i\Delta,t)}{\Delta} = \frac{-\lambda_i\Delta F_i(x-\alpha_i\Delta,t)}{\Delta} + \frac{\sum_{j\neq i} F_j(x+O(\Delta),t)\lambda_{ji}\Delta + O(\Delta^2)}{\Delta}.$$

Vervolgens laten we $\Delta \to 0$ en krijgen we

$$\frac{\partial F_i(x,t)}{\partial t} + \alpha_i \frac{\partial F_i(x,t)}{\partial x} = -\lambda_i F_i(x,t) + \sum_{j \neq i} F_j(x,t)\lambda_{ji}. \tag{1}$$

Omdat we geïnteresseerd zijn in de evenwichtsituatie laten we $t \to \infty$ en stellen we de eerste term in (1) gelijk aan nul. Als we vervolgens definïeren

$$F_i(x) = \lim_{t \to \infty} F_i(x,t)$$

kunnen we (1) herschrijven tot

$$\alpha_i \frac{dF_i(x)}{dx} = -\lambda_i F_i(x) + \sum_{j \neq i} F_j(x)\lambda_{ji},$$

ofwel in matrixnotatie

$$Df(x) = M^T F(x). \tag{2}$$

Hierbij is $f(x)$ een kolomvector met als i-de component $\frac{dF_i(x)}{dx}$ en $F(x)$ een kolomvector met als i-de component $F_i(x)$. Matrix D de diagonaalmatrix met de α_i's op de diagonaal. Matrix M is de Markovgenerator bestaande uit de overgangsrates behorende bij de situaties uit de tabel. We hebben nu een differentiaalvergelijking die we kunnen oplossen. De oplossingen zijn van de vorm

$$f_i(x) = \sum_j c_j \mu_{ij} e^{\sigma_j x}. \tag{3}$$

Hierbij is c_j een constante, μ_{ij} de i^e component van de j^e eigenvector van $D^{-1}M^T$ en σ_j de bijbehorende j^e eigenwaarde.

De μ_{ij}'s en bijbehorende σ_j's zijn eenvoudig te bepalen. Om de c_j's te achterhalen is het handig als we eerst naar de randvoorwaarden kijken. Om deze te kunnen bestuderen definieren we eerst de notatie $P_i(x)$. Deze stelt de kans voor dat in toestand i de buffer hoeveelheid x bevat. Als $\alpha_i > 0$ geldt dat $P_i(0) = 0$ omdat in toestand $(i,0)$ de buffer onmiddellijk begint vol te lopen. Het systeem kan zich echter wel enige tijd in toestand (i,K) bevinden als we er van uit gaan dat overtollige productie verloren gaat. De functie $F_i(x)$ heeft dus een sprong in $x = K$. Als $\alpha_i < 0$ geldt dat $P_i(0) > 0$ en dat de functie $F_i(x)$ continu is in $x = K$. Omdat de gevallen waarbij $x = K$ of $x = 0$ niet gedekt worden door differentiaalvergelijking (2), stellen we aparte vergelijkingen op voor deze randpunten. Deze zijn van de vorm

$$Df(K) = M^T P(K) \tag{4}$$
$$Df(0) = M^T P(0) \tag{5}$$

Waarbij $P(0)$ en $P(K)$ respectievelijk de vectoren zijn met $P_i(0)$ en $P_i(K)$ op de i-de component. Matrixvergelijkingen (4) en (5) bevatten genoeg vergelijkingen om alle c_i's, $P_i(K)$'s en $P_i(0)$'s te berekenen. We moeten hier bij wel rekening houden met het feit dat de som van alle kansen gelijk moet zijn aan 1. De c_i's, $P(K)_i$'s en $P(0)_i$'s moeten dus zo gekozen worden dat $F_1(K) + F_2(K) + F_3(K) + F_4(K) = 1$.

Met behulp van (3) en de inmiddels bekende c_i's, $P_i(K)$'s en $P_i(0)$'s, kunnen we de $F_i(K)$'s uitrekenen door middel van

$$\begin{aligned}
F_1(K) &= P_1(0) + \int_0^K f_1(y)dy + P_1(K) \\
F_2(K) &= P_2(0) + \int_0^K f_2(y)dy + P_2(K) \\
F_3(K) &= P_3(0) + \int_0^K f_3(y)dy + P_3(K) \\
F_4(K) &= P_4(0) + \int_0^K f_4(y)dy + P_4(K)
\end{aligned}$$

$F_i(K)$ stelt niets anders voor dan de kans dat het systeem zich in toestand i bevindt. Met behulp van $F_i(K)$ en de productiesnelheid van het gehele systeem in toestand i kunnen we de gemiddelde productiesnelheid van het systeem berekenen. In de komende paragrafen wordt hier meer over verteld.

2.2 Tijdafhankelijk failurerate

De matrix vergelijking (2) ziet er in de tijdafhankelijke situatie als volgt uit

$$\begin{pmatrix}\alpha_1 f_1(x)\\ \alpha_2 f_2(x)\\ \alpha_3 f_3(x)\\ \alpha_4 f_4(x)\end{pmatrix} = \begin{pmatrix}-(\lambda_1+\mu_2) & 0 & \lambda_1 & \mu_2\\ 0 & -(\mu_1+\lambda_2) & \lambda_2 & \mu_1\\ \mu_1 & \mu_2 & -(\mu_1+\mu_2) & 0\\ \lambda_2 & \lambda_1 & 0 & -(\lambda_1+\lambda_2)\end{pmatrix}^T \begin{pmatrix}F_1(x)\\ F_2(x)\\ F_3(x)\\ F_4(x)\end{pmatrix}$$

Omdat in toestand 3 beide machines stil staan geldt $\alpha_3 = 0$. We kunnen daarom de volgende vergelijking opstellen voor $F_3(x)$

$$F_3(x) = \frac{\lambda_1}{\mu_1+\mu_2}F_1(x) + \frac{\lambda_2}{\mu_1+\mu_2}F_2(x).$$

Deze vergelijking kunnen we vervolgens in de rest van de vergelijkingen substitueren. Als we dan ook nog alle α_i's invullen en hierdoor delen, komt de matrixvergelijking er als volgt uit te zien

$$\begin{pmatrix}f_1(x)\\ f_2(x)\\ f_4(x)\end{pmatrix} = \begin{pmatrix}\frac{\lambda_1\mu_1-(\lambda_1+\mu_2)(\mu_1+\mu_2)}{v_1(\mu_1+\mu_2)} & \frac{\lambda_2\mu_1}{v_1(\mu_1+\mu_2)} & \frac{\lambda_2}{v_1}\\ -\frac{\lambda_1\mu_2}{v_2(\mu_1+\mu_2)} & \frac{(\lambda_2+\mu_1)(\mu_1+\mu_2)-\lambda_2\mu_2}{v_2(\mu_1+\mu_2)} & -\frac{\lambda_1}{v_2}\\ \frac{\mu_2}{v_1-v_2} & \frac{\mu_1}{v_1-v_2} & -\frac{\lambda_1+\lambda_2}{v_1-v_2}\end{pmatrix}\begin{pmatrix}F_1(x)\\ F_2(x)\\ F_4(x)\end{pmatrix}$$

Zoals voorafgaand al is beschreven is de oplossing van vorm (3). Om de c_i's en de, in sommige toestanden, discontinue punten $P_i(K)$ en $P_i(0)$ te bepalen, maken we gebruik van (4) en (5). In de punten $x = 0$ en $x = K$ hebben respectievelijk (4) en (5) de volgende vorm

$$\begin{pmatrix}\alpha_1 f_1(x)\\ \alpha_2 f_2(x)\\ \alpha_3 f_3(x)\\ \alpha_4 f_4(x)\end{pmatrix} = \begin{pmatrix}-(\lambda_1+\mu_2) & 0 & \lambda_1 & \mu_2\\ 0 & -(\mu_1+\lambda_2) & \lambda_2 & \mu_1\\ \mu_1 & \mu_2 & -(\mu_1+\mu_2) & 0\\ \lambda_2 & \lambda_1 & 0 & -(\lambda_1+\lambda_2)\end{pmatrix}^T \begin{pmatrix}P_1(x)\\ P_2(x)\\ P_3(x)\\ P_4(x)\end{pmatrix}$$

Na het substitueren van

$$P_3(x) = \frac{\lambda_1}{\mu_1+\mu_2}P_1(x) + \frac{\lambda_2}{\mu_1+\mu_2}P_2(x),$$

