



FACULTEIT ECONOMIE EN BEDRIJFSKUNDE

DE IMPACT VAN PRODUCTIEONDERBREKING OP KITTING, EEN ANALYTISCHE STUDIE

UNIVERSITEIT GENT

FACULTEIT ECONOMIE EN BEDRIJFSKUNDE

Academiejaar: 2009-2010

Masterproef voorgedragen tot het bekomen van de graad van
Master in de Toegepaste Economische Wetenschappen: Handelsingenieur

Eline De Cuypere
onder leiding van:
Prof. dr. ir. H. Bruneel
thesisbegeleider:
Prof. dr. ing. D. Fiems

Ondergetekende verklaart dat de inhoud van deze masterproef mag geraadpleegd en/of gereproduceerd worden, mits bronvermelding.

De Cuypere Eline

Dankwoord

Ik mag deze thesis dan zelf geschreven hebben, zonder de steun en raadgevingen van verschillende mensen zou het mij niet gelukt zijn. Al deze personen wil ik graag met name bedanken. In de eerste plaats mijn promotor, prof. dr. ir. H. Bruneel voor het beschikbaar stellen van dit onderwerp. Daarnaast wil ik graag prof. dr. ing. D. Fiems en ir. Stijn De Vuyst danken voor de begeleiding van mijn thesis. Hun kennis en raadgevingen waren van goudwaarde. Mijn familie wil ik danken voor hun geloof in mij en de jarenlange steun. Verder wil ik een paar mensen in het bijzonder vermelden. Mr. Evert voor het nalezen van mijn thesis, mijn moeder voor de morele steun en mijn vader voor de interessante tips en advies.

Inhoudsopgave

Inleiding	1
1 Beschrijving van het kittingproces	3
1.1 Soorten leveringssystemen	3
1.2 Kittingproces	4
1.2.1 Kit	4
1.2.2 Kitting	5
1.2.3 Kitting in het Volvo Europa Trucks bedrijf	5
1.2.4 Belangrijkste voordelen en nadelen	6
1.3 Productieonderbreking	8
2 Markov-processen	11
2.1 Een wachttijmodel voor kitting in een stochastische omgeving	11
2.2 Kittingproces als Markov-model	11
2.2.1 Markoviaanse-aankomstprocessen	12
2.2.1.1 Poisson-proces	12
2.2.1.2 Onderbroken Poisson-proces	12
2.3 Kittingproces als een Markov-keten met continue-tijdparameter	13
2.3.1 Modelspecificatie	14
2.3.2 Generatormatrix	14
2.3.3 Stationaire waarschijnlijkheidsvector	16
3 Basis-kittingmodel	19
3.1 Beschrijving van het basismodel	19
3.1.1 Literatuur	19
3.1.2 Verschil met ons model	20

3.1.3	Beschrijving van een wachtlijnsysteem	21
3.1.4	Basisveronderstellingen	22
3.2	Constructie en analyse van het basismodel	22
3.2.1	Bepaling van het basismodel	22
3.2.2	Generatormatrix	24
3.2.3	Stationaire waarschijnlijkheidsvector	25
3.2.4	Prestatiematen	27
3.3	Methodologie in MATLAB	28
3.3.1	Sparse-methode	28
3.3.2	GMRES-methode	29
3.3.3	Optimalisatie procedures	31
3.4	Implementatie in MATLAB	32
3.4.1	Generatormatrix	32
3.4.2	Prestatiematen	34
3.5	Numerieke voorbeelden	38
4	Uitgebreide modellen	43
4.1	Beschrijving van de uitgebreide modellen	43
4.1.1	Eerste uitgebreid model voor kitting: het 1IPP-model	43
4.1.2	Tweede uitgebreid model voor kitting: het 2IPP-model	45
4.2	Bepaling van de ingangparameters	46
4.3	Numerieke voorbeelden bij variërende capaciteit	49
4.3.1	1IPP-model als $\sigma = 0,8$ en $\kappa = 10$	49
4.3.2	1IPP-model als $\sigma = 0,4$ en $\kappa = 10$	53
4.3.3	2IPP-model als $\sigma_1 = \sigma_2 = 0,8$ en $\kappa_1 = \kappa_2 = 10$	55
4.3.4	2IPP-model als $\sigma_1 = \sigma_2 = 0,4$ en $\kappa_1 = \kappa_2 = 10$	59
4.3.5	Modellen samen als $\sigma = 0,4$ en $\kappa = 10$	61
4.4	Numerieke voorbeelden bij variërende werklast	65
4.4.1	Basismodel	65
4.4.2	1IPP-model	67
5	Conclusies en verder onderzoek	71
5.1	Conclusies	71
5.2	Verder onderzoek	73

Bibliografie	i
Index	iii
A Bijlagen	v
A.1 Lijst van de gebruikte symbolen	v
A.2 Verstreken tijd	vii
A.3 Ingangparameters	ix
A.4 De generatormatrix in MATLAB voor het 2IPP-model	xi

Lijst van figuren

3.1	Kittingmodel volgens Som, Wilhelm en Disney	20
3.2	Schema voor het basismodel	23
3.3	Fragment van het toestandsdiagram voor het basismodel	23
3.4	Volledig toestandsdiagram voor het basismodel waarbij $C_1 = 3$ en $C_2 = 2$	24
3.5	Toestandsdiagram voor toestand (1,1)	24
3.6	Generatormatrix \mathbf{Q} voor het basismodel waarbij $C_1 = 3$ en $C_2 = 2$	25
3.7	Distributie van de bufferbezetting van buffer 1 ($C_1 = 20$)	27
3.8	Vergelijking in geheugenverbruik tussen de dense en sparse-matrix	29
3.9	Basismodel: Gemiddelde bezetting van buffer 1 bij variërende C_1	38
3.10	Basismodel: Variantie in de bezetting van buffer1 bij variërende C_1	39
3.11	Basismodel: Kans dat één van beide buffers leeg is bij variërende C_1	40
3.12	Basismodel: Kans dat buffer 1 vol is bij variërende C_1	41
4.1	Schema voor het 1IPP-model	44
4.2	Transitietoestanden voor het 1IPP-model	44
4.3	Fragment van het toestandsdiagram voor het 1IPP-model	44
4.4	Toestandsdiagram voor toestand (1,1,1)	45
4.5	Fragment van het toestandsdiagram voor het 2IPP-model	45
4.6	Schema voor het 2IPP-model	47
4.7	Transitietoestanden voor het 2IPP-model	47
4.8	1IPP-model: Kans dat één van beide buffers leeg is bij variërende C_1 en $\sigma = 0,8$	50
4.9	1IPP-model: Kans dat buffer 1 vol is bij variërende C_1 en $\sigma = 0,8$	51
4.10	1IPP-model: Kans dat buffer 2 vol is bij variërende C_2 en $\sigma = 0,8$	51
4.11	1IPP-model: Verlieskans voor beide buffers bij variërende C_1 en $\sigma = 0,8$	52

4.12	1IPP-model: Kans dat buffer 1 vol is bij variërende C_2 en $\sigma = 0,4$	53
4.13	1IPP-model: Kans dat buffer 2 vol is bij variërende C_1 (links) en C_2 (rechts) en $\sigma = 0,4$	54
4.14	1IPP-model: Verlieskans voor beide buffers bij variërende C_1 en C_2 en $\sigma = 0,4$	54
4.15	2IPP-model: Gemiddelde bezetting in buffer1 bij variërende C_1 en $\sigma_1 = \sigma_2 = 0,8$	56
4.16	2IPP-model: Kans dat één van beide buffers leeg is bij variërende C_1 en $\sigma_1 = \sigma_2 = 0,8$	56
4.17	2IPP-model: Kans dat buffer 1 vol is bij variërende C_1 en $\sigma_1 = \sigma_2 = 0,8$	57
4.18	2IPP-model: Verlieskans voor beide buffers bij variërende C_1 en $\sigma_1 = \sigma_2 = 0,8$	57
4.19	2IPP-model: Kans dat buffer 1 vol is bij variërende C_1 en $\sigma_1 = \sigma_2 = 0,4$	60
4.20	2IPP-model: Verlieskans voor beide buffers bij variërende C_1 en $\sigma_1 = \sigma_2 = 0,4$	60
4.21	Modellen samen: Gemiddelde bufferbezetting in buffer 1 bij variërende $C_1 = C_2$ en $\sigma = 0,4$	61
4.22	Modellen samen: Gemiddelde bezetting in buffer 2 bij variërende $C_1 = C_2$ en $\sigma = 0,4$	62
4.23	Modellen samen: Kans dat buffer 1 vol is bij variërende $C_1 = C_2$ en $\sigma = 0,4$	62
4.24	Modellen samen: Kans dat buffer 2 vol is bij variërende $C_1 = C_2$ en $\sigma = 0,4$	63
4.25	Modellen samen: Verlieskans voor beide buffers bij variërende $C_1 = C_2$ en $\sigma = 0,4$	63
4.26	Basismodel: Gemiddelde aantal in buffer 1 bij variërende $\lambda_1 = \lambda_2$	65
4.27	Basismodel: Gemiddelde aantal in buffer 2 bij variërende $\lambda_1 = \lambda_2$	66
4.28	Basismodel: Verlieskans voor beide buffers bij variërende $\lambda_1 = \lambda_2$	67
4.29	1IPP-model: Gemiddelde aantal in buffer 1 bij variërende $\lambda_1 = \lambda_2$ en $\sigma = 0,4$	67
4.30	1IPP-model: Gemiddelde aantal in buffer 2 bij variërende $\lambda_1 = \lambda_2$ en $\sigma = 0,4$	68
4.31	1IPP-model: Verlieskans voor beide buffers bij variërende $\lambda_1 = \lambda_2$ en $\sigma = 0,4$	69

Lijst van tabellen

3.1	Ingangparameters voor het basismodel	38
4.1	Ingangparameters voor het 1IPP-model als $\sigma = 0,8$	49
4.2	Ingangparameters voor het 1PP-model als $\sigma = 0,4$	53
4.3	Ingangparameters voor het 2IPP-model als $\sigma_1 = \sigma_2 = 0,8$ en $\kappa_1 = \kappa_2 = 10$	55
4.4	Ingangparameters voor het 2IPP-model als $\sigma_1 = \sigma_2 = 0,4$ en $\kappa_1 = \kappa_2 = 10$	59
4.5	Ingangparameters voor het 1PP-model als $\sigma = 0,4$	65
A.1	Verstreken tijd bij een eenmalige berekening van prestatiematen	vii
A.2	Verstreken tijd wanneer de capaciteit varieert	viii
A.3	Verstreken tijd wanneer de aankomstintensiteit varieert	viii
A.4	Ingangparameters als $\sigma = 0,8$ en $\kappa = 10$	ix
A.5	Ingangparameters als $\sigma = 0,4$ en $\kappa = 10$	x

Inleiding

De huidige productiesystemen zijn vaak samengesteld uit meerdere eenheden. De halffabricaten geproduceerd door de ene eenheid vormen het basismateriaal voor een andere eenheid of voor een assemblagelijijn. Daarenboven eisen de klanten een grotere variatie en een op maat gemaakt aanbod. Klanten willen natuurlijk hun bestellingen op tijd geleverd krijgen en tegen een lage prijs. Al deze tendensen leiden tot meer stukken op de werkvloer en vragen om een doeltreffende transportmethode van de eenheden naar hun assemblagelijijn. Uiteindelijk dringt het kittingproces zich meer en meer op.

Het doel van dit eindwerk is de impact van productieonderbrekingen op dit proces te onderzoeken. Daarvoor modelleren we het kittingproces als een wachtrij met buffers: aankomende onderdelen wachten in hun respectievelijke buffers om achteraf in een buffer voor kits gegroepeerd te worden. We stellen modellen op waarbij er geen en wel rekening wordt gehouden met tijdelijke inefficiënties in de productie van onderdelen, welke leiden tot leveringsonderbrekingen, om zo hun impact op het proces te bestuderen.

In de modellen beperken we ons tot de analyse van kits welke samengesteld zijn uit twee componenten. De onderdelenbuffers worden onderdeel per onderdeel aangevuld en kits vertrekken afzonderlijk, de ene na de andere. De bufferruimte, voorzien om de onderdelen tijdelijk op te slaan, heeft een beperkte capaciteit.

Om de prestatieparameters te kunnen berekenen bij een grote buffercapaciteit maken we gebruik van de GMRES-methode (Generalized Minimum Residual). De reden is dat deze iteratieve methode veel sneller stelsels van lineaire vergelijkingen voor ons oplost dan andere directe methodes. De gevonden oplossing is echter niet noodzakelijk exact. Wel biedt de functie `gmres` in MATLAB de mogelijkheid aan om na te gaan hoe nauwkeurig de oplossing is.

Dit eindwerk is opgebouwd in vijf delen. In hoofdstuk 1 worden de begrippen kittingproces en productieonderbreking verklaard. In hoofdstuk 2 bespreken we de modellering van het kittingproces als een Markov-keten met continue-tijdparameter. In hoofdstuk 3 beschrijven we het basismodel en de gehanteerde methodologie in MATLAB. Hierbij illustreren we de prestaties van het model aan de hand van numerieke voorbeelden. In hoofdstuk 4 beschrijven we de twee uitgebreide modellen en bestuderen we de gevolgen van productieonderbrekingen op het kittingproces. In het laatste hoofdstuk vatten we de belangrijkste resultaten samen en bieden we mogelijkheden aan voor verder onderzoek.

Om de tekst duidelijker te maken worden definities en belangrijke begrippen in cursief gezet, wordt de vertaling van Engelse woorden tussen haakjes geplaatst, worden de verwijzingen naar een ander deel in dit document tussen aanhalingstekens geplaatst en worden tenslotte Engelse woorden welke in de tekst geen vertaling krijgen, als leenwoord beschouwd. De index bevat een lijst met de voornaamste begrippen en een lijst met de gebruikte symbolen kan gevonden worden in de bijlage A.1.

Hoofdstuk 1

Beschrijving van het kittingproces

In sectie 1.1 bespreken we de bestaande leveringssystemen van materiaal aan de assemblagelijijn waarvan het kittingproces een voorbeeld is. In plaats van containers met gelijkaardige onderdelen aan de assemblagelijijn te leveren, i.e. de meer traditionele lijnvoorraad (line stocking), worden de onderdelen op voorhand gegroepeerd en naar de kitcontaineropslagplaats gebracht [Bozer en McGinnis, 1992]. De definitie van het kittingproces, de bijhorende voor- en nadelen en een uitleg over de toepassing van dit proces in het Volvo Europa Trucks bedrijf worden uiteengezet in sectie 1.2. Als de grondstoffen niet op tijd vervaardigd zijn, dan daalt de efficiëntie van het proces. Wat we onder productieonderbreking verstaan, wordt uitgelegd in sectie 1.3.

1.1 Soorten leveringssystemen

Volgens Johansson [1991] zijn er drie mogelijkheden om componenten naar het assemblagestation te brengen: continu, batchgewijs of met behulp van kitting. De drie methodes worden volgens twee kenmerken gecategoriseerd. Het eerste kenmerk betreft de fabricage van de deelcomponenten: gaat het om een doorlopende of een selectieve productie? Het tweede kenmerk omvat het sorteren van de componenten: worden de onderdelen per type of per assemblagestuk gegroepeerd? Deze drie leveringssystemen kunnen tegelijkertijd in het systeem bestaan en elkaar aanvullen.

Een continue productie is een proces waarbij het produceren nooit stopt. Deze processen maken dikwijls slechts één soort eindproduct in grote volumes. Johansson [1991] beschrijft een *continu leveringssysteem* als een proces waarbij het vereiste aantal componenten op een continue manier aan de assemblagemachine geleverd wordt. Alle vervaardigde componenten worden naar de assemblagewerkplaats gebracht.

In een batch-productie worden gelijkaardige producten in serie gemaakt [Frederix, 2008]. Nadat een bepaald product voldoende geproduceerd is, wordt op dezelfde machine met een andere serie begonnen. Johansson [1991] beschrijft het *batch-leveringssysteem* als een systeem waarbij materiaal verzameld wordt voor een specifiek aantal assemblagestukken. Elke batch bestaat uit een aantal gelijkaardige componenten, bestemd voor een bepaalde hoeveelheid eindproducten. Het grote verschil met de continue productie is dat er minder componenten aanwezig zijn op de werkplaats aangezien de overbodige componenten in de voorraadplaats blijven.

Aangezien bedrijven hun aanbod meer en meer diversifiëren, wordt de behoefte aan gevarieerde componenten alsmear groter. Johansson en Johansson [2006] verklaren dat dit continue productie onmogelijk maakt wegens te

hoge kapitaalkosten. De verhoging van het aantal verschillende stukken betekent meer nood aan opslagplaats. Toch is een grotere voorraadruimte niet de oplossing. Dit vergt niet enkel bijhorende ruimte maar impliceert ook langere operatorstarttijd (operator walking time) en zoektijden naar componenten (search time) in de assemblagelijijn.

Componenten verzamelen in kits, maakt het zoeken naar componenten gemakkelijker en de nodige voorraadgrootte kleiner [Ramakrishnan en Krishnamurthy, 2008]. Hierdoor is een derde soort leveringssysteem ontstaan, het kitting-proces, dat nader besproken wordt in het volgende deel.

Later is een vierde soort systeem ontstaan dat het *kitting*- en *batchproces* combineert. In de kitlocatie worden niet één maar meerdere gelijkaardige kits tegelijk voorbereid. Wanneer er voldoende componenten uit de containers gehaald zijn, worden deze teruggebracht naar de voorraad. Het aantal gelijkaardige kits dat tegelijkertijd geassembleerd is, noemen Bozer en McGinnis [1992] een kit-batchgrootte (kit batch size). Volgens Carlsson en Hensvold [2008] is het kitten in batch enkel logisch als er geen of weinig gevarieerde deelcomponenten zijn.

1.2 Kittingproces

1.2.1 Kit

Een *kit* is een specifieke verzameling van componenten en/of assemblagestukken die samen één of meerdere assemblage operaties ondersteunen voor een bepaald eindproduct. Het vereiste aantal van elke componentensoort wordt bepaald door de kitstructuur [Bozer en McGinnis, 1992]. Dit gaat als volgt. Om een kit te kunnen samenstellen, moet men de nodige deelcomponenten eerst vervaardigen en in hun respectievelijke buffers plaatsen. Daarna worden de verschillende onderdelen in een andere container verzameld (kitcontainer of kit genoemd) voordat deze naar de assemblagewerkpost gebracht worden [Bryznér en Johansson, 1995, Medbo, 2003, Bozer en McGinnis, 1992, Ramachandran en Delen, 2005, Ramakrishnan en Krishnamurthy, 2008, Som, Wilhelm, en Disney, 1994]. De kit bevat echter niet altijd alle componenten die nodig zijn om een eindproduct te maken, bijvoorbeeld wegens een te hoge productcomplexiteit of een te groot product. Volgens Bryznér en Johansson [1995] is het ontwerpen van een geschikte kitcontainer de sleutelfactor voor een efficiënt kittingproces. De kitcontainers moeten functioneel ontworpen zijn voor zowel het selectieproces van de componenten als voor het assemblageproces. Zo weet de arbeider gemakkelijk welke componenten er nodig zijn en kan hij snel ontbrekende stukken detecteren.

Bozer en McGinnis [1992] duiden twee soorten kits in de industriewereld aan:

- de kit-to-customer waarbij het eindproduct als een kit aan de klant geleverd wordt en
- de kit-to-manufacturing, ook gekend als een kit-to-assembly operatie, waarbij basisstukken in een container samengebracht worden om ze na één of meerdere assemblage operaties als eindproduct aan de klant te leveren.

Hoewel onze studie zich in het bijzonder richt op kit-to-assembly, zullen de resultaten onder meer toegepast kunnen worden op kit-to-customer operaties.

Een bijkomende opdeling kan gemaakt worden aan de hand van de assemblagemethode [Limère, 2007]. Een *stationaire kit* (stationary kit) is een kit die naar een werkpost gebracht wordt en daar blijft tot assemblage. Het te assembleren product beweegt van de ene werkplaats naar de andere, los van de stationaire kit. Bij de tweede methode gaan de *bewegende kit* (travelling kit) en het te assembleren product samen langs de verschillende werkposten. Onze later uitgevoerde analyse richt zich echter op het opstellen van een kit en niet op de assemblagemethode. De keuze

voor een bepaalde assemblagemethode is belangrijk bij de implementatie van het kittingproces wegens zijn grote invloed op de totale productiekost.

Bryznér en Johansson [1995] hebben daarnaast twee mogelijke selectiezones bepaald (zone picking-methodes) om een kit samen te stellen. In de *progressieve zone*, (progressive zoning) wordt er gewerkt van de ene werkpost naar de andere. Eens de te vervaardigen componenten alle zones doorlopen hebben, is de kit klaar. In een *gesynchroniseerde zone* (synchronised zoning) worden de bewerkingen van de verschillende werkposten tegelijkertijd gedaan. Daarna worden alle bewerkte componenten van elke zone samengebracht in een kit. In ons onderzoek gaat het om een *gesynchroniseerde zone*; de twee bewerkte basiscomponenten worden verzameld tot een kit en in een kitcontainer geplaatst, zie verder.

In het volgende deel bespreken we wat het begrip kitting inhoudt. Hoewel deze methode meer en meer toegepast wordt in de industriële wereld, is de impact van kitting op de efficiëntie van het productieproces nog maar weinig onderzocht.