α_3 is immers nul, en het delen door de α_i's, krijgen we (voor $x = 0$ en $x = K$)

$$\begin{pmatrix}f_1(x)\\ f_2(x)\\ f_4(x)\end{pmatrix} = \begin{pmatrix}\frac{\lambda_1\mu_1-(\lambda_1+\mu_2)(\mu_1+\mu_2)}{v_1(\mu_1+\mu_2)} & \frac{\lambda_2\mu_1}{v_1(\mu_1+\mu_2)} & \frac{\lambda_2}{v_1}\\ -\frac{\lambda_1\mu_2}{v_2(\mu_1+\mu_2)} & \frac{(\lambda_2+\mu_1)(\mu_1+\mu_2)-\lambda_2\mu_2}{v_2(\mu_1+\mu_2)} & -\frac{\lambda_1}{v_2}\\ \frac{\mu_2}{v_1-v_2} & \frac{\mu_1}{v_1-v_2} & -\frac{\lambda_1+\lambda_2}{v_1-v_2}\end{pmatrix}\begin{pmatrix}P_1(x)\\ P_2(x)\\ P_4(x)\end{pmatrix}$$

We kunnen de rechterkant van deze gelijkheid vervolgens gelijk stellen aan de oplossing van de differentiaalvergelijking in de punten $x = 0$ en $x = K$. Rekeninghoudend met de normeringsvergelijking die er voor moet zorgen dat de som van alle $F_i(K)$'s gelijk moet zijn aan 1, kunnen we de c_i's berekenen. In de tijdafhankelijke situatie moeten we ook nog onderscheid maken tussen de gevallen $v_1 > v_2$, $v_1 < v_2$ en $v_1 = v_2$. We vervolgen bovenstaande analyse voor respectievelijk deze situaties.

2.2.1 Situatie 1: $v_1 > v_2$

De randvoorwaarden bij de situatie $v_1 > v_2$ die we zonder berekeningen kunnen bepalen zijn

$$\begin{aligned} P_1(0) &= 0 \\ P_2(K) &= 0 \\ P_4(0) &= 0 \end{aligned}$$

Als we dit invullen in de eerdergenoemde matrixvergelijking

$$\begin{pmatrix} f_1(x) \\ f_2(x) \\ f_4(x) \end{pmatrix} = \begin{pmatrix} \frac{\lambda_1\mu_1-(\lambda_1+\mu_2)(\mu_1+\mu_2)}{v_1(\mu_1+\mu_2)} & \frac{\lambda_2\mu_1}{v_1(\mu_1+\mu_2)} & \frac{\lambda_2}{v_1} \\ -\frac{\lambda_1\mu_2}{v_2(\mu_1+\mu_2)} & \frac{(\lambda_2+\mu_1)(\mu_1+\mu_2)-\lambda_2\mu_2}{v_2(\mu_1+\mu_2)} & -\frac{\lambda_1}{v_2} \\ \frac{\mu_2}{v_1-v_2} & \frac{\mu_1}{v_1-v_2} & -\frac{\lambda_1+\lambda_2}{v_1-v_2} \end{pmatrix} \begin{pmatrix} P_1(x) \\ P_2(x) \\ P_4(x) \end{pmatrix}$$

en de rechterkant van de vergelijking gelijkstellen aan de uitkomst van de differentiaalvergelijking, krijgen we het volgende stelsel

$$\begin{aligned} \frac{\lambda_1\mu_1-(\lambda_1+\mu_2)(\mu_1+\mu_2)}{v_1(\mu_1+\mu_2)}P_1(K) + \frac{\lambda_2}{v_1}P_4(K) &= c_1u_{11}e^{\sigma_1K} + c_2u_{12}e^{\sigma_2K} + c_4u_{14}e^{\sigma_4K} \\ -\frac{\lambda_1\mu_2}{v_2(\mu_1+\mu_2)}P_1(K) - \frac{\lambda_1}{v_2}P_4(K) &= c_1u_{21}e^{\sigma_1K} + c_2u_{22}e^{\sigma_2K} + c_4u_{24}e^{\sigma_4K} \\ \frac{\mu_2}{v_1-v_2}P_1(K) - \frac{\lambda_1+\lambda_2}{v_1-v_2}P_4(K) &= c_1u_{41}e^{\sigma_1K} + c_2u_{42}e^{\sigma_2K} + c_4u_{44}e^{\sigma_4K} \\ \frac{\lambda_2\mu_1}{v_1(\mu_1+\mu_2)}P_2(0) &= c_1u_{11} + c_2u_{12} + c_4u_{14} \\ \frac{(\lambda_2+\mu_1)(\mu_1+\mu_2)-\lambda_2\mu_2}{v_2(\mu_1+\mu_2)}P_2(0) &= c_1u_{21} + c_2u_{22} + c_4u_{24} \\ \frac{\mu_1}{v_1-v_2}P_2(0) &= c_1u_{41} + c_2u_{42} + c_4u_{44} \\ \sum_1^4 \int_0^\infty f_i(x)dx + \sum_1^4 P_i(0) + \sum_1^4 P_i(K) &= 1 \end{aligned} \tag{6}$$

De enige onbekenden waar we in deze vergelijkingen mee te maken hebben zijn: $c_1, c_2, c_3, P_1(K), P_4(K)$ en $P_2(0)$. Deze kunnen we vinden door het bovenstaande stelsel op te lossen.

Na het vinden van alle $P_i(K)$'s, $P_i(0)$'s, c_i's, en daarmee de f_i's, kunnen we de $F_i(K)$'s vinden. Deze zijn van de volgende vorm

$$\begin{aligned} F_1(K) &= \int_0^K f_1(y)dy + P_1(K) \\ F_2(K) &= \int_0^K f_2(y)dy + P_2(0) \\ F_4(K) &= \int_0^K f_4(y)dy + P_4(K) \end{aligned}$$

Zoals eerder al is vermeld is $F_i(K)$ niets anders dan de kans dat het systeem zich in toestand i bevindt. We kunnen dit ook beschouwen als de fractie van de tijd dat het systeem zich in toestand i bevindt. De output van het systeem is te bereken doormiddel van te sommeren over de output van de verschillende toestanden vermenigvuldigd met de fractie van de tijd dat het systeem zich in deze situatie bevindt. Ofwel in het geval $v_1 > v_2$

$$T = F_1(K)\cdot 0 + (F_2(K) - P_2(0))v_2 + F_3(K)\cdot 0 + F_4(K)v_2.$$

Wat gelijk is aan

$$T = (F_2(K) - P(0))v_2 + F_4(K)v_2.$$

2.2.2 Situatie 2: $v_1 < v_2$

De randvoorwaarden bij de situatie $v_1 < v_2$ die we zonder berekeningen kunnen bepalen zijn

$$\begin{aligned} P_1(0) &= 0 \\ P_2(K) &= 0 \\ P_4(K) &= 0 \end{aligned}$$

Op dezelfde manier als in de voorgaande paragraaf krijgen we het volgende stelsel vergelijkingen

$$\begin{aligned} \frac{\lambda_1\mu_1 - (\lambda_1+\mu_2)(\mu_1+\mu_2)}{v_1(\mu_1+\mu_2)}P_1(K) &= c_1u_{11}e^{\sigma_1 K} + c_2u_{12}e^{\sigma_2 K} + c_4u_{14}e^{\sigma_4 K} \\ -\frac{\lambda_1\mu_2}{v_2(\mu_1+\mu_2)}P_1(K) &= c_1u_{21}e^{\sigma_1 K} + c_2u_{22}e^{\sigma_2 K} + c_4u_{24}e^{\sigma_4 K} \\ \frac{\mu_2}{v_1-v_2}P_1(K) &= c_1u_{41}e^{\sigma_1 K} + c_2u_{42}e^{\sigma_2 K} + c_4u_{44}e^{\sigma_4 K} \\ \frac{\lambda_2\mu_1}{v_1(\mu_1+\mu_2)}P_2(0) + \frac{\lambda_2}{v_1}P_4(0) &= c_1u_{11} + c_2u_{12} + c_4u_{14} \\ \frac{(\lambda_2+\mu_1)(\mu_1+\mu_2)-\lambda_2\mu_2}{v_2(\mu_1+\mu_2)}P_2(0) - \frac{\lambda_1}{v_2}P_4(0) &= c_1u_{21} + c_2u_{22} + c_4u_{24} \\ \frac{\mu_1}{v_1-v_2}P_2(0) - \frac{\lambda_1+\lambda_2}{v_1-v_2}P_4(0) &= c_1u_{41} + c_2u_{42} + c_4u_{44} \\ \sum_1^4 \int_0^\infty f_i(x)dx + \sum_1^4 P_i(0) + \sum_1^4 P_i(K) &= 1 \end{aligned}$$

De enige onbekenden waar we in deze vergelijkingen mee te maken hebben zijn: $c_1, c_2, c_3, P_1(K), P_4(0)$ en $P_2(0)$. Deze kunnen we vinden door het bovenstaande stelsel op te lossen.