1.2.2 Kitting

Kitting is het in een fabriek klaarmaken van een bepaald hoeveelheid componenten om deze daarna samen te brengen in een specifieke container [Carlsson en Hensvold, 2008, Limère, 2007]. Een klant kan bijvoorbeeld online een pc bestellen en daarbij uit verschillende soorten geheugen, drives en randapparatuur kiezen. De leverancier ontwerpt op basis daarvan een kit die achteraf geassembleerd en getransporteerd wordt als één unit.

Studies hebben aangetoond dat kitting zeer voordelig is bij sterke gevarieerde parallelle assemblagelijnen [Johansson en Johansson, 1990, Carlsson en Hensvold, 2008]. Volgens Ding en Balakrishnan [1990] is een kittingproces het meest adequaat bij elektronische industrieën. Deze industrieën hebben namelijk meerdere kleine stukken en gevarieerde parallelle assemblagelijnen. Een JIT-systeem (Just-In-Time system) van grotere stukken, i.e. een systeem waarbij de grondstoffen enkel afgeleverd worden als ze in de productie nodig zijn, kan ook voordeel halen uit een kittingproces. Volgens Schwind [1992] zijn waardevolle componenten ook geschikt voor kitting. De componenten kunnen na een grondige kwaliteitscontrole individueel aan de container toegevoegd worden. Het toepassen van een kittingproces is bijgevolg sterk afhankelijk van de behoeftes van de specifieke industrieën [Carlsson en Hensvold, 2008].

Naast de layout (work organisation) en de kenmerken van de basiscomponenten bepaalt de locatie voor het kitting op de werkvloer de keuze voor het al dan niet uitvoeren van een kittingproces [Bryznér en Johansson, 1995]. Bij een *centrale ophaalvoorraad* (central picking store) kan men van schaalvoordelen genieten omdat meerdere kits in dezelfde zone vervaardigd worden. Anderzijds kunnen er communicatiegebreken ontstaan door de afgelegen locatie van de kittingruimte. Het voordeel van een *gedecentraliseerd systeem* ligt vooral in de communicatie. Er is wel meer ruimte nodig en het is arbeidsintensiever dan een centrale ophaalvoorraad.

Hedendaags wordt het kittingproces in meerdere autobedrijven geïmplementeerd. We bezochten het bedrijf Volvo Europa Trucks op 27 oktober 2009 dat deze transportmethode gebruikt om onderdelen van vrachtwagens samen naar het assemblagestation te brengen.

1.2.3 Kitting in het Volvo Europa Trucks bedrijf

Het Volvo Europa Trucks bedrijf is opgericht in 1928 in Oostakker en is eigendom van de Volvo Group. De firma is van Zweedse oorsprong en produceert 30% van het totale volume vrachtwagens van de Volvo Group. Vandaag de

dag is het de grootste producent van zware vrachtauto's in Europa en de tweede belangrijkste ter wereld. Jaarlijks worden er meer dan honderdduizend vrachtwagens geproduceerd en verkocht.

In 2008 heeft Volvo Europa Truck de cabinetrimafdeling uitgebreid. Dit nieuwe industriële gebouw, volledig opgericht volgens het concept van *lean manufacturing*, houdt zich bezig met het aankleden (trimmen) van cabines. In deze afdeling, wordt onder andere het kittingproces toegepast als bevoorradingsstechniek.

In het bedrijf Volvo is de kitting- en sequentie-ruimte (sequence space) een aparte voorbereidingszone waar het materiaal herverpakt en gesorteerd wordt volgens chassisnummers. De stukken van een bepaald soort materiaal zitten meestal per stuknummer in een pallet om er voor te zorgen dat de juiste stukken bij de juiste vrachtwagen terecht komen. Wegens plaatsgebrek nabij de productielijn worden deze paletten eerst naar de kitting- en sequentie-ruimte gebracht. Sequentie wordt gebruikt voor stukken die speciaal gemaakt zijn voor een bepaald chassis. Deze zijn meestal variantgevoelige delen zoals bijvoorbeeld zetels waarvan Volvo dertig verschillende soorten levert. De stukken worden volgens een bepaalde volgorde in de rekken geordend en vervolgens naar de productielijn gebracht. Kitting daarentegen wordt eerder gebruikt voor meer algemene onderdelen die voor verschillende vrachtwagens kunnen dienen. Dit zijn meestal ook kleine stukken. Hierbij maken ze kits met onderdelen die bij elkaar horen. Dankzij deze zone wordt het werk op de posten waar de vrachtwagens moeten geassembleerd worden verlicht. Ook de productielijn is hierdoor beter georganiseerd door de beperktere plaatsinname.

Wel moet opgemerkt worden dat het kittingproces momenteel nog veel tijd in beslag neemt en er veel werknemers nodig zijn om dit proces uit te voeren. Volvo is er zich echter wel van bewust dat kitting momenteel de knelpunt (bottleneck) van de cabinetrimfabriek is en focust zich dan ook op een continue verbetering van dit proces. Het zou efficiënter zijn om deze voorbereidingszone dichterbij de assemblage te plaatsen om hierdoor minder stukken in omloop te hebben, wat bijdraagt tot een lagere onderhanden werk met zo een gunstig effect op de kosten.

In de toekomst zou het bedrijf Volvo Europa Truck graag het Pick-to-light systeem toepassen om het bevoorradingswerk te verbeteren. Dit systeem zorgt ervoor dat overal waar er een stuk uitgehaald moet worden er een licht gaat branden. Het voordeel hiervan is dat het sneller gaat en er minder kans is om een verkeerd onderdeel weg te halen.

In het volgende deel vatten we de belangrijkste voor- en nadelen van het kittingproces samen.

1.2.4 Belangrijkste voordelen en nadelen

Sommigen zien het kittingproces als een toepassing van lean manufacturing. Hoewel kitting geen waarde toevoegende activiteit is, kan het de volledige materiële behandelingstijd sterk verminderen [Ramakrishnan en Krishnamurthy, 2008] omdat bepaalde activiteiten, zoals het selecteren en grijpen van componenten, efficiënter worden. De operatorstaptijd wordt zo drastisch gereduceerd en in sommige gevallen zelfs geëlimineerd [Johansson en Johansson, 1990]. De kits van componenten worden immers als geheel naar het assemblagestation gebracht [Bozer en McGinnis, 1992, Ding en Balakrishnan, 1990, Medbo, 2003]. Indien de onderdelen in de juiste positie in de container geplaatst zijn, vermindert ook de assemblagetijd [Ramakrishnan en Krishnamurthy, 2008].

Volgens anderen past kitting dan weer niet bij lean manufacturing omdat de voorbereiding van kits ook tijd, ruimte en inspanning vergt. Het begrip *lean* richt zich vooral op het wegwerken van allerlei soorten verspillingen zoals voorraadvorming, kwaliteits- en tijdverlies. Het gaat hier dus om het produceren van de exacte hoeveelheid op het moment dat ernaar gevraagd wordt en dit op een efficiënte manier. Dit principe staat vaak haaks op het kittingproces waarbij kits meestal op voorhand klaargemaakt worden om onder andere de operatorstaptijd te reduceren. Dit heeft overbodige voorraad als gevolg. Bovendien kunnen onvolledige of verkeerd samengestelde kits voor groot kwaliteits-

en tijdverlies zorgen. Een nieuwe tendens probeert deze twee begrippen te verzoenen: *lean kitting*. De principes om lean kitting te bereiken zijn het voortdurend correct samenstellen van kits, het vermijden van machinestilstand door onvolledige of verkeerd samengestelde kits en het verwijderen van verspillingen zoals tijdverlies in het kittingproces [Vujosevic, 1997].

Limère [2007] bespreekt positieve eigenschappen van kitting in het artikel “A decision model for kitting”. De voornaamste worden hieronder opgesomd aangevuld met andere voordelen besproken in andere artikels en opgemerkt door het New Breed Logistics bedrijf [2007]:

- Een grote reeks van gelijkaardige componenten containers wordt vervangen door een gelimiteerd aantal kit-containers. Dit leidt tot:
 - een verbeterde controle en zichtbaarheid op de componentenstroom naar de werkvloer, wat belangrijk is voor dure deelcomponenten of stukken die kunnen verrotten,
 - minder te verrichten werk (Work-In-Process, afgekort als WIP) aan de werkposten, wat tot een kortere doorlooptijd leidt [Ramakrishnan en Krishnamurthy, 2008],
 - een lagere tussenvoorraad en voorraadwaarde in de productielijn van de componenten [Bozer en McGinnis, 1992, Ding en Balakrishnan, 1990, Medbo, 2003],
 - een besparing in de vervaardigings- en assemblageruimte,
 - een verhoogde flexibiliteit die productverandering vergemakkelijkt,
 - een mogelijkheid om de werklust beter te spreiden over de assemblagelijn.
- Stukken worden beschikbaar gesteld in een kit, die vaak in de correcte volgorde voor assemblage geplaatst worden. Dit leidt tot:
 - een verhoogde productiviteit door de zoektijd te reduceren,
 - een potentiële kwaliteitsverhoging omdat defecte stukken gemakkelijker opgespoord kunnen worden [Bozer en McGinnis, 1992, Schwind, 1992, Ramakrishnan en Krishnamurthy, 2008].
- Kits kunnen op een gestandaardiseerde manier samengesteld worden, wat de organisatie ten goede komt. Dit leidt tot:
 - de mogelijkheid om een gerobotiseerde handeling te implementeren, met tijdsbesparing en een efficiëntere voorbereiding van kits als gevolg,
 - een vermindering van de arbeidskosten (de operatorstaptijd daalt waardoor minder arbeiders nodig zijn en de kennis voor nieuw personeel om te kunnen kitten en assembleren beperkt blijft). Het selecteren, sorteren en samenbrengen in een container vraagt geen gekwalificeerde arbeiders [New Breed bedrijf 2007],
 - een verhoogde nauwkeurigheid en tevredenheid van de arbeiders indien het kittingproces aan de samenstellers toegewezen wordt. Ten eerste, gebeurt het selecteren van componenten accurater wanneer de operator verantwoordelijk is voor de hele taak. Ten tweede, ontwikkelt de taakuitbreiding een betere werkinhoud en -kwaliteit [New Breed bedrijf 2007].

Hoewel de voordelen overtuigend zijn, blijken bedrijven nog steeds te twijfelen over het invoeren van een kittingproces in hun assemblagelijijn [Carlsson en Hensvold, 2008]. De onzekere uitkomsten en de dure initiële uitvoering zijn hiervan de voornaamste oorzaken [Ramakrishnan en Krishnamurthy, 2008]. Limère [2007] heeft het over de volgende minpunten:

- de tijd en inspanning om een kit voor te bereiden (bovendien is deze activiteit op zich geen waarde toevoegende activiteit),
- de behoefte aan een bufferruimte waarin de klaargemaakte kits opgeslagen worden,
- de bijkomende planning voor de toewijzing van de verschillende stukken aan de kits,
- het risico van zware daling van de productkwaliteit door ontbrekende, verkeerde of defecte stukken in de kits en [Carlsson en Hensvold, 2008]
- de kans tot een grote efficiëntiedaling van de levering aan de werkposten indien er tijdelijk een tekort aan componenten is.

In dit eindwerk zullen we het laatste belangrijke nadeel analyseren en proberen te kwantificeren. Tijdelijke tekorten aan componenten geven we als oorzaak op van productieonderbrekingen.

1.3 Productieonderbreking

Een *productieonderbreking* (production interruption) stopt de productie gedurende een zekere tijdspanne, veroorzaakt door verwachte en/of onverwachte gebeurtenissen. De internetsite van OEE Industry Standard [2003] definieert drie grote groepen onderbrekingen.

Een eerste groep is de *storingstijd* (breakdown time), i.e. de periode waarin een machine niet opbrengt omdat ze met technische problemen kampt. Het kan bijvoorbeeld gaan om een panne of een beschadigde machine [McKinnon, 2000]. De tijd die verloren gaat doordat de componenten niet klaarstaan voor assemblage valt ook onder storingstijd.

Een tweede groep is de *wachttijd* (waiting time), die beschouwd wordt als de belangrijkste reden voor productieonderbrekingen. Er zijn een negental oorzaken voor dit tijdverlies:

- set-up tijd: de tijd om op een bepaalde machine over te schakelen van het produceren van een bepaalde component naar de productie van een andere component. Wanneer een product op een andere manier gemaakt moet worden, moet men de machine aanpassen aan de nieuwe bewerking. Dit veroorzaakt wachttijden in de productie. Het kan ook gaan om een ander product dat dezelfde bewerking nodig heeft. De machine moet dan aangepast worden om in staat te zijn het nieuwe product te kunnen verwerken.
- opstarttijd: soms kunnen de componenten niet meteen verwerkt worden en moeten deze bijvoorbeeld afgekoeld en/of uitgespoeld worden.
- tekort aan operatoren bij de machine: de machine is beschikbaar maar door een inefficiënte organisatie wordt die niet bediend wegens bijvoorbeeld afwezigheid, pauze, vergadering of training van de operator.
- kwaliteitsprobleem: de productie wordt gestopt omdat het product niet voldoet aan de kwaliteitseisen.

- vervangen van een hulpmiddel: de machine wordt gestopt om hulpmiddelen zoals zaagbladen of koelvloeistoffen te vervangen of bij te vullen.
- bijvullen van grondstof: de productie wordt gestopt om materialen zoals etiketten, dozen en grondstoffen bij te vullen.
- autonoom onderhoud en/of schoonmaak: machines moeten meestal uitgeschakeld worden voor een onderhouds- of schoonmaakbeurt, wat de productie doet vertragen. Shift-onderhoud wordt meestal op een vast tijdstip gepland (bijvoorbeeld om de twee maanden).
- preventief onderhoud: machines worden op een vast tijdstip gedurende een bepaalde tijd onderhouden.

Een derde, minder bekende groep is de *lijnremmertijd* (line restraint time). Het productieproces kan geen output meer produceren omdat het geen input meer krijgt van de lijn of omdat de opslagplaats voor de output vol staat. In een kittingproces is deze gelijk aan de tijd die verloren gaat in de kitzone doordat een onderdeelbuffer leeg is of doordat de kitbuffer vol is. De *out-of-stock-kans* (i.e. de kans dat één van beide onderdeelvoorraden leeg is) zal uit de kittingmodellen berekend worden. We veronderstellen dat de kitbuffer een onbeperkte capaciteit heeft zodat we met deze laatste soort interruptie geen rekening dienen te houden. Dergelijke analyses zijn wel van belang om ongebalanceerde situaties tussen werkposten zichtbaar te maken.

Omdat het proces zich voordoet in een stochastische omgeving moet dit leveringsysteem als een wachtlijn bestudeerd worden. We bespreken de benodigde theorie voor het analyseren van wachtlijnmodellen in het volgende hoofdstuk.

Hoofdstuk 2

Markov-processen

Op het einde van dit hoofdstuk komen we tot de conclusie dat een kittingproces als een Markov-keten met continue-tijdparameter kan gemodelleerd worden. Deze uitkomst leidt tot belangrijke vereenvoudigingen voor het opstellen en analyseren van het kittingproces als model. We zullen dan ook in staat zijn prestatiematen voor de kittingmodellen te berekenen.

Vooraleer we dit kunnen concluderen bepalen we eerst het kittingproces als een wachtlijn in een stochastische omgeving. Een stochastisch proces dat voor ons van groot belang is, is het Markov-proces. In sectie 2.2 bespreken we waarom we kitting als een Markov-model beschouwen. Twee belangrijke Markoviaanse-aankomstprocessen worden ook beschreven: het Poisson-proces en het Onderbroken Poisson-proces (Interrupted Poisson-Proces). We zullen verderop de afkorting (uit het Engels) IPP gebruiken. Tenslotte bespreken we de modellering van het kittingproces als een Markov-keten met continue-tijdparameter.

2.1 Een wachttijmodel voor kitting in een stochastische omgeving

We kiezen ervoor om het kittingproces als een wachtlijn te modelleren. Omdat de aanvoer van componenten en de verwerkingstijd tot kit moeilijk volledig te controleren valt, worden onderdelen op voorhand in een buffer opgeslagen, waarna ze gekit worden.

Het aantal onderdelen in de buffers verandert in de tijd en wordt bepaald door het aankomst- en kitverwerkingsproces. Bijgevolg is het opgestelde model een wachttijmodel in een stochastische omgeving (Schoute, 1995, p.33). Een specifiek en voor ons belangrijk stochastisch proces is het Markov-proces.

2.2 Kittingproces als Markov-model

Het *Markov-proces* is een stochastisch proces dat aan de *Markov-eigenschap* voldoet, ook wel de *geheugenloze eigenschap* genoemd. Dit betekent dat de afgelopen tijd in een gegeven toestand niet bepalend is voor het toekomstige gedrag van dit proces (Schoute 1995, p.33). Door enkel de huidige toestand te observeren, kunnen de toekomstige kansen ingeschat worden; de kennis van het verleden is hierbij irrelevant. Ook in het kittingproces zal er geen enkel

verband zijn tussen de tijd vooraleer een nieuwe component in de buffer toekomt en de huidige hoeveelheid in deze container (zie verder).

We benaderen het aankomst- en verwerkingsproces volgens de exponentiële distributies om de geheugenloze eigenschap te kunnen gebruiken en het kittingproces als een Markov-model te beschouwen. Voor de tussenaankomsttijden en de verwerkingstijd gebruiken we de negatiefexponentiële verdeling. Verder veronderstellen we dat de aankomsttijden van de opeenvolgende componenten onafhankelijk zijn (independ and identically distributed of i.i.d.), zodat het aankomstproces een Poisson-proces is.

In het volgende deel bespreken we twee soorten Markoviaanse-aankomstprocessen (Markovian Arrival Process, MAP): het Poisson-proces en het Onderbroken Poisson-proces (Interrupted Poisson-process, IPP). Het kittingmodel waarin geen productieonderbrekingen voorkomen, modelleren we volgens het Poisson-proces terwijl de modellen waarin dit wel kan gebeuren worden als een IPP gemodelleerd.

2.2.1 Markoviaanse-aankomstprocessen

2.2.1.1 Poisson-proces

Een aankomstproces van klanten is een *Poisson-proces* als en slechts als de tussenaankomsttijden van deze klanten onderling onafhankelijke stochastische variabelen zijn met dezelfde exponentiële kansverdeling. Industriële aankomstprocessen zoals het kittingproces, worden vaak als een Poisson-proces opgevat.

De kenmerken van een Poisson-proces zijn [Roubos, 2009]:

- de klanten komen één voor één aan.
- de tijd tussen twee opeenvolgende aankomsten volgt een continue kansverdeling: de negatiefexponentiële verdeling. De tijden tussen opeenvolgende klanten zijn bovendien onafhankelijk van elkaar.

We spreken van een Poisson-proces met parameter λ . Deze parameter wordt *aankomstintensiteit* (arrival rate) genoemd en is gelijk aan de verwachtingswaarde van het aantal aankomsten in een tijdinterval van lengte één.

Dankzij de geheugenloze eigenschap bezit het Poisson-proces de *PASTA-eigenschap* (Poisson Arrivals See Time Average). Toegepast op het kittingproces zijn de stochastische eigenschappen van het proces op aankomsttijdstippen van onderdelen en op willekeurige tijdstippen gelijk. Deze eigenschap zal van belang zijn wanneer we de prestatie van het kittingproces bespreken (infra, p.54 en p.58).

Naast het basismodel stellen we ook een model op dat rekening houdt met tijdelijke onderbrekingen in de productie. Het volgende deel bespreekt het aankomstproces voor dit soort model.

2.2.1.2 Onderbroken Poisson-proces

Vooraleer het begrip *Onderbroken Poisson-proces* uit te leggen is het interessant om zijn algemene versie te kennen: het *Markov Modulated Poisson-process* (MMPP). Een MMPP is een dubbel stochastisch proces waarbij de intensiteiten van het Poisson-proces gedefinieerd worden op basis van de toestand van een onderliggend Markov-proces [Popescu, 2000]. Dit Markov-proces “moduleert” aldus het Poisson-process.

Een MMPP krijgt verschillende benamingen naargelang het aantal mogelijke toestanden en intensiteiten die al dan niet gelijk zijn aan elkaar. Een Markov-keten met twee mogelijke toestanden en twee verschillende intensiteiten noemt men het *Switched Poisson-process*. Indien beide intensiteiten gelijk zijn, heet het model een *Ordinary Poisson-process*.

Het *Onderbroken Poisson Proces (IPP)*, wat ons het meest interesseert, is een stochastisch proces waarbij twee toestanden mogelijk zijn en waarvan één van deze twee een intensiteit heeft die gelijk is aan nul [Ackerley, 2003]. Hier onderbreekt (interrupt) het Markov-proces als het ware het Poisson-proces. Merk op dat de tussenaankomsttijden zowel voor het Poisson-proces als voor het IPP onafhankelijk zijn. Voor het IPP zijn deze wel niet langer exponentieel maar hyperexponentieel verdeeld.