Na het vinden van alle c_i's, en daarmee de f_i's, de $P_i(K)$'s en $P_i(0)$'s kunnen we de $F_i(K)$'s vinden. Deze zijn van de volgende vorm

$$\begin{aligned} F_1(K) &= \int_0^K f_1(y)dy + P_1(K) \\ F_2(K) &= \int_0^K f_2(y)dy + P_2(0) \\ F_4(K) &= \int_0^K f_4(y)dy + P_4(0) \end{aligned}$$

Tenslotte berekenen we de output.

$$T = F_1(K)\cdot 0 + (F_2(K) - P_2(0))v_2 + F_3(K)\cdot 0 + (F_4(K) - P_4(0))v_2 + P_4(0)v_1.$$

Wat gelijk is aan

$$T = (F_2(K) - P_2(0))v_2 + (F_4(K) - P_4(0))v_2 + P_4(0)v_1.$$

2.2.3 Situatie 3: $v_1 = v_2$

In de situatie waarbij $v_1 = v_2$ geldt er

$$F_3(x) = \frac{\lambda_1}{\mu_1+\mu_2}F_1(x) + \frac{\lambda_2}{\mu_1+\mu_2}F_2(x)$$
$$F_4(x) = \frac{\mu_2}{\lambda_1+\lambda_2}F_1(x) + \frac{\mu_1}{\lambda_1+\lambda_2}F_2(x).$$

Als we dit in de Markovgenerator substitueren, krijgen we

$$\begin{pmatrix}\alpha_1 f_1(x)\\ \alpha_2 f_2(x)\end{pmatrix} = \begin{pmatrix}-(\lambda_1+\mu_2) + \frac{\lambda_1\mu_1}{\mu_1+\mu_2} + \frac{\mu_2\lambda_2}{\lambda_1+\lambda_2} & \frac{\lambda_2\mu_1}{\mu_1+\mu_2} + \frac{\mu_1\lambda_2}{\lambda_1+\lambda_2}\\ \frac{\lambda_1\mu_2}{\mu_1+\mu_2} + \frac{\mu_2\lambda_1}{\lambda_1+\lambda_2} & -(\lambda_2+\mu_1) + \frac{\lambda_2\mu_2}{\mu_1+\mu_2} + \frac{\mu_1\lambda_1}{\lambda_1+\lambda_2}\end{pmatrix}\begin{pmatrix}F_1(x)\\ F_2(x)\end{pmatrix}$$

We weten dat in deze situatie geldt

$$\begin{aligned}P_1(0) &= 0\\ P_2(K) &= 0\end{aligned}$$

Net zoals bij de voorgaande situaties kunnen we een stelsel vergelijkingen opstellen waaruit we de $P_1(K), P_2(0), c_1$ en c_2 kunnen berekenen. Dit stelsel ziet er als volgt uit

$$\begin{aligned}(-(\lambda_1+\mu_2) + \frac{\lambda_1\mu_1}{\mu_1+\mu_2} + \frac{\mu_2\lambda_2}{\lambda_1+\lambda_2})P_1(K) &= c_1u_{11}e^{\sigma_1 K} + c_2u_{12}e^{\sigma_2 K}\\ (\frac{\lambda_1\mu_2}{\mu_1+\mu_2} + \frac{\mu_2\lambda_1}{\lambda_1+\lambda_2})P_1(K) &= c_1u_{21}e^{\sigma_1 K} + c_2u_{22}e^{\sigma_2 K}\\ (\frac{\lambda_2\mu_1}{\mu_1+\mu_2} + \frac{\mu_1\lambda_2}{\lambda_1+\lambda_2})P_2(0) &= c_1u_{11} + c_2u_{12}\\ (-(\lambda_2+\mu_1) + \frac{\lambda_2\mu_2}{\mu_1+\mu_2} + \frac{\mu_1\lambda_1}{\lambda_1+\lambda_2})P_2(0) &= c_1u_{21} + c_2u_{22}\\ \sum_1^4\int_0^\infty f_i(x)dx + \sum_1^4 P_i(0) + \sum_1^4 P_i(K) &= 1\end{aligned}$$

Vervolgens berekenen we

$$\begin{aligned}F_1(K) &= \int_0^K f_1(y)dy + P_1(K)\\ F_2(K) &= \int_0^K f_2(y)dy + P_2(0)\end{aligned}$$

en tenslotte de output doormiddel van

$$T = (F_2(K) - P_2(0))v_2 + F_4(K)v_2$$

2.3 Productieafhankelijk failurerate

De failurerates veranderen als de rates waarmee de machines *down* gaan productieafhankelijk zijn. Productieafhankelijk betekent namelijk dat de machine enkel *down* kan gaan als het iets te doen heeft. Als in toestand 1 de buffer vol zit betekent dit dat M_1 geblokkeerd wordt en er in deze situatie dus geldt $\lambda_1 = 0$. Als in toestand 2 de buffer leeg is, heeft M_2 niks te doen. Hieruit volgt dat in deze situatie $\lambda_2 = 0$. Dit betekent dat de vergelijkingen van de randcondities veranderen.

2.3.1 Situatie 1: $v_1 > v_2$

Als we de nieuwe λ_i's gebruiken bij het stelsel vergelijkingen behorende bij de situatie $v_1 > v_2$, krijgen we het volgende stelsel

$$\begin{aligned}
\frac{-\mu_2}{v_1}P_1(K) + \frac{\lambda_2}{v_1}P_4(K) &= c_1u_{11}e^{\sigma_1 K} + c_2u_{12}e^{\sigma_2 K} + c_4u_{14}e^{\sigma_4 K} \\
-\frac{\lambda_1\mu_2}{v_2(\mu_1+\mu_2)}P_1(K) - \frac{\lambda_1}{v_2}P_4(K) &= c_1u_{21}e^{\sigma_1 K} + c_2u_{22}e^{\sigma_2 K} + c_4u_{24}e^{\sigma_4 K} \\
\frac{\mu_2}{v_1-v_2}P_1(K) - \frac{\lambda_1+\lambda_2}{v_1-v_2}P_4(K) &= c_1u_{41}e^{\sigma_1 K} + c_2u_{42}e^{\sigma_2 K} + c_4u_{44}e^{\sigma_4 K} \\
\frac{\lambda_2\mu_1}{v_1(\mu_1+\mu_2)}P_2(0) &= c_1u_{11} + c_2u_{12} + c_4u_{14} \\
\frac{\mu_1}{v_2}P_2(0) &= c_1u_{21} + c_2u_{22} + c_4u_{24} \\
\frac{\mu_1}{v_1-v_2}P_2(0) &= c_1u_{41} + c_2u_{42} + c_4u_{44} \\
\sum_1^4 \int_0^\infty f_i(x)dx + \sum_1^4 P_i(0) + \sum_1^4 P_i(K) &= 1
\end{aligned}$$

Voor de rest gaat het berekenen van de output op dezelfde manier als in de voorgaande paragrafen is besproken. Uit het bovenstaande stelsel lossen we $c_1, c_2, c_3, P_1(K), P_4(K)$ en $P_2(0)$ op. Hiermee hebben we ook alle f_i's gevonden. Vervolgens berekenen we de $F_i(K)$'s doormiddel van

$$\begin{aligned}
F_1(K) &= \int_0^K f_1(y)dy + P_1(K) \\
F_2(K) &= \int_0^K f_2(y)dy + P_2(0) \\
F_4(K) &= \int_0^K f_4(y)dy + P_4(K)
\end{aligned}$$

en tenslotte berekenen we de output doormiddel van

$$T = (F_2(K) - P(0))v_2 + F_4(K)v_2.$$

2.3.2 Situatie 2: $v_1 < v_2$

Als we de nieuwe λ_i's toepassen in de situatie $v_1 < v_2$, krijgen we

$$\begin{aligned}
\frac{\mu_2}{v_1}P_1(K) &= c_1u_{11}e^{\sigma_1 K} + c_2u_{12}e^{\sigma_2 K} + c_4u_{14}e^{\sigma_4 K} \\
-\frac{\lambda_1\mu_2}{v_2(\mu_1+\mu_2)}P_1(K) &= c_1u_{21}e^{\sigma_1 K} + c_2u_{22}e^{\sigma_2 K} + c_4u_{24}e^{\sigma_4 K} \\
\frac{\mu_2}{v_1-v_2}P_1(K) &= c_1u_{41}e^{\sigma_1 K} + c_2u_{42}e^{\sigma_2 K} + c_4u_{44}e^{\sigma_4 K} \\
\frac{\lambda_2\mu_1}{v_1(\mu_1+\mu_2)}P_2(0) + \frac{\lambda_2}{v_1}P_4(0) &= c_1u_{11} + c_2u_{12} + c_4u_{14} \\
\frac{\mu_1}{v_2}P_2(0) - \frac{\lambda_1}{v_2}P_4(0) &= c_1u_{21} + c_2u_{22} + c_4u_{24} \\
\frac{\mu_1}{v_1-v_2}P_2(0) - \frac{\lambda_1+\lambda_2}{v_1-v_2}P_4(0) &= c_1u_{41} + c_2u_{42} + c_4u_{44} \\
\sum_1^4 \int_0^\infty f_i(x)dx + \sum_1^4 P_i(0) + \sum_1^4 P_i(K) &= 1
\end{aligned}$$