In de wachttijnanalyse, is een IPP onderverdeeld in twee periodes, namelijk de *actieve* en de *inactieve periode* [Heyman en Sobel, 1982]. We beginnen met een actieve periode en gedurende dit interval komen er componenten toe volgens een Poisson-proces met intensiteit λ . De lengte van deze periode heeft een exponentiële distributie met gemiddelde α^{-1} . Op het einde van een actieve periode begint een inactieve periode waarbij geen componenten aankomen; de lengte is ook exponentieel verdeeld met gemiddelde β^{-1} . Op het einde van deze periode begint een ander nieuwe actieve periode enzovoort. Alle inactieve en actieve periodes zijn i.i.d.

De parameter α heet *overgangsintensiteit* (transition rate) en kan beschouwd worden als een parameter die de transitie weergeeft om van een actieve naar een inactieve toestand over te gaan in een infinitesimaal tijdinterval. De parameter β stelt juist het tegengestelde voor. We noteren dat hoe korter de gemiddelde actieve periode (inactieve periode), hoe groter de intensiteit α (β) [Machihara, 2006].

Ter informatie, het IPP en het Switched Poisson-proces worden vaak gebruikt om verkeersstromen (traffic streams) in gedistribueerde communicatiesystemen te kenmerken. Meer specifiek worden ze toegepast in het onderzoek naar processen die te maken hebben met overstroom (overflow) in de telecomunicatie. Het IPP zorgt voor een eenvoudige beschrijving en vereenvoudigt bijgevolg de analytische studies [Tran-Gia, 1988]. Het werd voor het eerst geïntroduceerd door A. Kuczura als een proces voor overstroom in netwerken (circuit switch networks) [Machihara, 2006].

2.3 Kittingproces als een Markov-keten met continue-tijdparameter

Eerder hebben we gezien waarom we een kittingproces als een Markov-proces modelleren. Dit proces wordt *Markov-keten* genoemd indien zijn *toestandsruimte* (state space) discreet is; wat betekent dat de toestanden alleen waarden kunnen aannemen in een aftelbare verzameling. Doordat het aantal componenten in de buffers gehele waarden zijn (gaande van nul tot de capaciteitsbeperking van zijn respectievelijke buffer) beschouwen we ons proces als een Markov-keten.

Afhankelijk van de tijdparameter t , zijn er twee soorten Markov-ketens. Indien t een continue variabele is dan spreken we van een *Markov-keten met continue-tijdparameter* (Continuous time Markov Chain, afgekort als CTMC) en als t een discrete variabele is dan gaat het om een *discrete-tijd Markov-keten* (Discrete time Markov Chain, afgekort als DTMC). Zoals eerder vermeld is het belangrijk te onthouden dat de toestandsruimte bij een Markov-keten zowel bij een continue als bij een discrete tijdparameter altijd discreet is. We zijn geïnteresseerd in de evolutie van ons model in de tijd op een continue wijze.

Het proces als een Markov-keten modelleren laat ons toe de kans te bepalen dat een bepaalde toestand, i.e. het aantal componenten in de twee buffers, zich voordoet. Dit geeft ons achteraf de mogelijkheid om prestatie-maten te

berekenen zoals de gemiddelde bufferbezetting. In het volgende deel bespreken we de stappen om deze prestatie-maten te kunnen bepalen.

2.3.1 Modelspecificatie

Voordat we de Markov-theorie voor ketens met continue-tijdparameter introduceren, merken we op dat we een specifieke schrijfwijze voor vectoren en matrices hanteren. Vectoren geven we weer in vetgedrukte kleine letters en matrices in vet gedrukte hoofdletters. De elementen waaruit ze opgebouwd zijn, worden met hun corresponderende letter aangegeven maar niet in het vet gedrukt.

De waarschijnlijkheidsanalyse van Markov-ketens kan dankzij matrices en vectoren op een zeer compacte manier beschreven worden. Daarom is de matrixrekening van groot belang voor onze volledige analyse. In dit deel definiëren we eerst een belangrijke matrix, namelijk de generatormatrix \mathbf{Q} . Wanneer \mathbf{Q} bepaald is, krijgen we een volledig model voor onze Markov-keten met continue-tijdparameter gegeven zijn discrete toestandsruimte X . Op basis van deze matrix, kunnen we de stationaire waarschijnlijkheidsvector (steady state probability vector) berekenen. Elke waarde binnen deze vector geeft de kans weer dat een bepaalde toestand zich voordoet wanneer de keten evenwicht bereikt heeft. Tenslotte op basis van deze vector zullen we in staat zijn de prestatie-maten voor ons proces te bepalen. In het volgende hoofdstuk beschrijven we de procedure gevolgd om deze te berekenen.

2.3.2 Generatormatrix

We bepalen een stochastisch proces $X(t)$ als een Markov-keten met continue-tijdparameter waarbij $s, t, u \geq 0$ en alle niet-negatieve gehele waarden i, j en r tot de discrete toestandsruimte X behoren. Er geldt dat:

$$\mathbb{P}[X(t) = j \mid X(s) = i, X(u) = r, s \leq u < t] = \mathbb{P}[X(t) = j \mid X(s) = i]. \quad (2.1)$$

Deze definitie is gebaseerd op de Markov-eigenschap (supra, p.11). Stel nu:

$$p_{ij}(s, t) \equiv \mathbb{P}[X(t) = j \mid X(s) = i], \quad (2.2)$$

waarbij $t \geq s$.

We veronderstellen dat onze Markov-keten *homogeen* is. Een keten is homogeen indien alle *transitiefuncties* (transition function) $p_{ij}(s, t)$ enkel afhangen van het verschil $(t - s)$ en onafhankelijk zijn van de absolute tijdstippen s en t . Transitiefuncties geven de waarschijnlijkheid weer dat een toestand zich zal voordoen gegeven een huidige toestand. In het boek “Discrete Event Systems” geschreven door Cassandras en Lafortune [2008] voldoen transitiefuncties aan de *Chapman-kolmogorov vergelijking*.

We bewijzen dit door eerst de totale probabiliteitswet: $\mathbb{P}[A] = \sum_i \mathbb{P}[A \mid B_i] \mathbb{P}[B_i]$ toe te passen. We beschouwen $[X(u) = r]$ voor $s \leq u < t$ als de voorwaardelijke kans op de gebeurtenis $[X(t) = j \mid X(s) = i]$:

$$p_{ij}(t - s) = \mathbb{P}[X(t) = j \mid X(u) = r, X(s) = i] \cdot \mathbb{P}[X(u) = r \mid X(s) = i].$$

Het eerste deel van de vergelijking kan vereenvoudigd worden door de geheugenloze eigenschap te gebruiken:

$$\mathbb{P}[X(t) = j \mid X(u) = r, X(s) = i] = \mathbb{P}[X(t) = j \mid X(u) = r] = p_{rj}(t - u),$$

zodat,

$$p_{ij}(t - s) = \sum_r p_{ir}(u - s) \cdot p_{rj}(t - u),$$

wetende dat $s \leq u \leq t$. De geschreven formule is de Chapman-Kolmogorov vergelijking. Om die nu om te zetten in matrixvorm definiëren we:

$$\mathbf{H}(t - s) \equiv [p_{ij}(t - s)],$$

waarbij i en j tot de discrete toestandsruimte X behoren. We stellen vast dat $\mathbf{H}(s - s) = \mathbf{I}$ waarbij \mathbf{I} de identiteitsmatrix is, omdat de kans dat het proces in dezelfde toestand zit na tijdstip $s - s$ gelijk is aan één en de kans dat het evolueert naar een andere toestand gelijk is aan nul. De matrix $\mathbf{H}(t - s)$ is een *transitiematrix*. Deze matrix geeft de kansen weer om in een interval van lengte $t - s$ van de ene naar de andere toestand te gaan. De bovengeschreven formule kan dan omgezet worden in de matrixformule:

$$\mathbf{H}(t - s) = \mathbf{H}(u - s) \cdot \mathbf{H}(t - u). \quad (2.3)$$

Stel in de vooraafgaande de formule $t = t + \Delta t$, $s = 0$ en $u = t$. Dan krijgen we,

$$\mathbf{H}(t + \Delta t) = \mathbf{H}(t) \cdot \mathbf{H}(t + \Delta t - t) = \mathbf{H}(t) \cdot \mathbf{H}(\Delta t),$$

gegeven dat $0 \leq t \leq t + \Delta t$. De vergelijking wordt daarna aan beide zijden verminderd met de matrix $\mathbf{H}(t)$, met als resultaat:

$$\mathbf{H}(t + \Delta t) - \mathbf{H}(t) = \mathbf{H}(t) \cdot [\mathbf{H}(\Delta t) - \mathbf{I}].$$

Voor elke identiteitssmatrix \mathbf{I} geldt de elementaire eigenschap $\mathbf{I} \cdot \mathbf{A} = \mathbf{A}$. We delen beide leden door Δt en nemen de limiet ervan voor $\Delta t \rightarrow 0$ zodat:

$$\lim_{\Delta t \rightarrow 0} \frac{\mathbf{H}(t + \Delta t) - \mathbf{H}(t)}{\Delta t} = \mathbf{H}(t) \cdot \lim_{\Delta t \rightarrow 0} \frac{\mathbf{H}(\Delta t) - \mathbf{I}}{\Delta t}. \quad (2.4)$$

De linkerzijde van de vergelijking geeft ons de partiële afgeleide van $\mathbf{H}(t)$ voor t . We veronderstellen dat de afgeleide van alle elementen van de matrix $\mathbf{H}(t)$ bestaat. Elk element is gelijk aan een transitiefunctie $p_{ij}(t)$ waarbij de combinatie i en j verschilt. Laten we \mathbf{Q} definiëren als:

$$\mathbf{Q} \equiv \lim_{\Delta t \rightarrow 0} \frac{\mathbf{H}(\Delta t) - \mathbf{I}}{\Delta t} \quad (2.5)$$

In de Engelse literatuur heeft de matrix meerdere benamingen zoals de *transitie-intensiteitmatrix*, *intensiteit-generatormatrix* of de *infinitesimale generator*. Wij houden het bij de *generatormatrix*. Door de limiet van de transitiematrix voor $\Delta t \rightarrow 0$ te nemen, bekomen we de generatormatrix. Deze matrix is het continue geval van de transitiematrix. Het geeft de kansen weer om in een infinitesimaal tijdinterval van de ene naar de andere toestand te gaan. De vergelijking kan gereduceerd worden tot:

$$\frac{d\mathbf{H}(t)}{dt} = \mathbf{H}(t) \cdot \mathbf{Q}. \quad (2.6)$$

Deze differentiaalvergelijking wordt aangeduid als de *forward Chapman-Kolmogorov vergelijking*. Omdat de Markov-keten homogeen is, mogen we \mathbf{Q} als een constante matrix beschouwen. In het volgende deel bepalen we de stationaire waarschijnlijkheidsvector aan de hand van deze formule.

We definiëren de generatormatrix als $\mathbf{Q} = (q_{ij})$ waarbij i en j tot de toestandruimte X behoren. De parameter q_{ij} is gelijk aan de intensiteit om van toestand i naar j te evolueren. De voorwaarden waaraan \mathbf{Q} moet voldoen zijn dat:

- de som van alle waarden voor elke rij gelijk is aan nul,
- alle niet-diagonale waarden q_{ij} (met $i \neq j$) positief zijn en
- alle diagonale waarden q_{ii} negatief zijn.

De eerste eigenschap betekent dat een huidige toestand geëvolueerd moet zijn naar een nieuwe - verschillende of zelfde - toestand. Deze eigenschappen gelden voor alle generatormatrices.

2.3.3 Stationaire waarschijnlijkheidsvector

Een *waarschijnlijkheidstoestand* geeft de kans weer dat een toestand zich voordoet op een gegeven tijdstip. Deze kan bepaald worden aan de hand van de vergelijking:

$$\pi(t) = \pi(0) \cdot \mathbf{H}(t). \quad (2.7)$$

Dit betekent dat de waarschijnlijkheid dat een toestand zich voordoet op tijdstip t gelijk is aan de vermenigvuldiging van de initiële toestandswaarschijnlijkheid met zijn transitiematrix. We bekomen deze vergelijking aan de hand van de totale probabiliteitswet:

$$\mathbb{P}[X(t) = i] = \mathbb{P}[X(0) = j] \cdot \mathbb{P}[X(t) = i \mid X(0) = j]. \quad (2.8)$$

Als we het linker- en rechterlid van vergelijking (2.6) met de beginvector $\pi(0)$ vermenigvuldigen en gebruik maken van vergelijking (2.7) dan bekomen we:

$$\pi(0) \cdot \frac{d\mathbf{H}(t)}{dt} = \frac{d\pi(t)}{dt} = \pi(t) \cdot \mathbf{Q}. \quad (2.9)$$

Het voornaamste doel van een Markovmodel-analyse is het bepalen van de kans om de keten in een bepaalde toestand te vinden op een gegeven tijdstip [Cassandras en Lafortune, 2008]. Echter, het bekomen van de expliciete uitdrukking van de waarschijnlijkheidsvector (probability vector) $\pi(t)$ is meestal een zeer ingewikkelde taak. Bovendien is het regimetoestandgedrag (steady-state) van een systeem vaak belangrijker dan zijn gedrag op verschillende tijdstippen. Dit is dan ook veel gemakkelijker te berekenen en te analyseren dan zijn zogenaamd transient-gedrag. We concentreren ons op het bepalen van de limiet voor t gaande naar oneindig:

$$\pi = \lim_{t \rightarrow \infty} \pi(t). \quad (2.10)$$

Het symbool π is gelijk aan de *stationaire waarschijnlijkheidsvector*. Deze verzamelt alle *stationaire waarschijnlijkheidstoestanden*, i.e. de kansen dat een bepaalde toestand zich voordoet wanneer de keten evenwicht bereikt heeft. Indien de tijdparameter naar oneindig gaat ($t \rightarrow \infty$), dan gaat zijn afgeleide naar nul ($\frac{d\pi}{dt} \rightarrow 0$). De vector $\pi(t)$ is niet meer afhankelijk van t en zijn elementen convergeren naar een vaste waarde. Bijgevolg zal vergelijking (2.9) vereenvoudigen tot:

$$\pi \cdot \mathbf{Q} = 0 \quad (2.11)$$

De vermenigvuldiging van de stationaire waarschijnlijkheidsvector met zijn generatormatrix is gelijk aan nul. We gebruiken deze formule om de prestatiematen te berekenen. Merk op dat deze vergelijking, de vector π slechts op een factor na bepaalt. De *normalisatievoorwaarde* $\langle \pi, 1 \rangle = 1$ uitgelegd in sectie 3.2.3, laat toe deze factor te bepalen.

In het volgende hoofdstuk passen we de verschillende stappen om de prestatiematen te bepalen toe op het basis-kittingmodel.

Hoofdstuk 3

Basis-kittingmodel

Het bestudeerde kittingproces beschouwt twee basiscomponenten die na productie in hun respectievelijke buffers geplaatst worden. Om de impact van inefficiëntie in de productie na te gaan stellen we drie modellen op: het basismodel (die we in dit hoofdstuk zullen bespreken), het 1IPP-model waarbij één van beide componenten aan productieonderbreking onderhevig is en het 2IPP-model waarbij beide componenten dit ervaren. Daarna vergelijken we deze modellen aan de hand van prestatiematen.

In dit hoofdstuk beschrijven we eerst een kittingmodel uit de literatuur en leggen we de verschillen uit met ons basismodel. We sommen de belangrijkste veronderstellingen op die ook voor de uitgebreide modellen zullen gelden. In sectie 3.2 passen we de stappen die we in het vorige hoofdstuk besproken hebben, toe op het basismodel om de prestatie van dit model te bepalen (supra, p.13). Deze stappen worden daarna uitgevoerd in MATLAB. We beschrijven de gehanteerde methodologie in sectie 3.3 waarin we de belangrijkste procedures om ons programma te versnellen en te verbeteren uitleggen. In sectie 3.4 beschrijven we hoe we de prestatiematen in MATLAB berekend hebben. Op basis van de berekende prestatie-uitkomsten bekomen we numerieke voorbeelden die we in sectie 3.5 zullen bespreken.

3.1 Beschrijving van het basismodel

3.1.1 Literatuur

De meeste auteurs beschouwen een kittingproces als een wachtlijnsysteem dat zich in een stochastische omgeving bevindt. Voor redenen uitgelegd in het vorige hoofdstuk zullen we dit ook doen (supra, p.11). De elementen die het kittingmodel als een wachtlijnsysteem kenmerken, staan hieronder vermeld.

Hopp en Simon [1989] hebben een model ontwikkeld dat vaak gebruikt wordt om de prestatie van een assemblagelijn te analyseren. In hun artikel “Bounds and Heuristics for assembly-like queues” hebben ze een model met exponentieel verdeelde verwerkingstijden en tussenaankomsttijden volgens een Poisson-proces opgesteld. Ze leiden uit hun model grenzen af voor de capaciteit van de buffers. Dit model is vooral gebaseerd op het model van Lipper en Sengupta [1986]. De methode van Hopp en Simon is gemakkelijker te implementeren en de bepaling van een optimale bovengrens voor de capaciteit is nauwkeuriger, maar het is wel beperkt tot processen met twee basisonderdelen. De

methode van Lipper en Sengupta kan daarentegen toegepast worden op meer algemene systemen. Ook wij zullen ons in de analyse beperken tot twee soorten componenten.

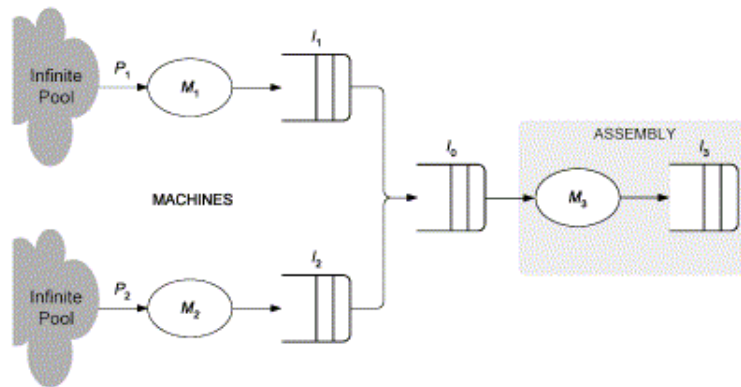
Som, Wilhelm en Disney [1994] beschouwen een kittingproces als een leveringssysteem voor assemblage dat vooral gebaseerd is op het model van Hopp en Simon. Figuur 3.1 geeft het schema voor dit proces.

De invoermachines die de componenten voorbereiden op assemblage zijn M_1 en M_2 , en de assemblagemachine is M_3 . De symbolen I_1 en I_2 vertegenwoordigen de buffers met eindige capaciteit voor de twee soorten onderdelen, I_0 de buffer voor de kits en I_3 de buffer voor het eindproduct.

De buffers I_1 en I_2 worden gevuld vanuit een oneindig aanbod. Indien er voldoende componenten van beide soorten beschikbaar zijn om een kit samen te stellen, dan worden de componenten vanuit I_1 en I_2 in buffer I_0 gesorteerd. Als een kit nog niet gemaakt kan worden, zal het beschikbare stuk in zijn respectievelijke buffer wachten tot de andere component beschikbaar is. Het is dus onmogelijk om tegelijkertijd een positief voorraadniveau in de buffers I_1 en I_2 te hebben. De voorraadpositie M in een buffer is verbonden aan de voorraadpositie $-M$ van de andere buffer. De voorraadniveaus kunnen enkel gelijk zijn als beide buffers leeg zijn.

Als de buffer I_0 leeg is en het werkstation M_3 vrij, dan wordt de kit meteen verwerkt tot een assemblagestuk. Indien dit niet het geval is, dan wordt de kit bewaard in buffer I_0 . Wanneer de werkpost M_3 een assemblage vervolledigd heeft, dan wordt het eindproduct geleverd aan buffer I_3 . De eindproducten worden uit de buffer I_3 gehaald en de vraag naar eindproducten wordt meteen ingewilligd als er stock aanwezig is.

We bespreken nu de verschillen en gelijkenissen van dit model met de modellen die wij zullen opstellen.



Figuur 3.1: Kittingmodel volgens Som, Wilhelm en Disney

3.1.2 Verschil met ons model

In het model van Som, Wilhelm en Disney [1994] worden twee processen bestudeerd, namelijk het assemblage- en kittingproces. We concentreren ons op het bestuderen van het kittingproces waarin de onderdelen, nadat ze in het proces ingevoegd zijn, exponentieel naar de kitbuffer gestuurd worden om een kit te vormen.

Hier merken we een belangrijk verschil op. We veronderstellen in onze modellen dat de verwerkingstijd exponentieel verdeeld is zodat een positief voorraadniveau tegelijkertijd in buffer 1 en 2 mogelijk is.

Een belangrijke gelijkenis met het beschreven model is de veronderstelling dat de buffers een eindige capaciteit hebben. Uiteraard hebben voorraadbuffers altijd een eindige capaciteit. Echter, als de capaciteit groot genoeg

is, kunnen we een proces met eindige capaciteit aan de hand van een model met ongelimiteerde capaciteit goed benaderen. Dit betekent dat er altijd voldoende ruimte is voor de aankomende onderdelen, wat de analyse vereenvoudigt. Helaas geldt de veronderstelling van de oneindige buffercapaciteit niet voor kittingprocessen. Als de capaciteit als oneindig verondersteld wordt, dan zal het model degraderen tot een onstabiel stochastisch model. Dit was al aangetoond door Latouche [1981] die wachtlijnen met gepaarde klanten bestudeerde. We kunnen zijn analyse zien als een abstractie van een kittingproces met twee soorten onderdelen. In het artikel “Assembly-like queues” bevestigt Harrison [1973] dat, om de stabiliteit in de operaties van een kittingproces te verzekeren, het noodzakelijke is een restrictie te leggen op de grootte van de buffer. Onder deze veronderstelling is de kans op lange termijn om een bepaalde voorraadpositie te hebben gelijk, en onafhankelijk van de actuele voorraadpositie. We veronderstellen dat de bovengrens voor de buffercapaciteit voor de twee componenten gelijk is aan C_1 en C_2 .

3.1.3 Beschrijving van een wachtlijnsysteem

We specificëren de belangrijkste elementen van het kittingproces als een wachtlijnsysteem (Bruneel, 2008, p.38).

Het meest elementaire element in de wachtlijntheorie is de *klant*. In onze analyse zijn de klanten alle componenten die in een container wachten om achteraf als een kit verwerkt te worden.

De *wachtlijndiscipline* is een tweede belangrijk aspect voor de modelbepaling. Het beschrijft de volgorde waarin en de manier waarop klanten uit de wachtlijn in de service-eenheid binnengelaten worden. We veronderstellen dat de componenten volgens hun aankomstvolgorde verwerkt worden tot een kit (first-come-first-served).

De twee volgende elementen zijn de ingangsvariabelen van ons model. De parameters *aankomsttijden* en *servicetijden* beschrijven samen de vraag van de klantenstroom, i.e. de stroom van de onderdelen naar de de kitbuffer-area.

Het *aankomstproces* van een wachtlijnsysteem beschrijft de mate waarin klanten zich bij het systeem aanbieden. In een kittingproces gaat het om een proces waarbij componenten aankomen in de buffers om daarna als een kit verwerkt te worden. We veronderstellen dat de aankomsten van de componenten zich willekeurig en onafhankelijk van elkaar voordoen zodat ze per definitie volgens een Poisson-proces met intensiteit λ aankomen (supra, p.12). Voor de uitgebreide modellen komen de componenten toe volgens een IPP met intensiteit λ in de actieve periodes (supra, p.13).

De *verwerkingstijd* van een klant is gelijk aan de tijd die de service-eenheid nodig heeft om deze klant volledig te bedienen. In de kittinganalyse is dit gelijk aan de tijd die nodig is om de componenten tot een kit te verwerken. Verwerkingstijden zijn meestal negatief exponentieel verdeeld, wat we ook in ons model veronderstellen. Het verwerkingsproces tot een kit wordt bepaald door de verwerkingsintensiteit μ (service rate).

Ten laatste is de *capaciteit* of *bufferruimte* een belangrijke ingangsvariabele. In het kittingproces geeft het de bovengrens weer van de buffers, i.e. een geheel getal dat aangeeft hoeveel onderdelen maximaal (tezelfdertijd) kunnen worden opgevangen. We veronderstellen in ons model dat de twee onderdelenbuffers een eindige grootte C_1 en C_2 hebben. Het gevaar bestaat dus dat onderdelen moeten geweigerd worden omdat ze vol zitten; men spreekt dan van *verlies* (loss). Om dit risico na te gaan bespreken we later de verlieskans in de buffer (infra, p.37).

De gebruikte symbolen in dit eindwerk worden in de bijlage A.1. “Lijst van de gebruikte symbolen” weergegeven. Voor de duidelijkheid sommen we de belangrijkste assumpties voor onze modellen op.

3.1.4 Basisveronderstellingen

Onze veronderstellingen zijn:

- het model beperkt zich tot twee componenten waarbij elke type nodig is om een kit op te stellen. We zullen deze component 1 en component 2 noemen.
- in het basismodel komen de componenten toe volgens een Poisson-proces terwijl dat in de uitgebreide modellen volgens een IPP gebeurt. Een aankomst van component 1 (of 2) behoort respectievelijk tot buffer 1 (of 2).
- de verwerkingstijd, i.e. de tijd die nodig is om de onderdelen te verwerken tot een kit, volgt de exponentiële distributie.
- de bufferruimtes 1 en 2 zijn langs boven begrensd door respectievelijk de capaciteit C_1 en C_2 . Wanneer de buffer zijn capaciteit bereikt heeft, worden bijkomende aankomsten niet in rekening gebracht. Er is geen restrictie op de capaciteit van de kitbuffer.

3.2 Constructie en analyse van het basismodel

We willen de prestatie van het kittingmodel nagaan. De stappen die we ondernemen om de prestatiematen te berekenen zijn:

1. het bepalen van de evenwichtsvergelijkingen van een model aan de hand van zijn toestandsdiagram,
2. het omzetten van deze vergelijking in een matrixvorm: de generatormatrix Q ,
3. het berekenen van de stationaire waarschijnlijkheidsvector π en
4. het berekenen van de prestatiematen.

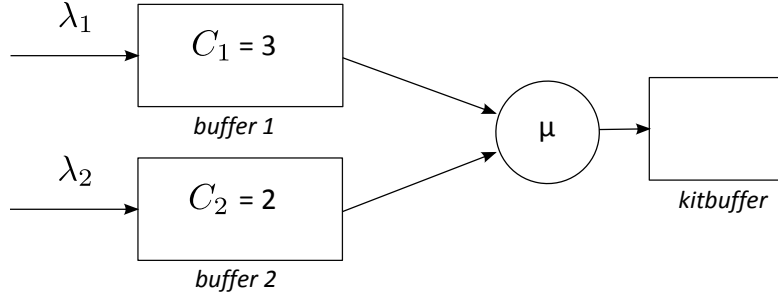
In deze sectie bespreken we de drie eerste stappen. De laatste stap wordt in het deel “Implementatie in MATLAB” uitgelegd. We beschrijven hier wel de prestatiematen van belang voor het kittingproces. Hoewel deze stappen voor het basismodel uitgelegd zijn, gelden ze ook voor de twee uitgebreide modellen.

3.2.1 Bepaling van het basismodel

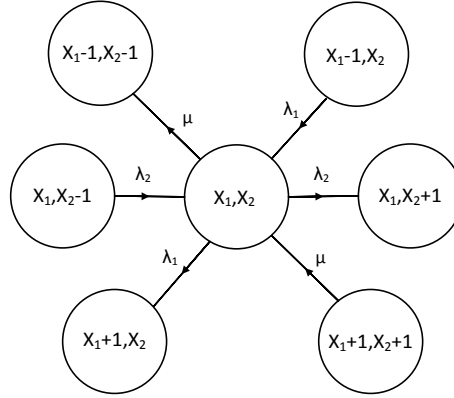
Twee stochastische variabelen bepalen de toestand van ons kittingproces: X_1 is gelijk aan het aantal componenten van type 1 in buffer 1 en X_2 is het aantal componenten van type 2 in buffer 2. Deze twee variabelen worden verzameld als een *toestand*, i.e. het koppel (X_1, X_2) , en nemen waarden aan binnen hun toestandsruimte. Het basismodel wordt dus gekenmerkt door een tweedimensionale toestand.

Figuur 3.2 stelt het schema voor van ons basismodel. De componenten komen respectievelijk toe met intensiteit λ_1 en λ_2 in de buffers 1 en 2. Daarna worden de componenten met intensiteit μ naar de kitbuffer gestuurd. Als voorbeeld, nemen we een capaciteit voor buffer 1 gelijk aan drie en voor buffer 2 gelijk aan twee.

Figuur 3.3 toont een fragment van het *toestandsdiagram* (transition rate diagram) voor het basis-kittingmodel. Dit is een hulpmiddel om de relaties tussen de toestanden voor te stellen. De twee waarden X_1 en X_2 (in een cirkel



Figuur 3.2: Schema voor het basismodel



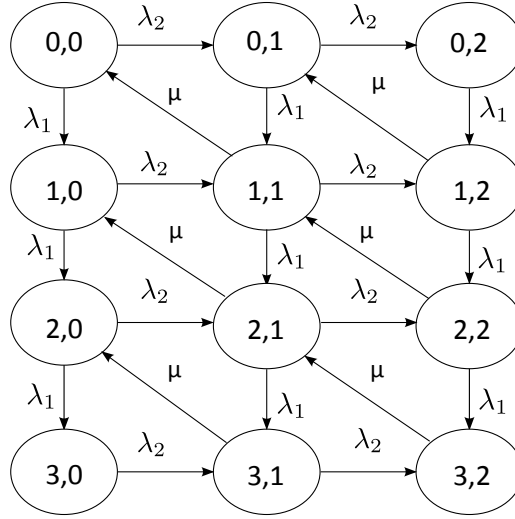
Figuur 3.3: Fragment van het toestandsdiagram voor het basismodel

geplaatst) geven het aantal componenten van type 1 en 2 in de buffers weer. De pijlen die de cirkels verbinden, tonen de richting aan waarnaar de toestand met een zekere intensiteit kan evolueren. Een toestand (X_1, X_2) kan bijvoorbeeld niet rechtstreeks naar een toestand $(X_1 + 1, X_2 + 1)$ gaan maar het omgekeerde is bijvoorbeeld wel mogelijk. Dit doet zich voor wanneer component 1 en 2 gestuurd worden naar de kitbuffer. Als een component van type 1 zich in het proces aandient dan evolueert toestand (X_1, X_2) naar toestand $(X_1 + 1, X_2)$.

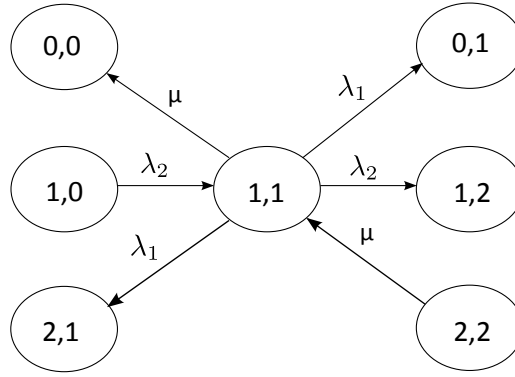
Figuur 3.4 schetst het volledige toestandsdiagram voor ons voorbeeld ($C_1 = 3, C_2 = 2$). Vanuit dit diagram, kunnen we de evenwichtsvergelijkingen bepalen. We bepalen bijvoorbeeld de evenwichtsvergelijking waarvoor het aantal componenten in beide buffers gelijk is aan één. In figuur 3.5 is zijn toestandsdiagram getekend. We leiden daaruit af dat componenten ofwel samengesteld worden tot een kit zodat de nieuwe toestand $(0, 0)$ wordt, ofwel komt er een component van type 1 of 2 toe zodat de nieuwe toestand respectievelijk $(2, 1)$ of $(1, 2)$ wordt. Deze transitie zijn de uitgaande pijlen van onze toestandsdiagram en worden respectievelijk bepaald door μ, λ_1 en λ_2 . Voor wat betreft de inkomende pijlen, zijn de mogelijke begintoestanden om naar toestand $(1, 1)$ te evolueren: $(0, 1)$, $(1, 0)$, $(1, 1)$, en $(2, 1)$. Deze zijn verbonden aan toestand $(1, 1)$ met dezelfde intensiteit als de uitgaande pijlen. Door de inkomende en uitgaande pijlen te bepalen voor elke toestand en die gelijk te stellen aan elkaar, bekomen we de evenwichtsvergelijking. De evenwichtsvergelijking voor toestand $(1, 1)$ is gelijk aan:

$$\pi_{1,0} \cdot \lambda_2 + \pi_{0,1} \cdot \lambda_1 + \pi_{2,2} \cdot \mu = \pi_{1,1} \cdot (\lambda_2 + \lambda_1 + \mu). \quad (3.1)$$

We definiëren de stationaire waarschijnlijkheidstoestand door het symbool $\pi_{i,j}$ (supra, p.17). Het is gelijk aan de kans dat buffer 1 en 2 respectievelijk i en j onderdelen omvatten. Deze waarschijnlijkheidstoestanden worden



Figuur 3.4: Volledig toestandsdiagram voor het basismodel waarbij $C_1 = 3$ en $C_2 = 2$



Figuur 3.5: Toestandsdiagram voor toestand (1,1)

in de derde stap bepaald. We stellen de generatormatrix op door alle evenwichtsvergelijkingen te bepalen en de verschillende intensiteiten in de matrix te plaatsen. Voor de twee uitgebreide modellen gaan we op dezelfde manier te werk. Het enige verschil is dat er meer evenwichtsvergelijkingen zijn doordat de toestandsruimte groter is voor eenzelfde capaciteitsbeperking (infra p.43 en p.46). We bepalen nu de generatormatrix voor ons voorbeeld.

3.2.2 Generatormatrix

De capaciteit van de buffers 1 en 2 bepalen de toestandsruimte van ons model. Voor het basismodel is het totaal aantal toestanden gelijk aan $(C_1 + 1) \cdot (C_2 + 1)$. Wanneer $C_1 = 3$ en $C_2 = 2$, zijn er bijgevolg twaalf mogelijke toestanden met voor elke toestand een evenwichtsvergelijking. Nadat alle vergelijkingen bepaald zijn, worden deze omgezet in een matrixvorm: we verkrijgen de generatormatrix Q . Omdat de som van de elementen in een rij gelijk moet zijn aan nul, geven we, ter vereenvoudiging in de notatie, de diagonale waarde (die gelijk is aan de negatieve som van de elementen in die rij) weer als een punt (supra, p.16). De generatormatrix voor ons voorbeeld wordt weergegeven in figuur 3.6.

Voor wat betreft de betekenis van zijn elementen, vertegenwoordigt de rijpositie dezelfde toestand als de kolompositie met als enig verschil dat de rij een huidige toestand voorstelt terwijl de kolom een nieuwe toestand weergeeft. In

$$\begin{array}{l}
(0,0) \\
(0,1) \\
(0,2) \\
(1,0) \\
(1,1) \\
(1,2) \\
(2,0) \\
(2,1) \\
(2,2) \\
(3,0) \\
(3,1) \\
(3,2)
\end{array}
\begin{bmatrix}
. & \lambda_2 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & . & \lambda_2 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & . & 0 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & . & \lambda_2 & 0 & \lambda_1 & 0 & 0 & 0 & 0 & 0 \\
\mu & 0 & 0 & 0 & . & \lambda_2 & 0 & \lambda_1 & 0 & 0 & 0 & 0 \\
0 & \mu & 0 & 0 & 0 & . & 0 & 0 & \lambda_1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & . & \lambda_2 & 0 & \lambda_1 & 0 & 0 \\
0 & 0 & 0 & \mu & 0 & 0 & 0 & . & \lambda_2 & 0 & \lambda_1 & 0 \\
0 & 0 & 0 & 0 & \mu & 0 & 0 & 0 & . & 0 & 0 & \lambda_1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & . & \lambda_2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mu & 0 & 0 & 0 & . & \lambda_2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu & 0 & 0 & 0 & .
\end{bmatrix}$$

Figuur 3.6: Generatormatrix \mathbf{Q} voor het basismodel waarbij $C_1 = 3$ en $C_2 = 2$

ons voorbeeld stelt de derde rij de huidige toestand $(0, 2)$ voor en de zesde kolom de nieuwe toestand $(1, 2)$. Indien we naar de waarde van de derde rij en de zesde kolom kijken, vinden we de parameter λ_1 terug. Deze intensiteit bepaalt het aankomstproces van component 1.

Iedere evenwichtsvergelijking bevindt zich in één kolom. De evenwichtsvergelijking voor toestand $(1, 1)$ bijvoorbeeld bevindt zich in de vijfde kolom. In deze kolom krijgen de rijwaarden die de toestanden $(0, 1)$, $(1, 0)$, $(1, 1)$ en $(2, 1)$ vertegenwoordigen, respectievelijk de waarde λ_1 , λ_2 , $-\lambda_1 - \lambda_2 - \mu$ en μ .

De volgende stap is om op basis van de generatormatrix de stationaire waarschijnlijkheidsvector te bepalen. We definiëren nogmaals kort deze vector maar nu aangepast aan onze kittinganalyse. Daarna tonen we hoe we deze zullen bepalen.

3.2.3 Stationaire waarschijnlijkheidsvector

Wat is de kans dat het kittingproces in regime zich in een bepaalde toestand bevindt, rekening houdend met de capaciteit van de buffers? De stationaire waarschijnlijkheidsvector is de vector van al deze waarden. Voor het voorbeeld waarbij $C_1 = 3$ en $C_2 = 2$ formuleren we de stationaire waarschijnlijkheidsvector $\boldsymbol{\pi}$ als:

$$\boldsymbol{\pi} = [\pi_{0,0}, \pi_{0,1}, \pi_{0,2}, \pi_{1,0}, \pi_{1,1}, \pi_{1,2}, \pi_{2,0}, \pi_{2,1}, \pi_{2,2}, \pi_{3,0}, \pi_{3,1}, \pi_{3,2}]. \quad (3.2)$$

Nadat we de vector voor ons voorbeeld gedefinieerd hebben, bespreken we de manier waarop we deze zullen bepalen. We vertrekken vanuit de vergelijking bepaald in het vorige hoofdstuk (supra, p.17):

$$\boldsymbol{\pi} \cdot \mathbf{Q} = \mathbf{0}. \quad (3.3)$$

Deze vergelijking betekent dat het product van de rijvector $\boldsymbol{\pi}$ met de generatormatrix \mathbf{Q} gelijk is aan een nulrijvector. Daarnaast bepalen we de *normeringsvoorwaarde*:

$$\sum_{i,j} \pi_{i,j} = 1. \quad (3.4)$$

Deze vergelijking verklaart dat de som van de toestandswaarschijnlijkheden in regime gelijk moet zijn aan één. De vector π kan niet volledig bepaald worden met enkel vergelijking (3.3), deze wordt slechts op een factor na bepaald. Door de normeringsvoorwaarde aan het stelsel van evenwichtsvergelijkingen toe te voegen, kunnen we een unieke oplossing voor de vector π vinden. Stel de eenvoudige generatormatrix:

$$Q = \begin{bmatrix} -a & a \\ b & -b \end{bmatrix}.$$

Uit deze matrix en de eerste vergelijking volgt dat de twee evenwichtsvergelijkingen gelijk zijn aan:

$$\begin{cases} \pi_1 \cdot (-a) + \pi_2 \cdot b &= 0 \\ \pi_1 \cdot a + \pi_2 \cdot (-b) &= 0 \end{cases}$$

Gegeven de waarden a en b hebben we onvoldoende informatie om het stelsel op te lossen omdat de vergelijkingen lineair afhankelijk zijn. Daarom maken we gebruik van de normeringsvoorwaarde die in ons voorbeeld gelijk is aan:

$$\pi_1 + \pi_2 = 1. \quad (3.5)$$

Omdat de twee vergelijkingen uit de formule (3.3) gelijk zijn aan elkaar, zullen we de waarden van de laatste kolom van de generatormatrix gelijk stellen aan één. Op die manier houden we rekening met de normeringsvoorwaarde. Dit wordt:

$$\begin{bmatrix} \pi_1 & \pi_2 \end{bmatrix} \cdot \begin{bmatrix} -a & 1 \\ b & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \end{bmatrix}. \quad (3.6)$$

De matrix $\begin{bmatrix} -a & 1 \\ b & 1 \end{bmatrix}$ noteren we als $\langle Q | \mathbf{1} \rangle$. Door beide leden met de matrix $\langle Q | \mathbf{1} \rangle^{-1}$ te vermenigvuldigen, vallen de generatormatrix en zijn inverse in het linkerlid weg, omdat de vermenigvuldiging van een matrix met zijn inverse gelijk is aan de eenheidsmatrix. Het geeft:

$$\begin{bmatrix} \pi_1 & \pi_2 \end{bmatrix} \cdot \langle Q | \mathbf{1} \rangle \cdot \langle Q | \mathbf{1} \rangle^{-1} = \begin{bmatrix} 0 & 1 \end{bmatrix} \cdot \langle Q | \mathbf{1} \rangle^{-1}, \quad (3.7)$$

zodat:

$$\begin{bmatrix} 0 & 1 \end{bmatrix} \cdot \langle Q | \mathbf{1} \rangle^{-1} = \begin{bmatrix} \pi_1 & \pi_2 \end{bmatrix}. \quad (3.8)$$

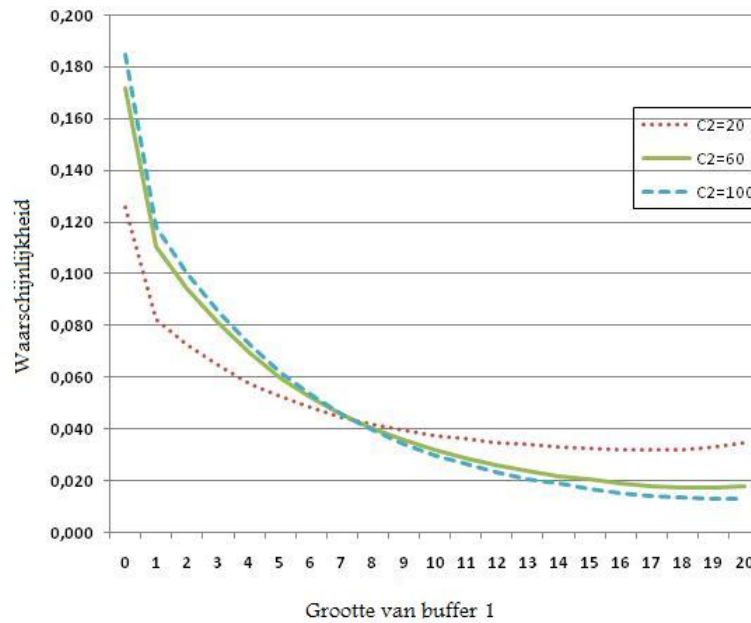
We kunnen deze vergelijkingen veralgemenen tot:

$$\pi \cdot Q = b \quad (3.9)$$

Wanneer $C_1 = 3$ en $C_2 = 2$ is de rijvector b een vector waarin de eerste elf kolomwaarden gelijk zijn aan nul en de twaalfde aan één. In de laatste kolom van de generatormatrix vervangen we zijn elementen door het getal 1.