Uit het bovenstaande stelsel berekenen we $c_1, c_2, c_3, P_1(K), P_4(0)$ en $P_2(0)$. Hiermee hebben we ook alle f_i's gevonden. Vervolgens berekenen we de $F_i(K)$'s door middel van

$$\begin{aligned}
F_1(K) &= \int_0^K f_1(y)dy + P_1(K) \\
F_2(K) &= \int_0^K f_2(y)dy + P_2(0) \\
F_4(K) &= \int_0^K f_4(y)dy + P_4(0)
\end{aligned}$$

Tenslotte berekenen we de output

$$T = (F_2(K) - P_2(0))v_2 + (F_4(K) - P_4(0))v_2 + P_4(0)v_1.$$

2.3.3 Situatie 3: $v_1 = v_2$

Na het toepassen van de nieuwe λ_i's in de situatie $v_1 = v_2$, reduceren de vergelijkingen tot

$$\begin{aligned}
0 &= c_1u_{11}e^{\sigma_1 K} + c_2u_{12}e^{\sigma_2 K} \\
(\lambda_1 * \frac{\mu_2}{\mu_1+\mu_2} + \mu_2 * \frac{\lambda_1}{\lambda_1+\lambda_2})P_1(K) &= c_1u_{21}e^{\sigma_1 K} + c_2u_{22}e^{\sigma_2 K} \\
(\lambda_2 * \frac{\mu_1}{\mu_1+\mu_2} + \mu_1 * \frac{\lambda_2}{\lambda_1+\lambda_2})P_2(0) &= c_1u_{11} + c_2u_{12} \\
0 &= c_1u_{21} + c_2u_{22} \\
\sum_1^4 \int_0^\infty f_i(x)dx + \sum_1^4 P_i(0) + \sum_1^4 P_i(K) &= 1
\end{aligned}$$

Vervolgens berekenen we weer

$$\begin{aligned}
F_1(K) &= \int_0^K f_1(y)dy + P_1(K) \\
F_2(K) &= \int_0^K f_2(y)dy + P_2(0)
\end{aligned}$$

en tenslotte de output

$$T = (F_2(K) - P_2(0))v_2 + F_4(K)v_2.$$

2.4 Voorbeeld

We zullen bovenstaande theorie proberen te verduidelijken met een voorbeeld. Stel we hebben een tijdsafhankelijke productielijn bestaande uit twee machines met daar tussen een buffer van grootte 5. Verder geldt voor het systeem het volgende

	M_1	M_2
v_i	8	6
λ_i	$\frac{1}{10}$	$\frac{1}{12}$
μ_i	$\frac{1}{2}$	$\frac{1}{2}$

Als we deze waarden invullen in (2) en beiden kanten vermenigvuldigen met D^{-1}, krijgen we de volgende matrixvergelijking

$$\begin{pmatrix} f_1(x) \\ f_2(x) \\ f_4(x) \end{pmatrix} = \begin{pmatrix} -0.0687 & 0.0052 & 0.0104 \\ -0.0083 & 0.0903 & -0.0167 \\ 0.2500 & 0.2500 & -0.0917 \end{pmatrix} \begin{pmatrix} F_1(x) \\ F_2(x) \\ F_4(x) \end{pmatrix}$$

Het berekenen van de eigenvectoren geeft ons

$$u_1 = \begin{pmatrix} 0.1931 \\ -0.0688 \\ -0.9788 \end{pmatrix}, u_2 = \begin{pmatrix} 0.1613 \\ 0.1935 \\ 0.9677 \end{pmatrix}, u_3 = \begin{pmatrix} 0.0950 \\ 0.4265 \\ 0.8995 \end{pmatrix}$$

Behorende bij de eigenwaarden

$$\sigma_1 = -0.1234, \sigma_2 = 0, \sigma_3 = 0.0533.$$

Stelsel (6) kunnen we nu schrijven als

$$\begin{pmatrix} -0.0687 \\ -0.0083 \\ 0.2500 \end{pmatrix} P_1(K) + \begin{pmatrix} 0.0104 \\ -0.0167 \\ -0.0917 \end{pmatrix} P_4(K) = c_1 \begin{pmatrix} 0.1042 \\ -0.0371 \\ -0.5281 \end{pmatrix} + c_2 \begin{pmatrix} 0.1613 \\ 0.1935 \\ 0.9677 \end{pmatrix} + c_3 \begin{pmatrix} 0.1240 \\ 0.5566 \\ 1.1740 \end{pmatrix}$$

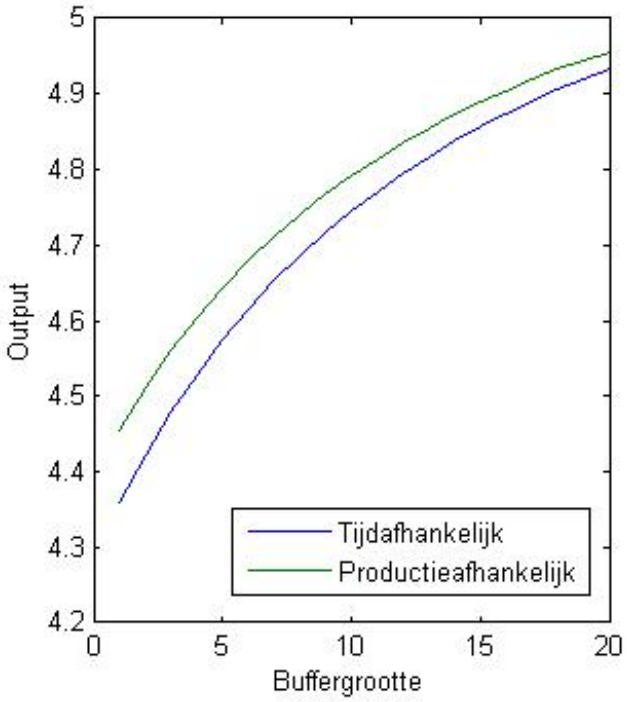
$$\begin{pmatrix} 0.0052 \\ 0.0903 \\ 0.2500 \end{pmatrix} P_2(0) = c_1 \begin{pmatrix} 0.1931 \\ -0.0688 \\ -0.9788 \end{pmatrix} + c_2 \begin{pmatrix} 0.1613 \\ 0.1935 \\ 0.9677 \end{pmatrix} + c_3 \begin{pmatrix} 0.0950 \\ 0.4265 \\ 0.8995 \end{pmatrix}$$

We hebben nu een stelsel van 6 vergelijkingen met 6 onbekenden dat we kunnen oplossen. Als we rekening houden met de normeringsvergelijking bij het oplossen van het stelsel, vinden we dat de productiesnelheid gelijk is aan 4.5742.

Als we de theorie van de productieafhankelijke situatie toepassen vinden we dat de productiesnelheid gelijk is aan 4.6433.

De volgende grafiek laat duidelijk het verschil zien in de tijdafhankelijke situatie en de productieafhankelijke situatie bij verschillende grootte van de buffer. Op de x-as staat de grootte van de buffer weergegeven. Op de y-as de productiesnelheid.

Voor de rest van het verslag gaan we er van uit dat de failurerates productieafhankelijk zijn.



Figuur 1: Output bij tijd,- en productieafhankelijke failurerate

3 N-machinesysteem

We zouden onze analyse graag willen uitbreiden naar een N-machinesysteem waarbij N het aantal machines is en $N > 2$. Zoals eerder vermeld gaan we er nu enkel nog van uit dat de failurerates productieafhankelijke zijn. De strategie die we gaan gebruiken is het opdelen van de productielijn en vervolgens ieder deel als een aparte machine beschouwen. Voordat we het gaan hebben over de verschillende manieren waarop men een productielijn kan opdelen, beginnen we eerst met een analyse van een 3-machinesysteem.

3.1 Analyse

We gaan uit van een 3-machinesysteem met een buffer na de eerste en de tweede machine. Om de theorie uit te leggen nemen we een voor de hand liggende opdeling van de lijn. We onderzoeken de productielijn door de eerste twee machines als één machine te beschouwen. Zo kunnen we de situatie reduceren naar een 2-machinesysteem. We maken in dit hoofdstuk ook onderscheid tussen *machine* en *systeem*. Bij een *machine* wordt namelijk één enkele machine bedoeld, bij een *systeem* de twee machines die samen als één machine beschouwd worden. Verder kent het *systeem* enkel twee toestanden. Toestand d betekend dat het *systeem* geen output heeft (dus *down* is) en toestand u dat er wel output is, ofwel: *up*.

We beginnen met het introduceren van enkele variabelen.