Voor we de berekende prestatie-maten beschrijven, geven we op figuur 3.7 de distributie van de bufferbezetting van buffer 1 met een capaciteit van twintig weer. Voor iedere toestand in buffer 1 sommeren we alle waarschijnlijkheden in buffer 2 (gaande van nul tot zijn capaciteitsbeperking). We bekomen een lijn in de grafiek waarin de som van alle waarschijnlijkheden gelijk is aan één. De gebruikte parameters zijn terug te vinden in tabel 1 in de bijlage A.3. “Ingangparameters”.

We observeren dat bij toename van C_2 , de kans op een lage bufferbezetting toeneemt en die op een hoge bufferbezetting afneemt. Bij toename van C_2 zal er minder verlies zijn in buffer 2. Hierdoor kunnen er meer kits gemaakt worden en zullen er minder componenten in buffer 1 aanwezig zijn. Het blijkt ook dat naarmate C_2 groter wordt, de waarschijnlijkheden naar eenzelfde waarde convergeren: de lijnen voor $C_2 = 60$ en $C_2 = 100$ zijn quasi gelijk aan elkaar. Deze waarneming is te verklaren door te bemerken dat het effect op verlies afneemt naarmate de capaciteit toeneemt.



Figuur 3.7: Distributie van de bufferbezetting van buffer 1 ($C_1 = 20$)

3.2.4 Prestatiematen

Het doel van deze analyse is om de impact van productie-inefficiënties in een kittingproces te kwantificeren. We definiëren de *prestatie-maten* voor ons model:

1. de gemiddelde bezetting in de buffer met zijn bijbehorende variantie,
2. de kans dat één van beide buffers leeg is,
3. de kans dat elke soort buffer vol is, en
4. de verlieskans in de buffers

De eerste prestatie-maat is de *gemiddelde bezetting in de buffer*. In de wachtrijtheorie valt deze prestatie-maat onder de noemer *gemiddelde systeembevolking* (Bruneel, 2008, p.46). Het geeft het gemiddelde aantal klanten weer dat

zich na lange tijd in het systeem bevindt. Een doeltreffende gemiddelde bufferbezetting ligt niet te dicht bij de capaciteitsbeperking om niet constant met een volle buffer te zitten (want daardoor gaan er componenten vaker verloren); deze moet ook steeds positief blijven omdat kitting anders niet mogelijk is. We bepalen daarnaast de *variantie*, die van belang is omdat ze de mate weergeeft waarin de waarden onderling verschillen.

De twee volgende begrippen, de *kans dat één van beide buffers leeg is* en de *kans dat elke soort buffer vol is*, zijn vanzelfsprekend. Indien één van beide buffers leeg is dan kan de kitting niet doorgaan en indien een buffer vol is dan gaan er componenten verloren. Uiteraard hoe groter deze kansen, hoe onstabiel ons proces. Hoe we deze prestatie-maten in MATLAB bepalen, bespreken we in sectie 3.4 “Implementatie in MATLAB” op p.32.

De laatste belangrijke prestatie-maat is de *verlieskans*, i.e. wat is de kans dat de componenten die zich aandienen niet meer in de buffer kunnen omdat die al vol zit? In de literatuur wordt deze kans bepaald door de verhouding tussen het aantal onderdelen die niet meer opgevangen kunnen worden omdat de buffer vol is en het totaal aantal componenten die toekomen per tijdseenheid. Deze prestatie-maat kan berekend worden voor beide buffers samen maar ook voor buffer 1 en 2 apart. Omdat we eenzelfde werklust bij beide buffers veronderstellen, i.e. de gemiddelde aankomstintensiteit over de productieve en onproductieve periode is dezelfde, zijn deze drie kansen gelijk aan elkaar. We bespreken het begrip werklust in het volgende hoofdstuk. Wat belangrijk is te onthouden is dat de verlieskans voor de buffers apart en te samen dezelfde is en dat voor het basismodel de werklust gelijk is aan de aankomstintensiteit doordat de productieperiode altijd actief is (infra, p.48).

We zullen de prestatie-maten voor alle drie modellen bepalen en hun uitkomsten vergelijken. In het volgende deel bespreken we de methodologie en de gevolgde procedures in MATLAB om de snelheid van ons programma alsmaar te verbeteren.

3.3 Methodologie in MATLAB

In het vorige deel hebben we de opeenvolgende stappen besproken om de gewenste waarschijnlijkheidsvector van de modellen te berekenen. Echter, naarmate de individuele onderdelenbuffers groter worden, wordt het bijna onmogelijk om deze maten handmatig te becijferen. Hiervoor gebruiken we het softwarepakket MATLAB 7.1..

Eerst komt de methode waarop we ons programma gebaseerd hebben, de sparse-methode, aan bod. In sectie 3.3.2 beschrijven we waarom we voor de GMRES-methode (Generalized Minimum Residual) gekozen hebben om de stationaire waarschijnlijkheidsvector te berekenen. De belangrijkste reden is dat deze iteratieve methode lineaire vergelijkingen voor sparse-systemen veel sneller oplost dan andere directe methodes. Als laatste, bespreken we de verschillende procedures die we implementeerden om het programma alsmaar beter te maken.

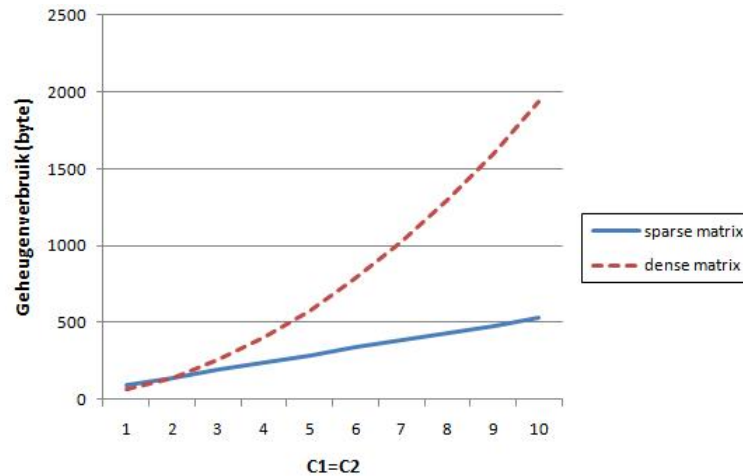
3.3.1 Sparse-methode

De numerieke berekeningen die we in MATLAB uitvoeren, zijn gebaseerd op de sparse-methode. Dit komt doordat de generatormatrix een sparse-structuur heeft. Een sparse-matrix is een matrix die hoofdzakelijk uit nul-elementen bestaat. Het gebruiken van sparse-matrices en daarbij horende gespecialiseerde algoritmes, leidt tot minder geheugenverbruik en procestijd, in vergelijking met standaard algoritmes. De reden hiervoor is dat de complexiteit (i.e. het aantal bewerkingen) kleiner is.

Als we bijvoorbeeld twee $N \times N$ matrices met elkaar willen vermenigvuldigen, dan blijkt volgens het Coppersmith-Winogradalgoritme dat twee dense matrices op $O(N^{2,376})$ tijd vermenigvuldigd worden terwijl de computertijd

gelijk is aan $O(N)$ voor sparse-matrices. De computationele complexiteit hangt lineair af van de rij (-en kolom-grootte) maar is onafhankelijk van het product van beide. We hebben deze informatie teruggevonden in de help-functie van MATLAB.

Qua opslag van gegevens blijkt het verbruik aan geheugen ook beperkter te zijn bij sparse- dan bij dense matrices (zie figuur 3.8). In MATLAB wordt de functie **sparse** gedefinieerd als $S = \text{sparse}(i, j, s)$ waarbij i de rijposities van de niet-nulelementen weergeeft, j vertegenwoordigt de kolomposities en s is gelijk aan de waarden die in de matrix worden geplaatst. Voor de twee eerste parameters i en j beschouwen we een computergeheugen van twee bytes. Beide zijn gehele waarden. De laatste parameter kan decimaal zijn zodat we zijn geheugenverbruik gelijkstellen aan vier bytes. In totaal gebruiken we acht bytes aan geheugen per element opgeslagen in de sparse-matrix. Voor de $N \times N$ generatormatrix van ons basismodel stellen we het totaal aantal niet-nulelementen gelijk aan aN waarbij a gelijk is aan het gemiddelde aantal transities per rij en N het aantal mogelijke toestanden. We stellen a gelijk aan drie (het aankomstproces van component 1 en 2 en het kitverwerkingsproces) en N gelijk aan $(C_1 + 1) \cdot (C_2 + 1)$. Het verbruikte geheugen voor het bewaren van deze matrix is gelijk aan $8 \cdot a \cdot N$. Voor de dense matrix moeten we in totaal N^2 elementen opslaan. Rekenen we opnieuw vier bytes per element, dan is het totale geheugenverbruik gelijk aan $4 \cdot N^2$ bytes. Uit figuur 3.8 leiden we af dat bij een capaciteit kleiner dan één, het opslaan van dense matrices minder geheugen vraagt dan voor sparse-matrices. Zodra de capaciteit groter is dan twee, geldt het omgekeerde en vergroot het verschil in verbruik naarmate de capaciteit groter wordt.



Figuur 3.8: Vergelijking in geheugenverbruik tussen de dense en sparse-matrix

Naast minder geheugenverbruik en procestijd kunnen we de generatormatrix gemakkelijker opstellen door het Kronecker-product te gebruiken (zie sectie 3.4). In het volgende deel bespreken we hoe we de stationaire waarschijnlijkheidsvector in MATLAB bepaald hebben.

3.3.2 GMRES-methode

Methode

We willen de evenwichtsvergelijking $\pi \cdot Q = \mathbf{b}$ oplossen waarbij π de onbekende is (zie vergelijking (3.9)). Ter herinnering \mathbf{b} is een rijvector met als laatste element één en met de andere elementen gelijk aan nul. Om het stelsel van lineaire vergelijking op te lossen kunnen we \mathbf{b} met de inverse van Q vermenigvuldigen. Hoewel de functie **inv**

in MATLAB de meest eenvoudige manier is om een inverse te bepalen, is deze niet efficiënt. Deze methode behoort tot de directe groep. Een nauwkeuriger directe manier om de inverse te bepalen is de vergelijking $\pi = \mathbf{b} \setminus \mathbf{Q}$ te schrijven. Echter, voor grote sparse-matrices is deze methode traag en soms zelfs onbruikbaar wegens onvoldoende geheugen. We zullen daarom gebruik maken van een andere familie van algoritmes: de indirecte of iteratieve groep. In MATLAB behoort de functie **gmres** tot deze soort algoritmes die specifiek op sparse-matrices wordt toegepast. De **gmres** functie poogt de lineaire vergelijking $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}$ voor \mathbf{x} op te lossen. \mathbf{A} is een $(N \times N)$ sparse-matrix en \mathbf{b} is een kolomvector met dezelfde kolomlengte als \mathbf{A} . We transponeren onze oorspronkelijke vergelijking $\pi \cdot \mathbf{Q} = \mathbf{b}$ zodat $\mathbf{Q} \cdot \pi = \mathbf{b}$ waarbij $\mathbf{Q} = \mathbf{Q}'$, $\mathbf{b} = \mathbf{b}'$ en $\pi = \pi'$. Het symbool $'$ betekent de getransponeerde van een vector. De **gmres** functie geeft π' weer zodat de oplossing nog getransponeerd moet worden.

De functie wordt in MATLAB gedefinieerd als: $x = \mathbf{gmres}(A, b, restart, tol, maxit)$. De eerste en tweede parameter zijn respectievelijk gelijk aan de generatormatrix \mathbf{Q} en de vector \mathbf{b} . Om de **gmres** functie te kunnen toepassen moet ook een stopcriterium bepaald worden. Zodra het residu, i.e. het verschil tussen de exacte en gevonden oplossing, onder een bepaalde waarde komt, dan stopt het algoritme met herhalen. De standaardwaarde (default value) van de parameter tol is gelijk aan $1e - 6$. We zullen deze gelijk stellen aan $1e - 7$ om nauwkeurige numerieke voorbeelden te bekomen. Naast A , b en tol hebben we ook de $restart$ parameter die het aantal herhalingen weergeeft vanaf dewelke het algoritme opnieuw start. De laatste parameter, $maxit$, bepaalt het maximaal aantal uitwendige (outer) iteraties. Om het maximaal aantal iteraties te kennen moeten we de twee laatste parameters met elkaar vermenigvuldigen. Praktisch bleken $restart$ en $maxit$ gelijk aan 120 voldoende.

Hoewel de **gmres** functie nauwkeurig en snel is, is de uitkomst slechts een benadering van de exacte oplossing. We bespreken hoe we de nauwkeurigheid van onze oplossing kunnen bepalen en verbeteren.

Nauwkeurigheid

De nauwkeurigheid van een oplossing kan bepaald worden met behulp van de **norm** functie. Als x_i het resultaat is van iteratie i , kunnen we de fout van het relatieve residu meten door $\mathbf{norm}(\mathbf{b} - \mathbf{A} \cdot \mathbf{x}_i) / \mathbf{norm}(\mathbf{b})$ te berekenen. Hoe dichter het relatieve residu van de waarde nul, hoe nauwkeurig je resultaat. De **norm** functie geeft een mate weer voor de omvang van de elementen in de vector en is gelijk aan de vierkantswortel van het kwadraat van elke element in de vector. Door te normeren is het residu onafhankelijk van de grootte van de vector en wordt dus relatief. Bij een relatief residu gelijk aan nul is de oplossing exact. Is het algoritme stabiel, i.e. het convergeert, dan vermindert het residu na elke iteratie.

Naast de nauwkeurigheid te meten kunnen we die ook verbeteren door de stationaire waarschijnlijkheidsvector π met de matrix \mathbf{Q}^d te vermenigvuldigen,

$$\mathbf{Q}_d = \frac{1}{\theta} \cdot \mathbf{Q} + \mathbf{I}. \quad (3.10)$$

Deze matrix wordt bekomen met behulp van uniformisatie. \mathbf{Q}_d is de transitie-matrix van een discrete-tijd Markov-keten met dezelfde evenwichtsvector π en met $\theta = \max_i |q_{ii}|$ (i.e. de grootste absolute diagonaalwaarde van de matrix \mathbf{Q}). Uit de Markov-theorie weten we dat:

$$\pi_k = \lim_{k \rightarrow \infty} [\pi_0 \cdot \mathbf{Q}_d^k] = \pi$$

waarbij π_0 een willekeurige beginvector is. We kunnen de nauwkeurigheid bijgevolg verbeteren door π_k herhaald met \mathbf{Q}_d te vermenigvuldigen. De oplossing wordt exacter naarmate $|\pi_k - \pi_k \cdot \mathbf{Q}_d|$ naar nul gaat.

3.3.3 Optimalisatie procedures

In dit deel beschrijven we de verbeteringprocedures voor het opstellen van de kittingmodellen. Vooruitgang werd vooral bij het ontwerpen van de stationaire waarschijnlijkheidsvector geboekt. Het plaatsen van de één-kolomvector in de generatormatrix en het bepalen van een nauwkeuriger initiële stationaire waarschijnlijkheidsvector waren de voornaamste.

We willen de verstreken tijd (elapsed time) waarin de verschillende stappen gemaakt worden versnellen. Om te weten welke functie het traagste is, meten we tijdsverschillen met behulp van de **tic** en **toc** functie. De functie **tic** start de tijdsmeting en de functie **toc** stopt het. Voor het basismodel blijkt het dat ongeveer 8% van de tijd naar het berekenen van de generatormatrix gaat, 83% naar de stationaire waarschijnlijkheidsvector en 9% naar de prestatiematen. In de bijlage A.2. “Verstreken tijd” wordt de afgelopen tijd (voor een eenmalige berekening en voor de numerieke voorbeelden) voor de drie modellen gegeven.

Voor deze modellen duurt het bepalen van de stationaire waarschijnlijkheidsvector het langst. We hebben daarom voornamelijk naar oplossingen gezocht om berekeningen voor deze vector te versnellen. Eerst werden deze erg versneld door de formule voor het plaatsen van een één-kolomvector in de generatormatrix te vereenvoudigen. Daarnaast hebben we een initiële vector bepaald om onze lineaire vergelijking op te lossen. We bespreken de gevolgde procedure.

Keuze van de initiële vector

Bij de **gmres** functie geldt: wanneer we de stationaire waarschijnlijkheidsvector berekenen dan is de standaardwaarde (default value) een nulvector. De eerste verbetering was om de initiële vector als equiprobable te beschouwen i.e. elke toestand heeft evenveel kans om voor te komen. Hoewel deze veronderstelling niet correct is versnelde dit de berekening. Dit komt doordat de som van de toestandswaarschijnlijkheden gelijk is aan één. Wanneer we de prestatiematen eenmalig berekenen is het moeilijk een betere initiële vector dan deze te bepalen.

Echter, wanneer we een plot maken waarin de capaciteit van de buffers varieert, dan zouden we de vorige berekende waarschijnlijkheidsvector kunnen gebruiken. Mits aanpassing van de initiële vector zou deze nauwkeuriger moeten zijn dan een equiprobabele vector. De reden is dat wanneer de capaciteit van de buffers in beperkte mate verandert, de kans groot is dat de toestandswaarschijnlijkheden dezelfde is of in de buurt van deze waarde blijft. Echter, het bepalen van deze vector is tijdrovend, om twee redenen. De stationaire waarschijnlijkheidsvector vergroot naarmate de capaciteitsbeperkingen C_1 en C_2 groter worden. Er moet dus een grotere initiële vector opgesteld worden dan de berekende stationaire waarschijnlijkheidsvector. Ten tweede, als C_1 groter wordt, heeft dit een ander effect op de bepaling van de initiële vector dan wanneer buffer 2 groter wordt (C_2). Beide vergroten de stationaire waarschijnlijkheidsvector maar niet op dezelfde plaats. We verduidelijken dit met een voorbeeld.

Gegeven dat $C_1 = 2$ en $C_2 = 2$. Indien de capaciteitsbeperking C_2 met één eenheid groter wordt dan gaan we van een stationaire waarschijnlijkheidsvector met de toestanden:

$$\left[(0,0) \quad (0,1) \quad (0,2) \quad (1,0) \quad (1,1) \quad (1,2) \quad (2,0) \quad (2,1) \quad (2,2) \right]$$

naar de vector:

$$\left[(0,0) \quad (0,1) \quad (0,2) \quad (0,3) \quad (1,0) \quad (1,1) \quad (1,2) \quad (1,3) \quad (2,0) \quad (2,1) \quad (2,2) \quad (2,3) \right].$$

Nu gegeven dat $C_2 = 2$ en $C_1 = 3$ is de vector gelijk aan:

$$\left[\begin{array}{cccccccccccc} (0,0) & (0,1) & (0,2) & (1,0) & (1,1) & (1,2) & (2,0) & (2,1) & (2,2) & \mathbf{(3,0)} & \mathbf{(3,1)} & \mathbf{(3,2)} \end{array} \right].$$

Om de initiële vector op te stellen maken we een matrix aan waarin we in iedere kolom de berekende vector gaan bewaren. Doordat de elementen van deze matrix hoofdzakelijk gelijk zijn aan nul bepalen we de matrix als sparse, wat de berekeningen doet versnellen. We laten eerst de waarde C_1 variëren en daarna C_2 . Het omgekeerde hebben we ook geprobeerd. Nadat we de vector berekend hebben, beschouwen we deze als een initiële waarde voor de volgende lus. Het probleem van deze aanpak was dat we bijvoorbeeld moesten overgaan van $C_1 = 20$ en $C_2 = 100$ naar $C_1 = 30$ en $C_2 = 10$. De verschillen tussen de twee vectoren waren te groot. In plaats van de vorige vector hebben we daarom de vector die het meest op de nieuwe te berekenen vector leek, gebruikt. Dit was in ons voorbeeld gelijk aan $C_1 = 20$ en $C_2 = 10$. De uitkomst bleef toch teleurstellend. Het bleek langer te duren en de nauwkeurigheid van de waarschijnlijkheidsvector was niet verhoogd. De initiële vectoren waren afhankelijk van de manier waarop ze berekend werden. We hebben ons dus beperkt tot een equiprobable vector als initiële waarde voor veranderde capaciteit.

Echter, in plaats van de capaciteit kunnen we ook de werklust, i.e. de gemiddelde aankomstintensiteit over de actieve en inactieve periode, laten variëren. Het grote voordeel is dat de positie in de vector van iedere toestandswaarde niet verandert, de grootte van de vector is onafhankelijk van de waarde gegeven aan de werklust. Bovendien, zoals bij variërende capaciteit, is de kans groot dat de toestandswaarde ongeveer dezelfde blijft wanneer de werklust enigszins verandert. Qua snelheid was de uitkomst duidelijk beter dan bij variërende capaciteit. Voor het basismodel ging de tijd die nodig was om de twee grafieken in sectie 4.4.1 te schetsen ongeveer van 770 naar 72 seconden, een reductie met een factor 10.

3.4 Implementatie in MATLAB

In dit deel bespreken we eerst hoe we de generatormatrix hebben opgesteld. Daarna vatten we de berekeningen in MATLAB voor iedere prestatiemaat kort samen. De verschillende stappen worden voor het basismodel uitgelegd. Deze gelden ook voor het 1IPP- maar niet voor het 2IPP-model. De inhoud van de generatormatrix moet op een andere manier opgesteld worden. We maken gebruik van twee for-lussen (zie bijlage A.4. “De generatormatrix in MATLAB voor het 2IPP-model”).

3.4.1 Generatormatrix

Zoals in sectie 3.3.1 vermeld, willen we onze generatormatrix sparse maken door ons enkel op de niet-nulintensiteiten te concentreren. Om dit te kunnen verwezenlijken stellen we onze matrix niet op aan de hand van een vertrekkende toestand door gebruik te maken van evenwichtsvergelijkingen maar op basis van de intensiteit. In het toestandsdiagram betekent dit dat we naar alle pijlen van de gekozen intensiteit kijken en onthouden vanuit welke toestand ze komen en naar welke toestand ze gaan. De veranderingen in toestand voor de drie intensiteiten zijn:

$$\lambda_1 : i = (X_1, X_2) \rightarrow j = (X_1 + 1, X_2) ; X_1 = 0, 1, \dots, C_1 - 1 ; X_2 = 0, 1, \dots, C_2,$$

$$\lambda_2 : i = (X_1, X_2) \rightarrow j = (X_1, X_2 + 1) ; X_1 = 0, 1, \dots, C_1 ; X_2 = 0, 1, \dots, C_2 - 1,$$

$$\mu : i = (X_1, X_2) \rightarrow j = (X_1 - 1, X_2 - 1) ; X_1 = 1, 2, \dots, C_1 ; X_2 = 1, 2, \dots, C_2.$$

De waarden i en j zijn respectievelijk de huidige en toekomstige toestand en worden bepaald door het aantal componenten in buffer 1 en 2, met name X_1 en X_2 .

Om de generatormatrix te bepalen stellen we eerst een matrix op waarbij de drie parameters i , j en s bepaald worden. We kiezen om de matrix op te stellen volgens de volgorde λ_1 , λ_2 en μ . Voor $\lambda_1 = 1$ en $C_1 = 3$ en $C_2 = 2$ is de matrix:

$$Q = \begin{bmatrix} 0 & 0 & | & 1 & 0 & | & 1 \\ 0 & 1 & | & 1 & 1 & | & 1 \\ 0 & 2 & | & 1 & 2 & | & 1 \\ 1 & 0 & | & 2 & 0 & | & 1 \\ 1 & 1 & | & 2 & 1 & | & 1 \\ 1 & 2 & | & 2 & 2 & | & 1 \\ 2 & 0 & | & 3 & 0 & | & 1 \\ 2 & 1 & | & 3 & 1 & | & 1 \\ 2 & 2 & | & 3 & 2 & | & 1 \end{bmatrix}$$

De eerste twee kolommen zijn gelijk aan de huidige toestand i , de twee volgende aan de nieuwe toestand j en de laatste aan de intensiteit λ_1 . Een instrument in MATLAB om deze matrix sneller te schrijven is het *Kronecker-product*. Aangeduid met het symbool \otimes , het is een nieuwe matrix die ontstaat door elk element van de eerste matrix te vervangen door het product van dat element met de tweede matrix. Deze product-methode wordt in MATLAB gesteund door de functie **kron**(X, Y) waarbij X en Y respectievelijk de eerste en tweede matrix zijn. In ons voorbeeld herhaalt de eerste kolom van de matrix Q drie keren de waarden nul, één en twee. De formule in MATLAB voor deze kolom in Q is $[kron((0:2)', ones(3,1))]$ en is in de wiskunde gelijk aan:

$$\begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \end{bmatrix}$$

Om volledig te zijn moet de matrix Q nog aangevuld worden met de toestandsveranderingen verbonden aan de intensiteiten λ_2 en μ . Nadat deze bepaald zijn, zetten we de verschillende delen onder elkaar in een matrix.

Echter, deze matrix laat ons nog steeds niet toe een sparse-matrix op te stellen aangezien de toestand door twee waarden bepaald wordt: het aantal componenten van type 1 en type 2. Als tweede stap geven we een unieke nummering aan elke toestand. Dit getal zal de rij- en kolompositie in de standaardmatrix weergeven. Voor het basismodel is de formule: $X_1 \cdot (C_2 + 1) + X_2 + 1$ zodat bijvoorbeeld de waarde verbonden aan een huidige toestand $(0,0)$ gelijk is aan $[0 \cdot (2 + 1)] + 0 + 1 = 1$ gaande naar een nieuwe toestand $(1,0)$ is: $[1 \cdot (2 + 1)] + 0 + 1 = 4$. Deze toestandsverandering wordt veroorzaakt door een aankomstproces van component 1 zodat de intensiteit λ_1 is. Indien deze gelijk is aan één dan is de sparse-versie voor dit voorbeeld gelijk aan: $(1,4)1$. We doen dit voor alle toestandsveranderingen en definiëren de matrix als sparse.

De sparse generatormatrix is nu quasi volledig bepaald. Enkel de diagonaalelementen moeten nog berekend worden. Als laatste stap stellen we deze gelijk aan de negatieve som van alle intensiteiten van de overeenkomstige rij (supra, p.16).

We bekomen tenslotte de gewenste sparse-generatormatrix. Hoewel de formulering voor sparse-matrices ingewikkelder lijkt, gaat het schrijfwerk naarmate de matrix groter wordt sneller dan bij een standaardmatrix doordat we enkel de niet-nulwaarden bepalen. Bovendien kunnen deze waarden systematisch weergegeven worden dankzij het Kronecker-product. Daarnaast biedt MATLAB de mogelijkheid om lineaire vergelijkingen van sparse-matrices sneller op te lossen. We gebruiken de functie **gmres** om de stationaire waarschijnlijkheidsvector te bepalen. Deze functie houdt rekening met de sparse-structuur van de generatormatrix en staat uitgelegd in sectie 3.3.2. We beschrijven nu de procedure gevolgd om de prestatiematen te berekenen. De toegepaste formules gelden voor de drie modellen.

3.4.2 Prestatiematen

Om de prestatiematen sneller te kunnen berekenen, hebben we alle opgestelde vectoren en matrices sparse gemaakt.

Gemiddelde bezetting in de buffer

We willen de *gemiddelde bezetting* kennen voor buffer 1 en 2. De stappen nodig om de gemiddelde buffergrootte voor component 1 te berekenen, zijn dezelfde als voor component 2. De berekening maakt gebruik van het begrip verwachtingswaarde. De formule om deze voor buffer 1 te bepalen is:

$$E[X_1] = \langle \mathbf{i}, \mathbf{q}_1 \rangle \quad (3.11)$$

waarbij \mathbf{i} een vector is die waarden aanneemt gaande van nul tot C_1 en \mathbf{q}_1 de *kansmassafunctie* (probability mass function) van de bufferbezetting voor buffer 1 is. Stel dat q_1 de kansmassafunctie is voor buffer 1 gegeven dat $C_1 = C_2 = 2$. Dit geeft:

$$\mathbf{q}_1 = \left[\sum_{i=0}^2 \pi_{0,i}, \sum_{i=0}^2 \pi_{1,i}, \sum_{i=0}^2 \pi_{2,i} \right] \quad (3.12)$$

Om het gemiddelde te bepalen hebben we dus eerst alle waarden nodig die aangeven wat de kans is dat buffer 1 een specifieke waarde aanneemt binnen zijn capaciteitsbeperking. We vragen ons dus af wat de kans is dat het aantal componenten in buffer 1 gelijk is aan nul en zo gaande tot de waarde C_1 . Het resultaat is de kansmassafunctie voor buffer 1. In MATLAB, bekomen we deze door de stationaire waarschijnlijkheidsvector $\boldsymbol{\pi}$ te vermenigvuldigen met een opgestelde matrix. De eigenschappen van deze matrix zijn dat:

- het totaal aantal kolommen gelijk is aan $(C_1 + 1)$,
- het totaal aantal rijen gelijk is aan $(C_1 + 1) \cdot (C_2 + 1)$,
- het aantal 1'en in elke kolom gelijk is aan het aantal $C_2 + 1$, de andere waarden zijn gelijk aan nul.

In ons voorbeeld is de matrix gelijk aan:

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

In deze matrix vertegenwoordigt elke kolom een aantal X_1 in buffer 1 en elke rij een toestand (X_1, X_2) . We plaatsen een één wanneer de toestand hetzelfde aantal in buffer 1 heeft als deze in de kolom. In ons voorbeeld heeft de matrix \mathbf{A}_1 drie kolommen waarbij elke kolom het aantal 0, 1 en 2 in buffer 1 voorstelt. Voor de eerste kolom waar $X_1 = 0$, zijn de waarden één geplaatst voor de toestanden $(0, 0)$, $(0, 1)$ en $(0, 2)$. Het product van $\boldsymbol{\pi}$ met \mathbf{A}_1 geeft als resultaat de kansmassafunctie \mathbf{q}_1 . Aan de hand van vergelijking (3.11) bekomen we de gemiddelde bezetting in buffer 1.

Voor buffer 2 geldt dezelfde procedure. De matrix \mathbf{A}_2 zal wel andere eigenschappen hebben. Die zijn dat:

- het totaal aantal kolommen gelijk is aan $(C_2 + 1)$,
- het totaal aantal rijen gelijk is aan $(C_1 + 1) \cdot (C_2 + 1)$,
- het aantal 1'en in elke kolom gelijk is aan $C_1 + 1$, de andere waarden zijn gelijk aan nul.

Dezelfde stationaire waarschijnlijkheidsvector is van toepassing en de matrix is gelijk aan:

$$\mathbf{A}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Variantie

Naast het bepalen van de verwachtingswaarde, is het ook belangrijk de *variantie* te kennen om te weten in welke mate deze waarde varieert. Wanneer we de verwachtingswaarde hebben dan kan deze prestatie maat zeer snel bepaald worden. De formule die we in MATLAB zullen toepassen is:

$$Var[X] = E(X^2) - (E(X))^2 \quad (3.13)$$

waarbij $E(X^2)$ voor buffer 1 bepaald wordt aan de hand van de formule $\langle \mathbf{q}_1 \cdot \mathbf{i}^2 \rangle$.

Kans dat één van beide buffers leeg is

De volgende prestatiemaat is de *kans dat één van beide buffers leeg is*. Om deze kans te bepalen vermenigvuldigen we een opgestelde vector \mathbf{a}_1 met de stationaire waarschijnlijkheidsvector. De formule is:

$$K_1 = \mathbf{a}_1' \cdot \boldsymbol{\pi} \quad (3.14)$$

De eigenschappen van de vector \mathbf{a}_1 zijn dat:

- het totaal aantal kolommen gelijk is aan 1,
- het totaal aantal rijen gelijk is aan $(C_1 + 1) \cdot (C_2 + 1)$,
- het aantal 1'en in elke kolom gelijk is aan $C_2 + C_1 + 1$, de andere waarden zijn gelijk aan nul.

In het voorbeeld waarbij $C_1 = 3$ en $C_2 = 2$ moet deze vector gelijk zijn aan:

$$\mathbf{a}_1' = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.15)$$

Elke kolom vertegenwoordigt een toestand gaande van toestand $(0,0)$ tot toestand $(3,2)$. Het getal 1 wordt geplaatst voor de toestanden $(0,0)$, $(0,1)$, $(0,2)$, $(1,0)$, $(2,0)$ en $(3,0)$. Door \mathbf{a}_1' met $\boldsymbol{\pi}$ te vermenigvuldigen geven we de kansen aan waarvoor één of beide buffers leeg zijn.

Kans dat de buffer vol is

Een vierde belangrijke prestatiemaat is de *kans dat een buffer vol is*. Om de kans dat buffer 1 vol is te bepalen volgen we dezelfde redenering dan voor de kans dat één van beide buffers leeg is. We bepalen een kolomvector \mathbf{a}_2 met de volgende eigenschappen:

- het totaal aantal rijen is gelijk aan $(C_1 + 1) \cdot (C_2 + 1)$,
- het totaal aantal kolommen is gelijk aan 1,
- het totaal aantal 1'en is gelijk aan $C_2 + 1$.

De kolomvector voor $C_1 = 3$ en $C_2 = 2$ is gelijk aan:

$$\mathbf{a}_2' = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad (3.16)$$

De kolomvector krijgt de waarde 1 voor de toestanden $(3,0)$, $(3,1)$ en $(3,2)$. De formule om de kans dat buffer 1 vol is te bepalen is gelijk aan:

$$K_2 = \mathbf{a}_2' \cdot \boldsymbol{\pi} \quad (3.17)$$

Voor de kans dat buffer 2 vol is, gelden dezelfde eigenschappen als bij vector \mathbf{a}_2 behalve dat het totaal aantal 1'en gelijk is aan $C_1 + 1$ in plaats van $C_2 + 1$. Het totaal aantal toestanden waarbij $X_2 = C_2$ is gelijk aan $C_1 + 1$. Voor hetzelfde voorbeeld is de kans dat buffer 2 vol is gelijk aan:

$$K_3 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}' \cdot \boldsymbol{\pi} \quad (3.18)$$

De kolomvector krijgt de waarde één voor de toestanden $(0, 2)$, $(1, 2)$, $(2, 2)$ en $(3, 2)$.

Verlieskans voor de buffers

Ten laatste de *verlieskans* **PLR** (pakket loss ratio). We stellen deze gelijk aan:

$$\mathbf{PLR} = \frac{\text{verlies}}{\lambda_1 + \lambda_2} \quad (3.19)$$

waarbij het verlies aan componenten voor beide buffers bepaald wordt door de formule $\lambda_1 + \lambda_2 - 2 \cdot \text{TP}$. Deze formule is gebaseerd op het idee dat alles wat er in het proces binnenkomt er ook uitgaat. De doorvoer (throughput) TP is het aantal kits per tijdseenheid die door het kittingproces gemaakt wordt. Deze is gelijk aan:

$$TP = \frac{Kans_{bezet}}{\frac{1}{\mu}} = \mu \cdot Kans_{bezet} \quad (3.20)$$

waarbij $Kans_{bezet}$ de kans is dat het kitwerkstation bezig is. Deze kans is gelijk aan de kans dat geen van beide buffers leeg is, i.e. $1 - K_1$. Om de doorvoer te kennen moeten we dus dit getal delen door de gemiddelde tijd die nodig is om een kit te maken. We hebben deze gegevens zodat de verlieskans zeer gemakkelijk te berekenen is in MATLAB.

In de volgende sectie bespreken we de numerieke voorbeelden die we in MATLAB gegenereerd hebben.

3.5 Numerieke voorbeelden

Ingangparameters

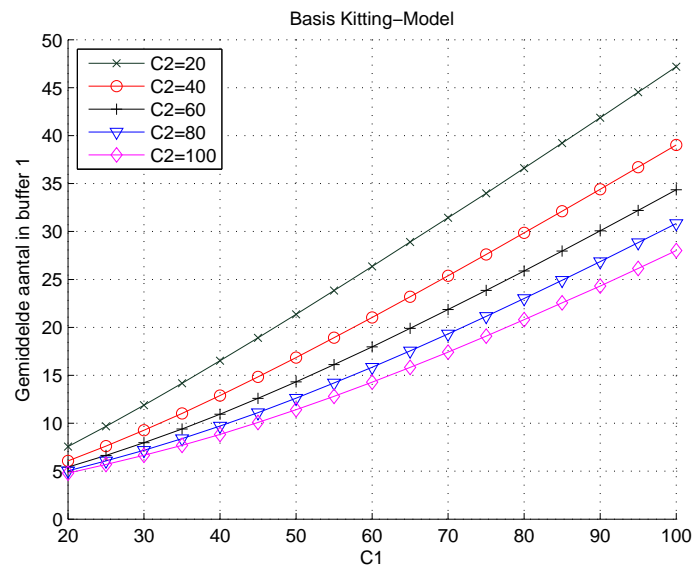
Voor het basismodel gebruiken we de parameters uit tabel 3.1. In een tijdinterval van lengte één komen gemiddeld 0,8 onderdelen toe in beide buffers en één kit wordt verwerkt. De aankomstintensiteit is in het basismodel gelijk aan de werklust; we veronderstellen bijgevolg een 80% werklust aan beide bronnen. Het begrip werklust staat uitgelegd in sectie 4.2. Alle numerieke voorbeelden bij variërende capaciteit worden met dezelfde werklust gemodelleerd.

Basismodel	
Aankomstintensiteit in buffer 1	$\lambda_1 = 0,8$
Aankomstintensiteit in buffer 2	$\lambda_2 = 0,8$
Verwerkingsintensiteit	$\mu = 1$

Tabel 3.1: Ingangparameters voor het basismodel

Grafieken

Gemiddelde bezetting van buffer 1

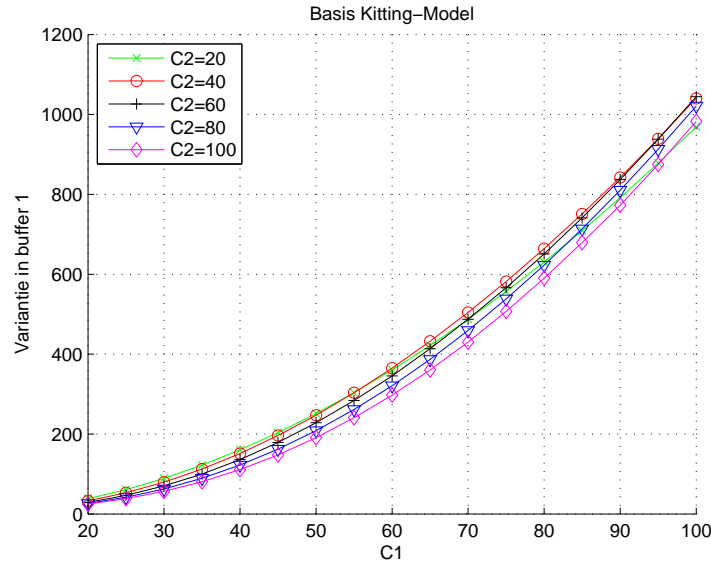


Figuur 3.9: Basismodel: Gemiddelde bezetting van buffer 1 bij variërende C_1

In figuur 3.9 zetten we de gemiddelde bezetting uit van buffer 1 waarbij de capaciteit varieert. Gegeven C_2 , hoe groter C_1 , hoe groter de gemiddelde bezetting. Dit betekent dat de intensiteit waarmee kits verwerkt worden niet voldoende hoog is ten opzichte van de aankomstintensiteit van de onderdelen.

We stellen ook vast dat het gemiddelde stijgt naarmate C_2 kleiner is omdat buffer 2 naast vaker vol ook vaker leeg zal zijn. Het zal bijgevolg vaker voorkomen dat de kitting niet kan doorgaan, wat leidt tot een stijging in buffer 1.

Variantie in de bezetting van buffer 1



Figuur 3.10: Basismodel: Variantie in de bezetting van buffer1 bij variërende C_1

Figuur 3.10 toont de variantie van de bezetting van buffer 1. Deze stijgt naarmate C_1 groter wordt. Dit betekent dat wanneer het aantal X_1 meer kan variëren, de variantie toeneemt.

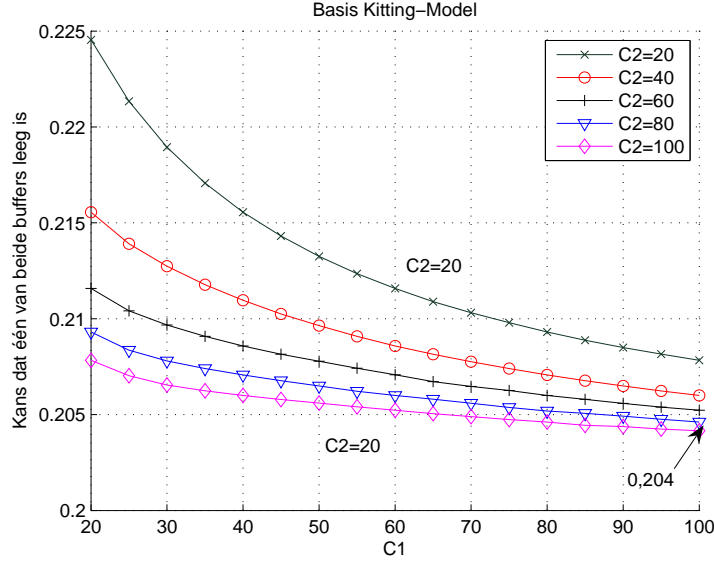
We merken ook op dat de variantie enigszins groter is bij een kleinere C_2 . We zien dat de kans dat buffer 1 vol is en de kans dat één van beide buffers leeg is groter zijn wanneer C_2 kleiner is, wat de trend van deze grafiek bevestigt. Wat wel frappant is aan deze grafiek, is dat de curven dicht tegen elkaar liggen, terwijl dit helemaal niet zo is voor de gemiddelde bezetting.