$r_{i,j}$	aantal sprongen van het *systeem* per tijdseenheid van toestand i naar toestand j
Π_i	fractie van de tijd dat het systeem in toestand i zit
λ	rate waarmee het *systeem down* gaat
μ	rate waarmee het *systeem up* gaat
v	productiesnelheid van het *systeem*
λ_m	rate waarmee *machine i down* gaat
μ_m	rate waarmee *machine i up* gaat
v_m	productiesnelheid van *machine i*

Om de eerste twee machines als één machine te kunnen beschouwen, moeten we de μ en de λ van dit systeem te weten komen. Bekend is dat

$$\frac{\frac{1}{\mu}}{\frac{1}{\mu}+\frac{1}{\lambda}} = \Pi_d \tag{7}$$

We moeten dus maar één van de twee zien te vinden. In dit geval berekenen we μ eerst. Dit doen we met behulp van de volgende gelijkheid

$$r_{d,u} = \Pi_d \mu \tag{8}$$

Het aantal keer dat het systeem namelijk gemiddeld per tijdseenheid van *down* naar *up* springt, is gelijk aan de fractie van de tijd dat het systeem *down* is maal de rate waarin het systeem van *down* naar *up* gaat. De fractie van de tijd dat het systeem geen output heeft, Π_d, kunnen we als volgt berekenen

$$\Pi_d = F_1(K) + F_3(K) + P_2(0)$$

In situatie 1 en 3 is machine 2 namelijk *down*. In situatie 2 is machine 1 *down* en 2 *up*. Is de buffer echter leeg, dan hebben we geen output. Het aantal sprongen per tijdseenheid van toestand d naar u berekenen we als volgt

$$r_{d,u} = F_1(K)\mu_2 + (F_3(K) - P_3(0))\mu_2 + P_2(0)\mu_1$$

Als de machines namelijk in toestand 1 verkeren, machine 1 *up* en machine 2 *down*, en machine 2 gaat *up*, dan gaat het gehele systeem *up*. Verkeren de machines in toestand 3, allebei *down*, dan gaat het systeem *up* als machine 2 *up* gaat en de buffer niet leeg is. Tenslotte kan het systeem nog *up* gaan als in toestand 2, machine 1 *down* en machine 2 *up*, de buffer leeg is en machine 1 *up* gaat.

Met behulp van (7) en (8) kunnen we nu de μ en λ berekenen van het systeem, bestaande uit de eerste twee machines. Met behulp van de theorie van de voorgaande hoofdstukken kunnen we vervolgens de output berekenen van dit systeem. Als we deze output conditioneren op het gegeven dat het systeem *up* is, krijgen we de snelheid van het systeem, v. Deze wordt berekend met

$$v = \frac{(F_2(K) - P_2(0))v_2 + (F_4(K) - P_4(0))v_2 + P_4(0)v_1}{F_2(K) - P_2(0) + F_4(K)}. \tag{9}$$

Merk op dat als $v_1 > v_2$, waarbij $P_4(0) = 0$, (9) reduceert tot v_2. Nu we λ, μ en v van het systeem gevonden hebben, weten we alles om de eerste twee machines als één te beschouwen.

In de komende paragrafen komen drie methodes aanbod waarbij bovenstaande theorie op verschillende manieren wordt toegepast. Met ieder van deze methodes kan de output van een lijn, waarbij $N > 2$, benaderd worden.

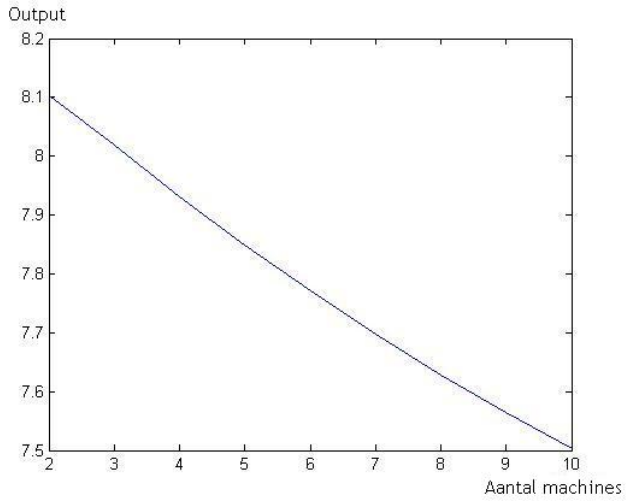
3.2 Methode 1

De eerste methode gaat zoals beschreven in de analyse. We beginnen met het samennemen van de eerste twee machines en beschouwen dit vervolgens als één machine. Vervolgens nemen we dit systeem samen met de derde machine in de lijn etc.

We beschouwen de volgende situatie:

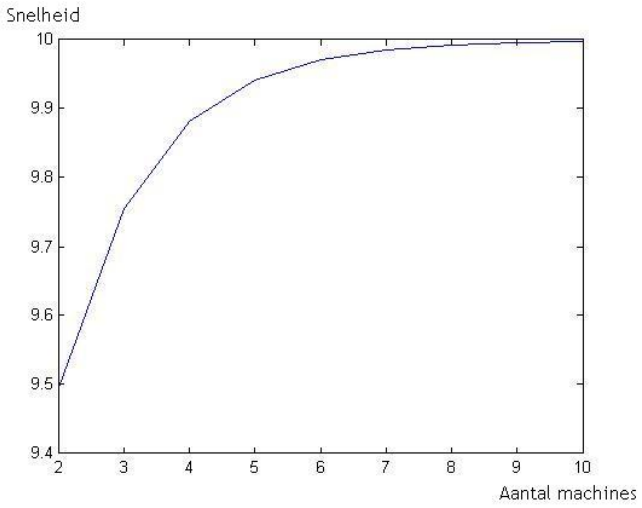
	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}
λ	1/10	1/10	1/10	1/10	1/10	1/10	1/10	1/10	1/10	1/10
μ	1	1	1	1	1	1	1	1	1	1
v	9	10	10	10	10	10	10	10	10	10

Na iedere machine, tot en met de voorlaatste, bevindt zich een buffer ter grootte 30 en alle machines zijn productieafhankelijk. De twee figuren die volgen zijn respectievelijk de snelheid en de output van het totale systeem bij het samennemen van een x aantal machines volgens methode 1. In de tabel die op iedere figuur volgt staan de waarden die zijn gebruikt bij het plotten van de desbetreffende figuur.



Figuur 2: Output bij methode 1

x	2	3	4	5	6	7	8	9	10
Output	8,1041	8,0179	7,9313	7,8483	7,7703	7,6972	7,6288	7,5645	7,5039



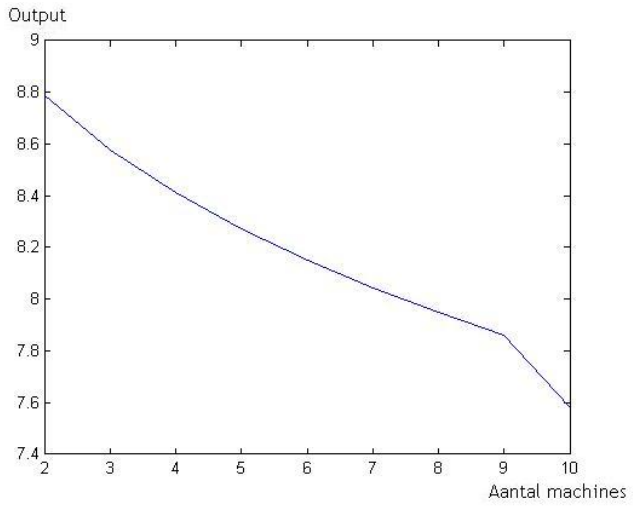
Figuur 3: Snelheid bij methode 1

x	2	3	4	5	6	7	8	9	10
Snelheid	9,4926	9,7549	9,8818	9,9419	9,9707	9,9849	9,992	9,9957	9.9976

Opvallend is dat de snelheid toeneemt naarmate je meer machines samen neemt. Dit lijkt tegenstrijdig maar is eenvoudig te verklaren. In dit voorbeeld produceert de volledige productielijn, gegeven dat hij *up* is, alleen met snelheid 9 als alle buffers leeg zijn. Anders produceert deze met snelheid 10. Dus de gemiddelde snelheid van de lijn (gegeven dat de lijn *up* is) is: 9(gem. tijd dat alle buffers leeg zijn) + 10(1−gem. tijd dat alle buffers leeg zijn). Hoe meer buffers er in een lijn zitten hoe kleiner de kans dat ze allemaal leeg zijn. De snelheid neemt dus toe.

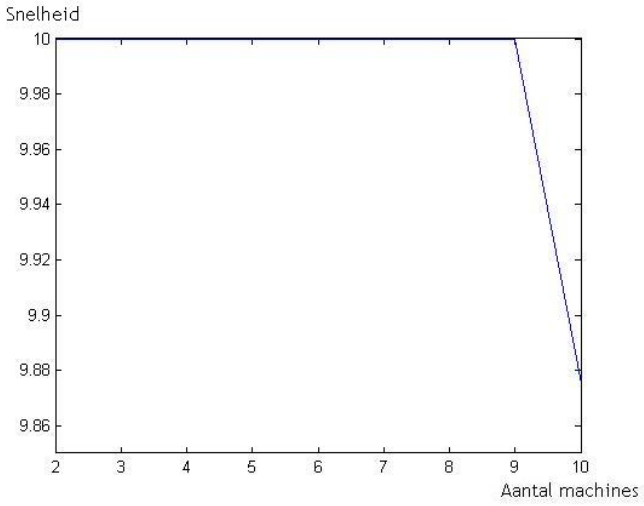
3.3 Methode 2

Een andere manier om de output van een productielijn te bepalen is door van achter naar voor te aggregeren. We kunnen de laatste en voorlaatste machine beschouwen als één machine en dit systeem vervolgens samennemen met de 3e laatste. Op deze manier kunnen we terugwerken tot aan machine 1. Als we dezelfde productielijn beschouwen als bij het voorbeeld van methode 1 krijgen we de hier op volgende grafieken. In de tabel die op iedere figuur volgt staan de waarden die zijn gebruikt bij het plotten van de desbetreffende figuur.



Figuur 4: Output bij methode 2

x	2	3	4	5	6	7	8	9	10
Output	8.7893	8.5781	8.4112	8.2717	8.1510	8.0442	7.9482	7.8609	7.5797



Figuur 5: Snelheid bij methode 2

x	2	3	4	5	6	7	8	9	10
Output	10	10	10	10	10	10	10	10	9.8751

3.4 Methode 3

Methode 3 is een combinatie van de twee voorgaande methodes. We passen methode 1 toe op de eerste helft van de productielijn en we passen methode 2 toe op de tweede helft van de productielijn zodat we beide delen als één machine kunnen beschouwen. Nu hebben we zogenaamd twee machines waarvan we de gezamenlijke output kunnen berekenen. In het voorgaande voorbeeld zou dit resulteren in een snelheid van 9.9854 en een output van 7.5226.

4 Simulatie

In dit hoofdstuk vergelijken we de resultaten van voorgaande methodes met een simulatie. In de tabellen in de komende paragrafen staat op iedere regel de eigenschappen van een bepaalde productielijn. De variabelen in de tabel zijn als volgt gedefinieerd

v_m	snelheid machine m in duizendtallen per tijdseenheid
K_i	grootte buffer i in duizendtallen
Mi	schatting van de output door methode i in eenheden per tijdseenheid
S	output volgens de simulatie in eenheden per tijdseenheid

4.1 Machines met gelijke snelheid

We beginnen met het onderzoeken van productielijnen met identieke machines en constante of dalende/stijgende buffergroottes. Alle machines hebben $\lambda = \frac{1}{10}$, $\mu = 1$ en $v = 10$. De dik gedrukte waarde is de benadering die het dichtst bij de uitkomst van de simulatie ligt.

v_1	v_2	v_3	v_4	K_1	K_2	K_3	$M1$	$M2$	$M3$	S
10	10	10	10	1	1	1	7250	7250	**7254**	7283
10	10	10	10	25	25	25	8326	8326	**8329**	8423
10	10	10	10	100	100	100	**8827**	**8827**	**8827**	8869
10	10	10	10	5	25	50	**8249**	8099	8208	8283
10	10	10	10	50	25	5	8099	**8249**	8208	8280

Bij een constante buffergrootte blijkt methode 3 het beste te werken. Bij stijgende buffergrootte methode 1. Bij dalende buffergrootte methode 2.

Vervolgens bekijken we een lijn waarbij de downrate net zo groot is als de uprate. Voor iedere machine geldt $\lambda = 1$ en $\mu = 1$.

v_1	v_2	v_3	v_4	K_1	K_2	K_3	$M1$	$M2$	$M3$	S
10	10	10	10	1	1	1	**2183**	**2183**	2130	2261
10	10	10	10	25	25	25	**3820**	**3820**	3818	3984
10	10	10	10	100	100	100	**4576**	**4576**	4574	4647

Methode 3 lijkt het nu slechter te doen, ook bij de constante kleine buffers. De benaderingen zijn over het algemeen ook slechter geworden vergeleken met de voorgaande tabel.

4.2 Machines met verschillende snelheden

Vervolgens bekijken we machines met dalende/stijgende snelheden en constante/dalende/stijgende buffergroottes.

v_1	v_2	v_3	v_4	K_1	K_2	K_3	$M1$	$M2$	$M3$	S
1	5	10	25	1	1	1	837	**839**	**839**	872
1	5	10	25	25	25	25	**909**	**909**	**909**	909
1	5	10	25	100	100	100	**909**	**909**	**909**	909
1	5	10	25	5	25	50	**909**	**909**	**909**	909
25	10	5	1	50	25	5	**909**	**909**	**909**	909
25	10	5	1	1	1	1	**839**	837	**839**	872
25	10	5	1	25	25	25	**909**	**909**	**909**	909
25	10	5	1	100	100	100	**909**	**909**	**909**	909
25	10	5	1	5	25	50	**909**	**909**	**909**	909
25	10	5	1	50	25	5	**909**	**909**	**909**	909

Bij een productielijn met een machine met een zeer lage snelheid, rekent iedere methode de goede waarde uit. Om goed te kunnen zien welke methode het beste werkt bij wisselende

snelheden/buffergroottes, is bij de volgende tabel de machine met $v = 1$ vervangen door een machine met $v = 50$.

v_1	v_2	v_3	v_4	K_1	K_2	K_3	$M1$	$M2$	$M3$	S
25	10	5	50	1	1	1	3767	3746	**3773**	3794
25	10	5	50	25	25	25	**4540**	4538	**4540**	4540
25	10	5	50	100	100	100	**4546**	**4546**	**4546**	4546
25	10	5	50	5	25	50	**4540**	**4540**	**4540**	4540
25	10	5	50	50	25	5	**4410**	**4410**	**4410**	4410

Alle methodes lijken goed te werken. Bij een kleine constante buffergrootte is methode 3 iets beter.

4.3 Lange productielijnen

Als laatste bekijken we de invloed van de lengte van de lijn op de drie methodes. De lijn bestaat uit N identieke machines met $\lambda = \frac{1}{10}$, $\mu = 1$ en $v = 10$. De *fout* die in de tabel staat weergegeven, is het percentage dat de bijbehorende methode naast de werkelijke waarde zit.

N	$M1/M2$	$fout$	$M3$	$fout$	S
4	8326	1.2	8329	1.1	8423
6	8026	2.8	8026	2.8	8252
8	7791	4.6	7804	4.4	8148
10	7595	6.4	7614	6.1	8077
12	7427	8.1	7451	7.7	8025
14	7280	9.7	7308	9.3	7984

Bij ieder tweetal machines dat bij de lijn komt, lijkt de fout bij methode 1 en 2 met 1.7 procent toe te nemen. Bij methode 3 met 1.6. Naarmate de lengte van de lijn toeneemt, wordt methode 3 beter dan methode 1 en 2.

4.4 Productielijnen zonder buffer

In deze paragraaf gaan we controleren of de methode ook werkt als $K = 0$. Naast dat we de resultaten kunnen controleren met behulp van de simulatie, is het in dit geval ook mogelijk om de output te controleren met een handmatige berekening. Hoe deze berekening in zijn werk gaat, laten we zien aan de hand van een voorbeeld.

We beschouwen de volgende situatie:
Een productielijn met drie machines, ieder met $\lambda = \frac{1}{10}$, $\mu = 1$ en $v = 10$.

De rate waarmee het proces van situatie i naar situatie j gaat moet gelijk zijn aan de rate waarmee het proces van situatie j naar situatie i gaat. Beschouw $\Pi_{(\ldots)}$ als de fractie van de tijd dat het systeem in de situatie beschreven in het subscript bevindt. We kunnen dan vervolgens de volgende balansvergelijkingen opstellen:

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{10} & -1 & 0 & 0 \\ \frac{1}{10} & 0 & -1 & 0 \\ \frac{1}{10} & 0 & 0 & -1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \Pi_{(+,+,+)} \\ \Pi_{(+,+,-)} \\ \Pi_{(+,-,+)} \\ \Pi_{(-,+,+)} \end{pmatrix}$$

Als we dit stelsel oplossen vinden we dat $\Pi_{(+,+,+)} = \frac{10}{13}$. Dit is de enige waarde die we nodig hebben aangezien in de andere situaties het systeem geen output geeft. De output van het systeem wordt bepaald door de machine met de laagste snelheid. De output is dus gelijk aan $\Pi_(+,+,+) \cdot min_i(v_i)$, ofwel in deze situatie: $\frac{10}{13} \cdot 10 \approx 7.69$.