De grafieken voor deze twee prestatiematen gelden ook voor buffer 2. Omwille van de symmetrie hebben beide componenten dezelfde aankomst- en verwerkingsintensiteit.

Kans dat één van beide buffers leeg is

Figuur 3.11 toont de kans dat één van beide buffers leeg is bij variërende waarden C_1 . In deze grafiek daalt de kans naarmate C_1 en C_2 groter worden. Het is logisch dat bij een voldoende hoge aankomstintensiteit, de kans op een lege buffer zal verkleinen bij toenemende capaciteit.

Wanneer de capaciteit naar oneindig gaat dan streeft de stock-out-kans naar de waarde $1 - \frac{\lambda}{\mu} = 0,2$ in ons voorbeeld, waarbij de fractie $\frac{\lambda}{\mu}$ gelijk is aan de *belasting*. In dit geval hebben we geen verlies, i.e. alle componenten die toekomen gaan er uit als kits. Voor $C_1 = C_2 = 100$ convergeert de waarde naar 0,204. De reden voor deze kleine discrepantie is het verlies. We bespreken de relatie tussen de stock-out- en de verlieskans in het volgende hoofdstuk (infra, p.49).



Figuur 3.11: Basismodel: Kans dat één van beide buffers leeg is bij variërende C_1

Kans dat buffer 1 vol is

De kans dat buffer 1 vol is staat getekend in figuur 3.12. Hoe groter zijn capaciteit, hoe kleiner de kans dat deze vol zal zijn. De kans wordt ook kleiner naarmate C_2 groter wordt. Minder componenten gaan verloren als buffer 2 groter is zodat er meer kits kunnen verwerkt worden. Om die reden is de kans dat buffer 1 vol is kleiner. We bepalen de verlieskans voor buffer 1 en komen uit dat deze gelijk is aan de kans dat de buffer vol is.

Voor $C_1 = C_2 = 20$ is de kans dat de buffer 1 vol is gelijk aan 0,031. Op de vorige figuur is de stock-out-kans voor dezelfde capaciteitwaarden gelijk aan 0,225. Dit wil zeggen dat de doorvoer (wanneer $\mu = 1$), gelijk is aan $1 - 0,225 = 0,775$ in plaats van 0,8 wanneer er geen verlies is (supra, p.37). Het aantal componenten dat per tijdseenheid verloren gaat in de buffer is dus gelijk aan $0,8 - 0,775 = 0,025$. Omgerekend naar de **PLR** voor buffer 1 is deze gelijk aan:

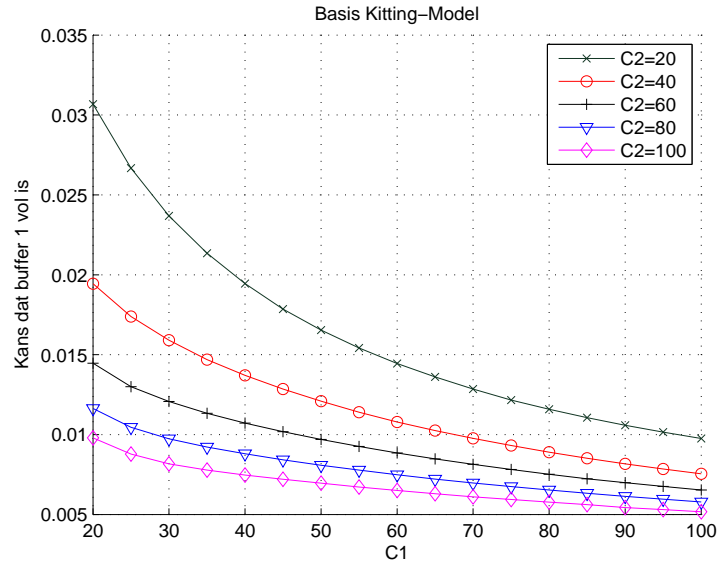
$$PLR_1 = \frac{0,025}{\lambda_1} = 0,3125 \quad (3.21)$$

De berekende verlieskans is bijgevolg gelijk aan de kans dat de buffer vol is. Deze prestatiemaat berekent de verlieskans voor buffer 1 maar dit kan ook voor component 2 apart en voor de buffers samen. We stellen deze parameters respectievelijk gelijk aan **PLR₂** en **PLR**. Om deze laatste te bepalen passen we de vergelijking (3.19) toe. Doordat de aankomstintensiteit λ_1 en λ_2 gelijk zijn aan elkaar geldt de vergelijking:

$$PLR = PLR_1 = PLR_2$$

Deze gelijkheid zal ook gelden voor de uitgebreide modellen omdat we eenzelfde werklast aan beide bronnen veronderstellen (supra, p.48).

Het feit dat de verlieskans en de kans dat de buffer vol is gelijk zijn aan elkaar is een gevolg van de PASTA-eigenschap bij het Poisson-proces (supra, p.12). Wanneer de buffer vol zit en er komt een onderdeel in het proces



Figuur 3.12: Basismodel: Kans dat buffer 1 vol is bij variërende C_1

toe dan zal deze component verloren gaan. Doordat de verdeling van het aantal onderdelen bij aankomst van een nieuw onderdeel hetzelfde is als de verdeling van het aantal onderdelen op een willekeurige tijdstip, zullen deze twee kansen gelijk zijn.

Hoofdstuk 4

Uitgebreide modellen

Tot nog toe veronderstelden we dat componenten volgens een Poisson-proces in de kittingbuffers aankwamen. Zo een model veronderstelt een gelijkmatige levering van componenten. Om de impact van interrupties in de levering te bestuderen, stellen we naast het basismodel twee andere modellen op: het 1IPP-model en het 2IPP-model. Het eerste model houdt rekening met mogelijke onderbrekingen in de vervaardiging van component 1 en het tweede model bij zowel component 1 als 2.

In sectie 4.1 beschrijven we het 1IPP- en 2IPP-model aan de hand van toestandsdiagrammen. Ten opzichte van het basismodel moeten er twee bijkomende parameters bepaald worden: α en β . In sectie 4.2 bespreken we de parameters voor beide modellen. We b eindigen dit hoofdstuk met enkele numerieke voorbeelden.

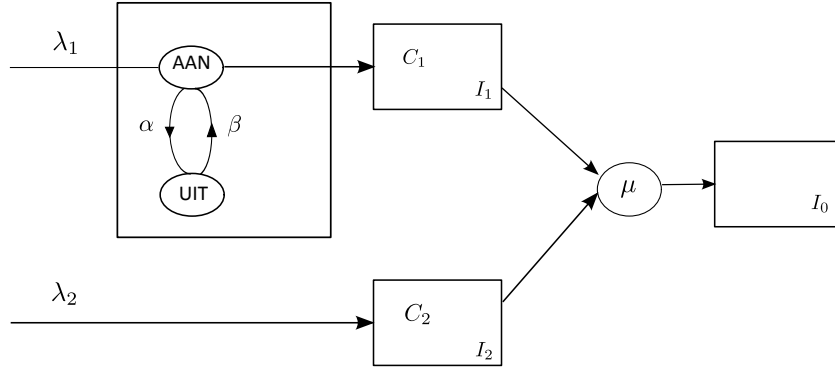
4.1 Beschrijving van de uitgebreide modellen

4.1.1 Eerste uitgebreid model voor kitting: het 1IPP-model

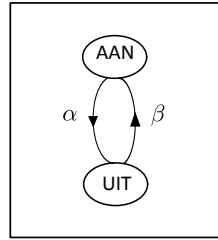
Figuur 4.1 geeft schematisch het 1IPP-model weer. Het verschil met het basismodel situeert zich bij het aankomst-proces van component 1. We modelleren dit proces niet langer als een Poisson-proces maar als een IPP (supra, p.13). Dit betekent dat we ervan uit gaan dat component 1 niet altijd kan vervaardigd worden, omdat bijvoorbeeld de machine een zwaar technisch probleem heeft (supra, p.8).

Het model kan ofwel in een *inactieve toestand* zitten waarbij component 1 niet geproduceerd wordt ofwel in een *actieve toestand* waarbij beide componenten wel in het proces aankomen (supra, p.13). Ter herinnering, voor een IPP met parameters α en β , de intensiteit gaande van een actieve naar een inactieve toestand is gelijk aan α en het omgekeerde is gelijk aan β . Figuur 4.2 schetst de overgang tussen de twee toestanden van het aankomstproces.

Deze twee aankomsttoestanden hebben als gevolg dat er in totaal twee keer meer mogelijke toestanden zijn dan in het basismodel i.e. $(C_1 + 1) \cdot (C_2 + 1) \cdot 2$. Bovendien wordt een toestand in het 1IPP-model door drie waarden bepaald: de aankomsttoestand i (i.e. of het proces zich al dan niet in een actieve toestand bevindt) en het aantal componenten in buffer 1 en 2, respectievelijk X_1 en X_2 . We zullen de inactieve en actieve toestand als een binaire getal voorstellen. Het laat ons toe het model te kunnen opstellen en analyseren in MATLAB. In tegenstelling tot het basismodel is het 1IPP-model bijgevolg niet gekenmerkt door een tweedimensionale maar door een driedimensionale toestand, i.e. het tripel (i, X_1, X_2) (supra, p.22).

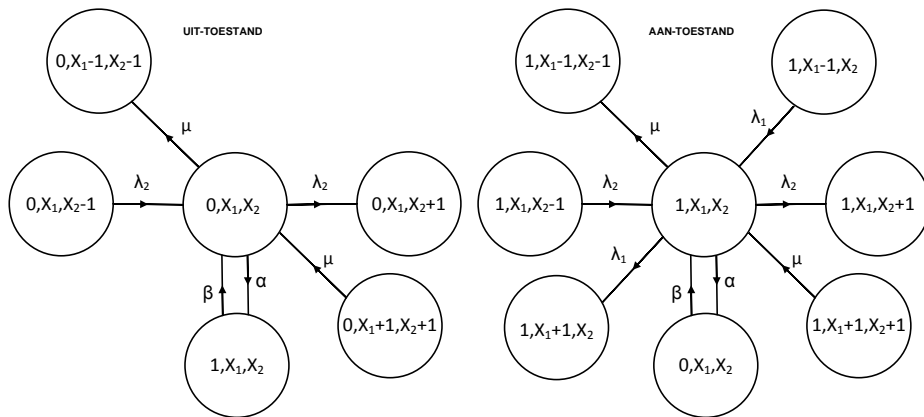


Figuur 4.1: Schema voor het 1IPP-model



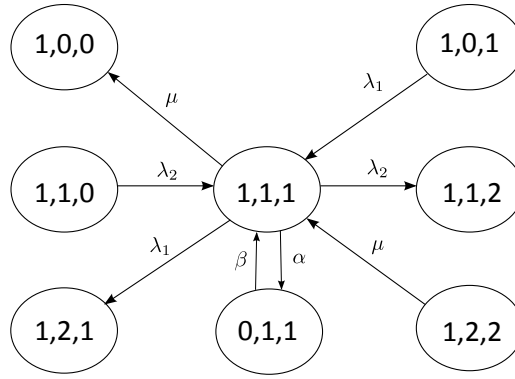
Figuur 4.2: Transitietoestanden voor het 1IPP-model

In figuur 4.3 tonen we een fragment van het toestandsdiagram voor het 1IPP-model. Merk op dat we een onderscheid maken tussen de toestanden waarin het aankomstproces niet actief is (links) en wel actief is (rechts). We observeren dat een toestand bijvoorbeeld niet van toestand $(0, X_1, X_2)$ naar toestand $(0, X_1 + 1, X_2)$ kan evolueren maar wel naar toestand $(0, X_1, X_2 + 1)$. Het gevolg van een inactieve toestand is immers dat X_1 niet kan veranderen. We veronderstellen ook dat de evolutie van toestand $(0, X_1, X_2)$ naar toestand $(1, X_1 + 1, X_2)$ in één keer niet mogelijk is. Eerst kan het model evolueren naar een actieve toestand en in een tweede fase kunnen componenten van type 1 in de buffer aankomen maar dat kan niet tegelijkertijd gebeuren.



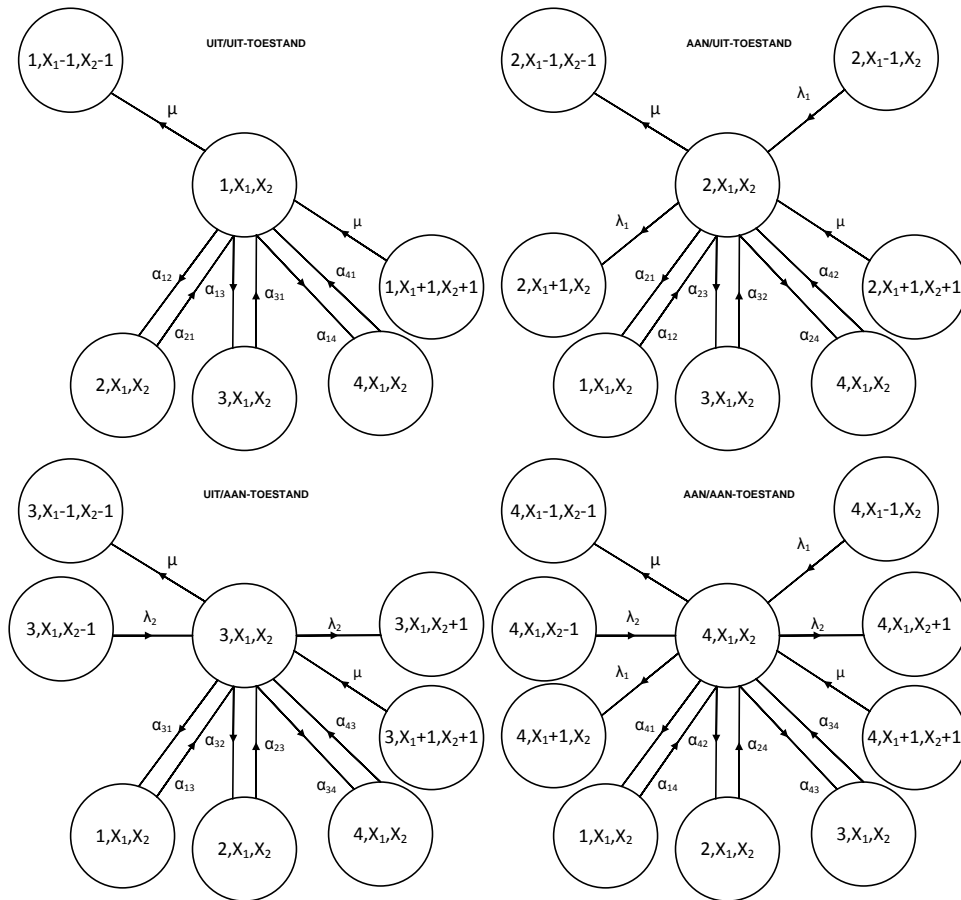
Figuur 4.3: Fragment van het toestandsdiagram voor het 1IPP-model

In figuur 4.4 schetsen we het toestandsdiagram van een kittingproces dat in een actieve toestand zit en waar elke buffer één component bevat.

Figuur 4.4: Toestandsdiagram voor toestand $(1,1,1)$

4.1.2 Tweede uitgebreid model voor kitting: het 2IPP-model

Voor dit model zijn tijdelijke productieonderbrekingen bij beide componenten mogelijk. Er zijn in totaal vier aankomsttoestanden: beide aankomstprocessen zitten in een inactieve (1) of een actieve toestand (4), ofwel zit buffer 1 in een actieve toestand en buffer 2 in een inactieve toestand (2) of omgekeerd (3).



Figuur 4.5: Fragment van het toestandsdiagram voor het 2IPP-model

In figuur 4.5 geven we vier toestandsdiagrammen weer voor het 2IPP-model. Elk toestandsdiagram is verbonden aan een aankomsttoestand. Het eerste toestandsdiagram (bovenaan links) bepaalt alle mogelijke toestandstransities als beide aankomstprocessen zich in een inactieve toestand bevinden. Het tweede diagram (bovenaan rechts) schetst aankomsttoestand 2, het derde aankomsttoestand 3 en het vierde aankomsttoestand 4.

Zoals in het 1IPP-model wordt een toestand bepaald door drie waarden, namelijk de aankomsttoestand en het aantal onderdelen in buffer 1 en 2. De toestand $(4, X_1, X_2)$ betekent dat beide aankomstprocessen in een actieve toestand zitten en dat het aantal componenten in buffer 1 en 2 gelijk is aan X_1 en X_2 . Dit keer hebben we vier en niet twee aankomsttoestanden, zodat er in totaal $(C_1 + 1) \cdot (C_2 + 1) \cdot 4$ mogelijke toestanden zijn.

Daarnaast zijn er in totaal twaalf mogelijke overgangsintensiteiten. We plaatsen deze in een generatormatrix \mathbf{A} waarbij de waarde $\alpha_{i,j}$ gelijk is aan de intensiteit om van toestand i naar toestand j te evolueren zodat:

$$\mathbf{A} = \begin{bmatrix} . & \alpha_{1,2} & \alpha_{1,3} & \alpha_{1,4} \\ \alpha_{2,1} & . & \alpha_{2,3} & \alpha_{2,4} \\ \alpha_{3,1} & \alpha_{3,2} & . & \alpha_{3,4} \\ \alpha_{4,1} & \alpha_{4,2} & \alpha_{4,3} & . \end{bmatrix}$$

De diagonaalwaarden voor de generatormatrix zijn gelijk aan de negatieve som van de intensiteiten van de overeenkomstige rij (supra, p.16).

4.2 Bepaling van de ingangparameters

Vooraleer we de parameters bepalen voor de twee IPP-modellen, specificeren we de generatormatrix \mathbf{A} voor het 2IPP-model. Deze matrix is in de vorige sectie op een algemene manier gedefinieerd waardoor beide bronnen mogelijk afhankelijk zijn van elkaar. We zullen echter bijkomend veronderstellen dat de aankomstprocessen onafhankelijk zijn van elkaar. Figuur 4.6 schetst het 2IPP-model met onafhankelijke bronnen. Het aankomstproces van component 2 is, zoals voor component 1, gemodelleerd als een IPP. Figuur 4.6 toont ons dat het aantal mogelijke overgangen beperkter is. We geven die weer in figuur 4.7.

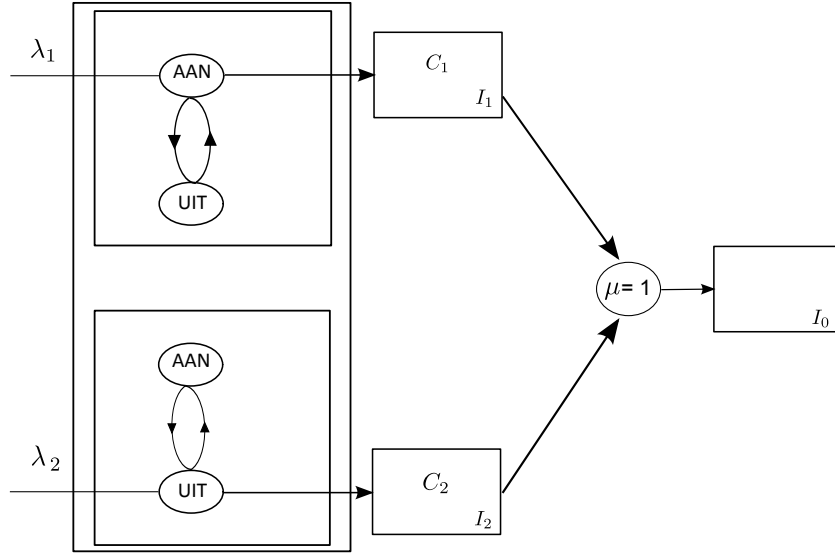
Voor het onafhankelijke geval herdefiniëren we de matrix als:

$$\mathbf{A} = \begin{bmatrix} . & \beta_1 & \beta_2 & 0 \\ \alpha_1 & . & 0 & \beta_2 \\ \alpha_2 & 0 & . & \beta_1 \\ 0 & \alpha_2 & \alpha_1 & . \end{bmatrix}$$

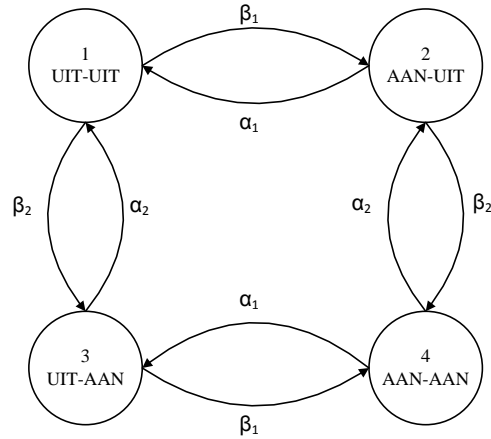
In deze situatie is het onmogelijk om bijvoorbeeld van aankomsttoestand 1 naar 4 in één keer te evolueren: beide aankomstprocessen kunnen niet tegelijkertijd van een inactieve naar een actieve toestand veranderen.

We kunnen nu de overgangsintensiteiten voor de twee IPP-modellen veralgemenen tot:

- α_i = de intensiteit om van een actieve naar een inactieve toestand te gaan voor component i ($i = 1$ voor het 1IPP-model en $i = 1$ of 2 voor het 2IPP-model),
- β_i = de intensiteit om van een actieve naar een inactieve toestand te gaan voor component i ($i = 1$ voor het 1IPP-model en $i = 1$ of 2 voor het 2IPP-model).