In het algemeen, bij een productielijn van n machines, zien de balansvergelijkingen er als

volgt uit:

$$\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \lambda_1 & -\mu_1 & 0 & \dots & 0 \\ \lambda_2 & 0 & -\mu_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \lambda_n & 0 & 0 & \dots & -\mu_n \\ 1 & 1 & \dots & 1 & 1 \end{pmatrix} \begin{pmatrix} \Pi_{(+,+,\dots,+,+)} \\ \Pi_{(+,+,\dots,+,-)} \\ \vdots \\ \Pi_{(+,-,\dots,+,+)} \\ \Pi_{(-,+,\cdots,+,+)} \end{pmatrix}$$

In de oplossing van dit stelsel geldt $\Pi_{(+,+,\dots,+,+)} = \frac{\prod_1^n \mu_i}{Determinant(M)}$ waarbij M de matrix in de balansvergelijking is. In de volgende tabel staat weergegeven wat handmatig uitgerekend wordt (H), wat de methodes (Mi) teruggeven en wat de simulatie (S) teruggeeft.

v_1	v_2	v_3	v_4	λ	μ	H	$M1$	$M2$	$M3$	S
10	10	10	10	1/10	1	7143	7143	7143	7143	7143
25	10	5	50	1/10	1	3571	3571	3571	3571	3571
25	10	5	50	1	1	1000	1000	1000	1000	1000
10	10	10	10	1	1	2000	2000	2000	2000	2000
$v_1 = 10$	$v_2 = 10$	...	$v_{10} = 10$	1/10	1	5000	5000	5000	5000	5000

Het is duidelijk dat geen van de methodes problemen heeft met het feit dat $K = 0$ en in ieder geval de exacte waarde wordt uitgerekend.

5 Verbeteringen

5.1 Analyse

Men zou verwachten dat de waarden die de verschillende methodes teruggeven hoger zouden zijn dan die van de simulatie. Bij het aggregeren van de machines gaan we er namelijk van uit dat deze machines geen invloed ondervinden van de daarop volgende machines. Dit is in de realiteit echter niet het geval.

Stel bijvoorbeeld dat we een productielijn hebben van drie machines. Als we methode 1 toepassen, aggregeren we de eerste twee machines en negeren we de rest van de lijn. Is de buffer achter machine 2 echter vol en machine 3 *down*, dan kan machine 2 niet produceren. Dit wordt niet meegenomen in de analyse. De machines zijn in de berekening dus beter dan dat ze in werkelijkheid zijn. We zouden daarom dus schattingen verwachten die hoger zijn dan de simulatie.

Dit is echter niet het geval. Zelfs niet als we bovenbeschreven situatie proberen te construeren met goede en slechte machines, zie de hierop volgende tabel. De goede machine, met $v = 100$, heeft $\lambda = \frac{1}{100}$ en $\mu = 1$. Een slechte machine, met $v = 1$, heeft $\lambda = 1$ en $\mu = \frac{1}{10}$.

v_1	v_2	v_3	v_4	K_1	K_2	K_3	$M1$	$M2$	$M3$	S
100	1	1	1	1	1	1	45	45	45	47
100	100	1	1	1000	1	1	**57**	**57**	**57**	57
100	100	100	1	1000	1000	1	**91**	**91**	**91**	91

De methodes geven weer de juiste of lagere schattingen aan.

Het feit dat de methodes lagere schattingen maken dan de simulatie kan misschien als volgt verklaard worden. Naast het feit dat machine 2 niet kan produceren als buffer 2 vol zit en machine 3 *down* is, geldt ook $\lambda_2 = 0$. We zitten immers in de productieafhankelijke situatie. In realiteit gaat de machine dus minder vaak *down*.

We zouden dus de methodes misschien kunnen verbeteren als we betere schattingen maken van de parameters. We zouden dat op de volgende manier kunnen doen.

5.2 Correctie van de parameters

Als we naar het systeem M_1/M_2 kijken houden we er normaal geen rekening mee dat M_2 geblokkeerd kan zijn omdat buffer 2 vol is en M_3 down. We zeggen in het vervolg dat M_2 dan *blocked* is. Bij M_2/M_3 gaan we er ook onterecht van uit dat het systeem altijd input heeft. Als dit niet het geval is zeggen we dat M_2 *starved* is. We gaan proberen om de up, -en downrates van M_2 hierop te corrigeren. In het komende gedeelte noteren we kansen uit het eerste subsysteem, M_1/M_2, op de bekende manier, $F_i(K)$,$P_i(0)$ en $P_i(K)$. Kansen uit het tweede subsysteem, M_2/M_3, geven we aan met een accent, dus $F_i'(K)$,$P_i'(0)$ en $P_i'(K)$. In subsysteem M_1/M_2 beschouwen we M_2 als een *departure*-machine. In subsysteem M_2/M_3 beschouwen we M_2 als een *arrival*-machine. De parameters van deze machines worden gekenmerkt doormiddel van een subscript, respectievelijk met een d en een a

5.2.1 Corrigeren op *starvation*

Downrate
Om te beginnen corrigeren we de *downrate* van M_2 in M_2/M_3 op *starvation*. De correctie voor het feit dat M_2 *down* gaat omdat hij *starved* is, noteren we met λ_s. De *downrate* voor M_2 in M_2/M_3 wordt dan $\lambda_a = \lambda_2 + \lambda_s$. λ_s berekenen we op de volgende manier. Eerst rekenen we op de bekende manier de systemen M_1/M_2 en M_2/M_3 door. Vervolgens berekenen we de

rate weermee in toestand 2 de buffer in systeem $M1/M2$ leeg raakt doormiddel van $f_2(0)v_2$. Er geldt dan

$$\lambda_s = \frac{f_2(0)v_2 + P(\text{M}_1/\text{M}_2 \text{ is in toestand 4 en x} = 0)\lambda_1}{P(\text{M}_2 \text{ is up})}.$$

Ofwel

$$\lambda_s = \frac{f_2(0)v_2 + P_4(0)\lambda_1}{F_4(K) + F_2(K) - P_2(0)}.$$

Uprate
De rate waarmee M_2 in M_2/M_3 *up* gaat, μ_a, kunnen we vervolgens berekenen doormiddel van de vergelijking

$$\frac{1}{\mu_a} = \frac{\lambda_2}{\lambda_2 + \lambda_s}\frac{1}{\mu_2} + \frac{\lambda_s}{\lambda_2 + \lambda_s}\frac{1}{\mu_1}$$

Snelheid
Logischerwijs is de *troughput* van M_1/M_2 gelijk aan die van M_2/M_3. We kunnen de formule van de *troughput* voor M_2 in M_2/M_3 dus gelijk stellen aan de formule voor de *troughput* van M_2 in M_1/M_2. We krijgen dus

$$T_d = P_4'(K)v_3 + (F_4'(K) - P_4'(K) + F_1'(K) - P_1'(K))v_a.$$

Hieruit kunnen we v_a oplossen.

5.2.2 Corrigeren op *blocking*

Nu we de parameters van M_2 gecorrigeerd hebben op *starvation*, rekenen we met deze parameters eerst M_2/M_3 door. Met de $F_i'(K)$'s, $P_i'(0)$'s en $P_i'(K)$'s die we vinden gaan we M_2 in M_1/M_2 corrigeren op het feit dat M_2 *blocked* kan zijn.

Downrate
Deze correctie voor de *downrate* noteren we met λ_b. We berekenen deze op de volgende manier: De rate weermee in toestand 1 de buffer in systeem $M2/M3$ vol raakt berekenen we met $f_1'(K)v_2$. Er geldt dan

$$\lambda_b = \frac{f_1'(K)v_2 + P'(\text{M}_2/\text{M}_3 \text{ is in toestand 4 enx} = \text{K})\lambda_3}{P'(\text{M}_2 \text{ is up})}.$$

Ofwel

$$\lambda_b = \frac{f_1'(K)v_2 + P_4'(K)\lambda_3}{F_4'(K) + F_1'(K) - P_1'(K)}.$$

De nieuwe *downrate* voor M_2 in M_1/M_2 wordt dan

$$\lambda_d = \lambda_2 + \lambda_b.$$

Uprate
De nieuwe *uprate* voor M_2 in M_1/M_2 kunnen we berekenen doormiddel van

$$\frac{1}{\mu_d} = \frac{\lambda_2}{\lambda_2 + \lambda_b}\frac{1}{\mu_2} + \frac{\lambda_b}{\lambda_b + \lambda_2}\frac{1}{\mu_3}.$$

Snelheid
Om de nieuwe snelheid v_d te berekenen merken we eerst op dat $v_d = v_3$ als de buffer in het systeem M_2/M_3 vol is en M_2 en M_3 beiden *up* zijn. In alle andere gevallen is $v_d = v_2$, mits M_2 *up* is. We berekenen daarom eerst

$$y = \frac{\text{fractie van de tijd dat M}_2 \text{ en M}_3 \textit{ up} \text{ zijn en buffer 2 vol}}{\text{fractie van de tijd dat M}_2 \textit{ up} \text{ is}}.$$

Ofwel

$$y = \frac{P_4'(K)}{F_1'(K) - P_1'(K) + F_4'(K)}.$$

v_d vinden we vervolgens doormiddel van

$$v_d = (1 - y)(\text{originele snelheid machine 2}) + y(\text{originele snelheid machine 3}).$$

5.3 Iteratie-methode

In bovenstaande theorie verbeteren we de parameters van M_1/M_2 aan de hand van berekeningen in M_2/M_3 en andersom. Als we dezelfde theorie dus een tweede keer zouden toepassen, dus met verbeterde parameters van beide subsystemen, dan krijgen we nog betere schattingen van de betrokken parameters. We zouden in principe deze theorie kunnen blijven toepassen totdat de parameters convergeren. Deze methode noemen we de iteratie-methode. Deze methode leggen we uit aan de hand van een productielijn van vier machines en drie buffers. Langere productielijnen kunnen op analoge wijze geïtereerd worden.

Stap 1.
Eerst berekenen we de *arrival*-parameters en *departure*-parameters uit van M_2, dus respectievelijk de parameters gecorrigeerd op *starvation* en *blockation*.

Stap 2.
Vervolgens berekenen we de *arrival*-parameters en *departure*-parameters uit van M_3. Bij deze berekening geven we M_2 echter de *arrival*-parameters mee die we in stap 1 hebben berekend.

Stap 3.
Hierna berekenen we opnieuw de *arrival*-parameters en *departure*-parameters uit van M_2. Bij deze berekening geven we M_3 echter de *departure*-parameters mee die we in stap 2 hebben berekend.

Stap 4.
In deze voorlaatste stap berekenen we nog een keer de *arrival*-parameters en *departure*-parameters uit van M_3. Bij deze berekening geven we M_2 echter de *arrival*-parameters mee die we in stap 3 hebben berekend.

Stap 5.
Nu kunnen we de *output* van het systeem M_3/M_4 berekenen waarbij we M3 de *arrival*-parameters meegeven die in Stap 4 zijn berekend.

Deze stappen blijken genoeg om de parameters te laten convergeren.

5.4 Resultaten

De iteratie-methode hebben we toegepast op dezelfde productielijnen als in hoofdstuk 4. Hieronder zijn dezelfde tabellen nog eens terug te vinden met in de voorlaatste kolom de resultaten van de nieuwe methode. Het blijkt dat de nieuwe methode beter is in alle gevallen, behalve in het geval van wisselende snelheden en zeer grote of kleine buffers.

5.4.1 Wisselende buffergrootte

Alle machines hebben $\lambda = \frac{1}{10}$, $\mu = 1$ en $v = 10$ tenzij anders vermeld.

v_1	v_2	v_3	v_4	K_1	K_2	K_3	$M1$	$M2$	$M3$	$M4$	S
10	10	10	10	1	1	1	7250	7250	**7254**	7285	7283
10	10	10	10	25	25	25	8326	8326	**8329**	8447	8423
10	10	10	10	50	50	50	8630	8630	**8631**	8707	8688
10	10	10	10	100	100	100	**8827**	**8827**	**8827**	–	8869
10	10	10	10	1000	1000	1000	9061	9061	9061	–	9067
10	10	10	10	5	25	50	**8249**	8099	8208	8292	8283
10	10	10	10	5	25	100	8276	8152	8249	–	8308
10	10	10	10	50	25	5	8099	**8249**	8208	8292	8280
10	10	10	10	100	25	5	8079	8237	8249	–	8310

In de tabel is te zien dat als er in de lijn een buffer van grootte 100 te vinden is, methode 4 geen output geeft. Dit is vanwege de numerieke instabiliteit van onze laatste methode. Dit verschijnsel doet zich voor als er een buffergrootte groter dan 50 in de lijn te vinden is.

Methode 4 is echter een methode die rekening houdt met blokkering of uitputting van een buffer. Bij buffers van deze grootte hebben we daar geen last van en de andere methodes kunnen daarom gebruikt worden om een goede schatting van de output te maken. We zien in de tabel ook dat als de grote buffer achteraan staat, dat we dan het beste methode 1 kunnen nemen. Staat de grote buffer vooraan, dan kunnen we beter methode 2 kiezen.

Vervolgens bekijken we nog eens een lijn waarbij de *downrate* net zo groot is als de *uprate*. Voor iedere machine geldt $\lambda = 1$ en $\mu = 1$.

v_1	v_2	v_3	v_4	K_1	K_2	K_3	$M1$	$M2$	$M3$	$M4$	S
10	10	10	10	1	1	1	**2183**	**2183**	2183	–	2261
10	10	10	10	10	10	10	3229	3229	3158	–	3357
10	10	10	10	25	25	25	**3820**	**3820**	3818	4003	3984
10	10	10	10	50	50	50	4208	4260	4258	4390	4372
10	10	10	10	100	100	100	**4576**	**4576**	4574	–	4647

Weer zien we dat methode 4 instabiel wordt bij grote buffers. Bij deze machines wordt de methode echter ook instabiel bij kleine buffers. In beide gevallen kunnen de overige methodes als een redelijke schatting gebruikt worden.

5.4.2 Wisselende buffergrootte/Snelheden

v_1	v_2	v_3	v_4	K_1	K_2	K_3	$M1$	$M2$	$M3$	$M4$	S
25	10	5	50	1	1	1	3767	3746	**3773**	3792	3794
25	10	5	50	25	25	25	**4540**	4538	**4540**	4541	4540
25	10	5	50	100	100	100	**4546**	**4546**	**4546**	4546	4546
25	10	5	50	5	25	50	**4540**	**4540**	**4540**	4541	4540
25	10	5	50	50	25	5	**4410**	**4410**	**4410**	4410	4410

Alle methodes zijn in dit geval even goed, behalve bij de kleine buffergroottes waar methode 4 het beste resultaat geeft. Het is opmerkelijk dat nu zelfs bij de zeer grote buffers methode 4 goede resultaten weet te geven.

5.4.3 Lange productielijnen

Als laatste bekijken we de invloed van de lengte van de lijn op de nieuwe methode. De lijn bestaat uit identieke machines met $\lambda = \frac{1}{10}$, $\mu = 1$ en $v = 10$.

N	$M1/M2$	$fout$	$M3$	$fout$	$M4$	$fout$	S
4	8326	1.2	8329	1.1	8447	0.2	8423
6	8026	2.8	8026	2.8	8290	0.5	8252
8	7791	4.6	7804	4.4	8183	0.4	8148
10	7595	6.4	7614	6.1	8099	0.3	8077
12	7427	8.1	7451	7.7	8029	0.05	8025
14	7280	9.7	7308	9.3	7970	0.6	7984

Hier is duidelijk te zien dat de vierde methode veel betere schattingen maakt dan de eerder besproken methodes.

6 Conclusie

We hebben verschillende manieren bestudeerd om bepaalde typen productielijnen te aggregeren. De conclusie die we hier uit kunnen trekken is dat het wel degelijk uitmaakt welke methode er gebruikt wordt bij een bepaald type productielijn. Als we methode 4 buiten beschouwing laten kunnen we opmerken dat:

- Methode 1 werkt het beste bij een productielijn waarbij alle machines dezelfde snelheid hebben en de buffercapaciteit van op één volgende buffers niet daalt.
- Methode 2 werkt het beste bij een productielijn waarbij alle machines dezelfde snelheid hebben en de buffercapaciteit van op één volgende buffers niet stijgt.
- Methode 3 werkt het beste bij een productielijn waarbij de buffercapaciteit niet veranderd en lange productielijnen.
- Methode 1, 2 en 3 werken goed bij een productielijn met enkel grote buffers.
- Alle methodes werken goed bij een productielijn die minstens één trage machine bevat.

We hebben tijdens het onderzoek ook geconstateerd dat de iteratie-methode, methode 4, de schatting in veel gevallen beter maakt. Voornamelijk in het geval van lange productielijnen blijkt methode 4 het stukken beter te doen. Bij een productielijn waarbij alle machines dezelfde snelheid hebben en die zeer grote buffers bevat, laat de methode het echter afweten. In de gevallen waarbij methode 4 niet goed werkt kunnen de overige methodes als goede schatting gebruikt worden.

7 Bronvermelding

I.J.B.F. Adan, J. van der Wal, *Difference and differential equations in Stochastic Operations Research*, Department of Mathematics and Computing Science, University of Technology, Eindhoven, 1998, pp 123 – 127.

M.B.M. De Koster, *Estimation of line efficiency by aggregation*, Int. J. Prod. Res., 1987, vol. 25, no. 4, pp 615 – 626