Figuur 4.6: Schema voor het 2IPP-model



Figuur 4.7: Transitietoestanden voor het 2IPP-model

De parameters α_i en β_i bepalen het onderbrekingsproces volledig. Als alternatief kan dit proces ook gekarakteriseerd worden door σ_i en κ_i die als volgt worden gedefinieerd. De parameter σ_i is de fractie van de tijd dat de toestand voor component i actief is,

$$\sigma_i = \frac{\beta_i}{\alpha_i + \beta_i}.$$

We noemen deze het *actieve percentage*. Het symbool κ_i , die we de *omschakeltijd* noemen, is gelijk aan de som van de gemiddelde lengte van de inactieve en actieve periode,

$$\kappa_i = \frac{1}{\alpha_i} + \frac{1}{\beta_i} = E[AAN] + E[UIT].$$

Naast α en β , moet μ ook bepaald worden. Om de verwerkingsintensiteit te kwantificeren moet deze vergeleken worden met κ_i . We veronderstellen dat er gemiddeld één kit per tijdseenheid kan worden samengesteld: $\mu = 1$.

Als laatste bepalen we λ_i aan de hand van de vergelijking:

$$\lambda_i = \sigma_i \cdot \lambda_i^*.$$

Dit betekent dat het product van de aankomstintensiteit in de actieve periode λ_i^* met het actieve percentage σ_i gelijk is aan de *werklast* λ_i van component i . Deze is gelijk aan de gemiddelde aankomstintensiteit over de actieve en inactieve periode. Omdat de productieperiode altijd actief is in het basismodel, is σ_i gelijk aan één en is de werklast gelijk aan de aankomstintensiteit.

We stellen ook vast dat de werklast dezelfde moet zijn voor beide componenten. Als dit niet het geval is en als de buffers voldoende groot zijn dan zal de buffer met de hoogste werklast zo goed als altijd vol zijn. Het systeem kan dan beschouwd worden als een wachtrij met slechts één buffer, namelijk degene die altijd vol zit. In dit eindwerk nemen we voor alle numerieke voorbeelden aan dat de werklast gelijk is aan 0,8 en dat de omschakeltijd gelijk is aan tien. Dit betekent dat in een omschakeltijd van tien tijdseenheden de componenten enkel gedurende gemiddeld acht tijdseenheden kunnen aankomen.

Tenslotte merken we op dat wanneer er naar eenzelfde werklast gestreefd wordt, het verkleinen van het actieve percentage σ_i gepaard gaat met een grotere aankomstintensiteit λ^* tijdens de actieve periodes.

4.3 Numerieke voorbeelden bij variërende capaciteit

In dit deel bespreken we de verschillen tussen de modellen bij een 80% werklast en variërende capaciteit. Voor de IPP-modellen stellen we de omschakeltijd gelijk aan tien en het actieve percentage gelijk aan 80% of 40%.

4.3.1 1IPP-model als $\sigma = 0,8$ en $\kappa = 10$

De parameters voor deze numerieke voorbeelden worden weergegeven in tabel 4.1.

Ingangparameters

1IPP-model	
Aankomstintensiteit in buffer 1 (actieve periode)	$\lambda_1^* = 1$
Aankomstintensiteit in buffer 2	$\lambda_2 = 0,8$
Overgangsintensiteit (uit naar aan)	$\beta = 0,5$
Overgangsintensiteit (aan naar uit)	$\alpha = 0,125$
Verwerkingsintensiteit	$\mu = 1$

Tabel 4.1: Ingangparameters voor het 1IPP-model als $\sigma = 0.8$

Grafieken

Voor de gemiddelde bufferbezetting en zijn variantie blijken we de verschillen met het basismodel zo goed als niet te zien. De kansen op volle en lege buffers zijn wel verschillend: ze liggen enigszins hoger in het 1IPP-model. We bespreken deze verschillen aan de hand van grafieken.

Kans dat één van beide buffers leeg is

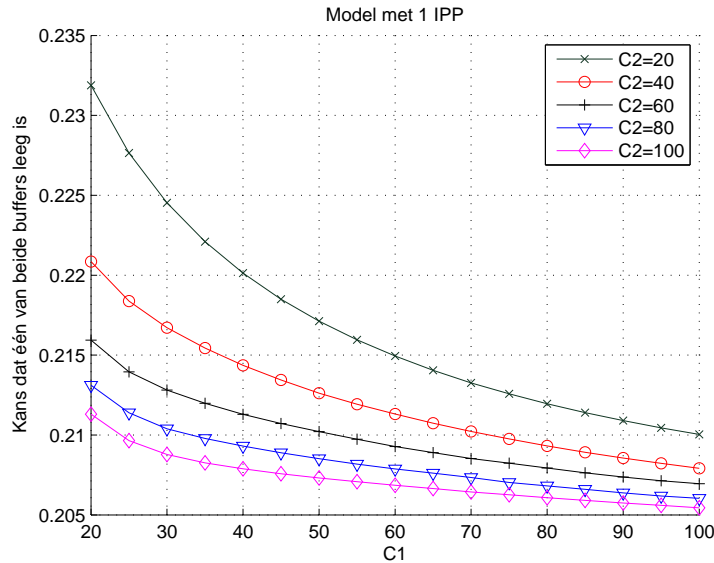
Figuur 4.8 toont de kans dat één van beide buffers leeg is bij variërende C_1 en een 80% actief percentage. Deze kans is theoretisch gerelateerd aan de verliesprobabiliteit (supra, p.37),

$$PLR = 1 - \frac{2 \cdot (1 - K_1) \cdot \mu}{\lambda_1 + \lambda_2} = 1 - \frac{2 \cdot \mu}{\lambda_1 + \lambda_2} + \frac{K_1}{\lambda_1 + \lambda_2}, \quad (4.1)$$

of nog,

$$K_1 = 1 - \frac{(1 - PLR) \cdot (\lambda_1 + \lambda_2)}{2 \cdot \mu} \quad (4.2)$$

Bij toenemende buffercapaciteit zal de verlieskans afnemen want er is meer ruimte om componenten op te slaan. We leiden aan de hand van bovenstaande formule dat K_1 ook zal afnemen. Bovendien in vergelijking met het basismodel, zal er bij het 1IPP model meer verlies zijn. In de actieve periodes zal de bufferbezetting snel toenemen waardoor de kans op verlies vergroot. Door het bovenstaande verband zal K_1 dan ook groter zijn bij het 1IPP model in vergelijking met het basismodel.



Figuur 4.8: 1IPP-model: Kans dat één van beide buffers leeg is bij variërende C_1 en $\sigma = 0,8$

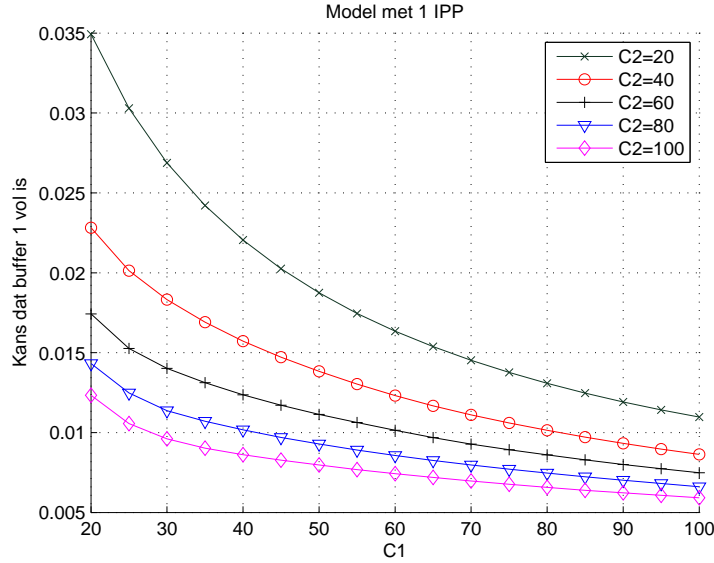
Kans dat buffer 1 vol is

We leiden uit figuur 4.9 en 3.12 af dat de kans dat buffer 1 vol is ook groter is in het 1IPP- dan in het basismodel. Hoewel de productieperiodes van component 1 korter zijn, komen er meer componenten toe in die periode, waardoor zijn buffer vaker vol zal zijn.

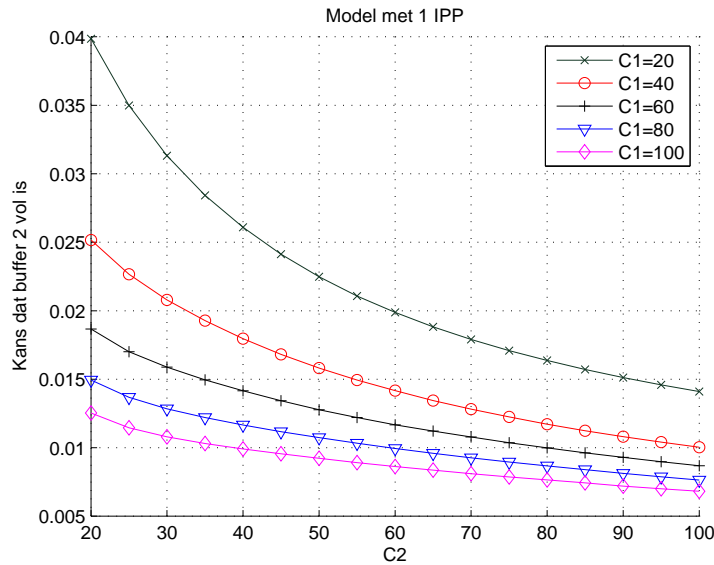
Kans dat buffer 2 vol is

Niet alleen voor buffer 1 is de kans groter maar ook voor buffer 2 (zie figuur 4.10 en 3.12). In tegenstelling tot in het basismodel zijn deze twee kansen niet gelijk aan elkaar. Het kittingmodel is immers niet meer symmetrisch. De kans dat buffer 2 vol is, is hier groter dan die voor buffer 1. We leiden daaruit af dat de impact van tijdelijke productieonderbrekingen van de ene component vooral invloed heeft op de andere buffer. Verder onderzoek moet uitwijzen of dit ook zo is voor andere waarden van σ en κ .

Wanneer de capaciteit van beide buffers voldoende groot is, dan is de uitkomst, zoals voor de stock-out-kans, zo goed als gelijk als die in het basismodel. Het is wel opvallend dat voor buffer 2 het verschil met het basismodel groter blijft dan voor buffer 1.



Figuur 4.9: 1IPP-model: Kans dat buffer 1 vol is bij variërende C_1 en $\sigma = 0,8$

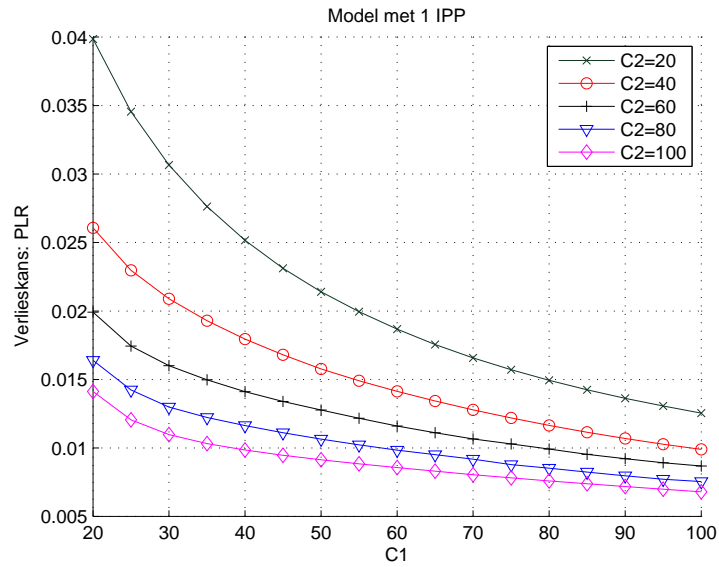


Figuur 4.10: 1IPP-model: Kans dat buffer 2 vol is bij variërende C_2 en $\sigma = 0,8$

Verlieskans voor beide buffers

De verlieskans voor beide buffers weergegeven in figuur 4.11 is, omwille van dezelfde reden als bij de vorige kansen, ook groter dan in het basismodel. Wanneer we C_2 laten variëren in plaats van C_1 dan merken we dat de verlieskans gelijk is aan de kans dat buffer 2 vol is. Dit komt doordat de aankomst van component 2 nog steeds volgens het Poisson-proces gemodelleerd is.

Samengevat stellen we vast dat naarmate de capaciteit groter wordt, de invloed van het productieonderbrekende



Figuur 4.11: 1IPP-model: Verlieskans voor beide buffers bij variërende C_1 en $\sigma = 0,8$

karakter verzwakt. De onregelmatigheid in productie wordt goed opgevangen worden wanneer de buffers groot zijn. Bovendien heeft de inefficiëntie in de productie van de ene component vooral invloed op de kansen van de andere buffer. In het volgende deel versterken we de verschillen tussen het 1IPP- en basismodel door de aankomstintensiteit van component 1 te verdubbelen.

4.3.2 1IPP-model als $\sigma = 0,4$ en $\kappa = 10$

We stellen λ_1^* gelijk aan twee en beschouwen een 40% actief percentage om eenzelfde werklast te behouden. We bespreken de verschillen met het vorige model.

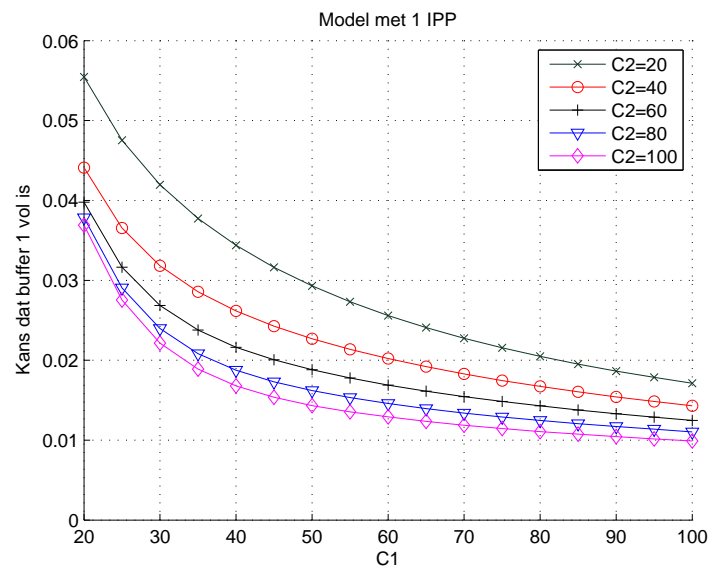
Ingangparameters

1IPP-model	
Aankomstintensiteit in buffer 1 (actieve periode)	$\lambda_1^* = 2$
Aankomstintensiteit in buffer 2	$\lambda_2 = 0,8$
Overgangsintensiteit (uit naar aan)	$\beta = 0,1667$
Overgangsintensiteit (aan naar uit)	$\alpha = 0,25$
Verwerkingsintensiteit	$\mu = 1$

Tabel 4.2: Ingangparameters voor het 1IPP-model als $\sigma = 0,4$

Grafieken

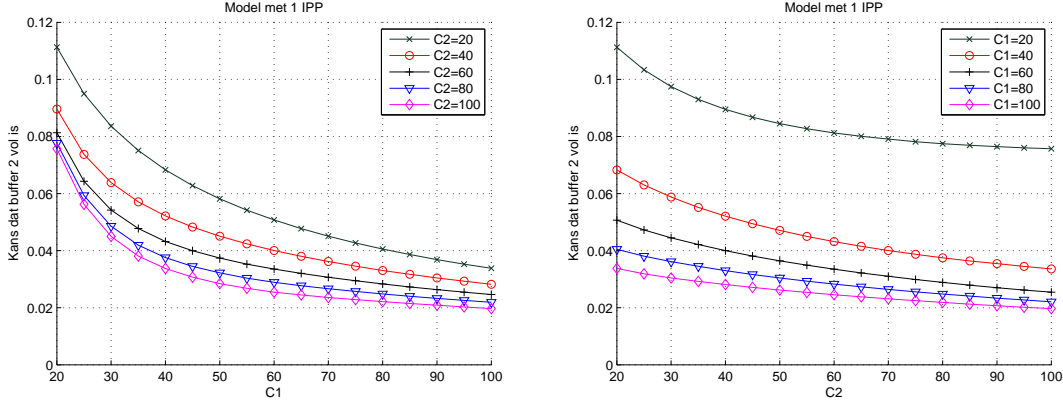
Kans dat buffer 1 vol is



Figuur 4.12: 1IPP-model: Kans dat buffer 1 vol is bij variërende C_2 en $\sigma = 0,4$

Figuur 4.12 toont de kans dat buffer 1 vol is bij variërende C_1 en een 40% actief percentage. Deze kans is hier groter dan bij het vorige model want in de actieve periodes komen er twee keer meer componenten van type 1 toe, waardoor zijn buffer vaker vol zal zijn.

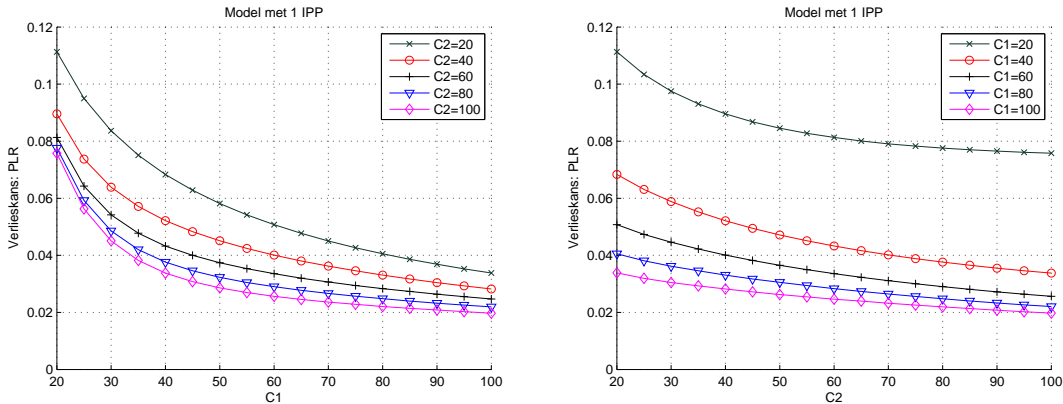
Kans dat buffer 2 vol is



Figuur 4.13: 1IPP-model: Kans dat buffer 2 vol is bij variërende C_1 (links) en C_2 (rechts) en $\sigma = 0,4$

We bespreken de kans dat buffer 2 vol is bij variërende C_2 . In deze grafiek ligt de lijn voor $C_1 = 20$ veel hoger dan die voor andere C_1 waarden. Bovendien blijft het verschil groot als $C_2 = 100$. We leiden daaruit af dat het effect van productieonderbreking van component 1 op buffer 2 sterk is wanneer de capaciteit van buffer 1 klein is.

Verlieskans in beide buffers



Figuur 4.14: 1IPP-model: Verlieskans voor beide buffers bij variërende C_1 en C_2 en $\sigma = 0,4$

Zoals in het vorige 1IPP-model concluderen we dat bij variërende C_2 of C_1 de verlieskans gelijk is aan de kans dat buffer 2 vol is. Zoals in het basismodel is de gelijkheid een gevolg van de PASTA-eigenschap. Het aankomstproces van component 2 wordt nog steeds gemodelleerd als een Poisson-proces. In het volgende deel bespreken we de verschillen tussen het 2IPP-, 1IPP- bij een 80% actief percentage en het basismodel.

4.3.3 2IPP-model als $\sigma_1 = \sigma_2 = 0,8$ en $\kappa_1 = \kappa_2 = 10$

In het 2IPP-model lijden beide componenten aan productieonderbrekingen. Doordat we hun aankomstprocessen identiek en onafhankelijk van elkaar modelleren, vertonen dezelfde prestatiematen voor de twee buffers (bijvoorbeeld de kans dat buffer 1 en buffer 2 vol is) eenzelfde trend. Dit zoals in het basismodel.

Ingangparameters

2IPP-model	
Aankomstintensiteit in buffer 1 (actieve periode)	$\lambda_1^* = [0 \quad 1 \quad 0 \quad 1]$
Aankomstintensiteit in buffer 2 (actieve periode)	$\lambda_2^* = [0 \quad 0 \quad 1 \quad 1]$
Generatormatrix	$A = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 \\ 0.125 & 0 & 0 & 0.5 \\ 0.125 & 0 & 0 & 0.5 \\ 0 & 0.125 & 0.125 & 0 \end{bmatrix}$
Verwerkingsintensiteit	$\mu = [1 \quad 1 \quad 1 \quad 1]$

Tabel 4.3: Ingangparameters voor het 2IPP-model als $\sigma_1 = \sigma_2 = 0,8$ en $\kappa_1 = \kappa_2 = 10$

Grafieken

Gemiddelde bezetting in buffer 1

Voor wat betreft het gemiddelde aantal in de buffer, als we figuur 4.15 vergelijken met die van het 1IPP- en basismodel dan stellen we vast dat deze quasi identiek zijn. Hetzelfde geldt voor de gemiddelde bezetting in buffer 2. Hieruit leiden we af dat bij een 80% actief percentage de impact van productieonderbrekingen van één of van beide onderdelen amper effect heeft op de gemiddelde bezetting.

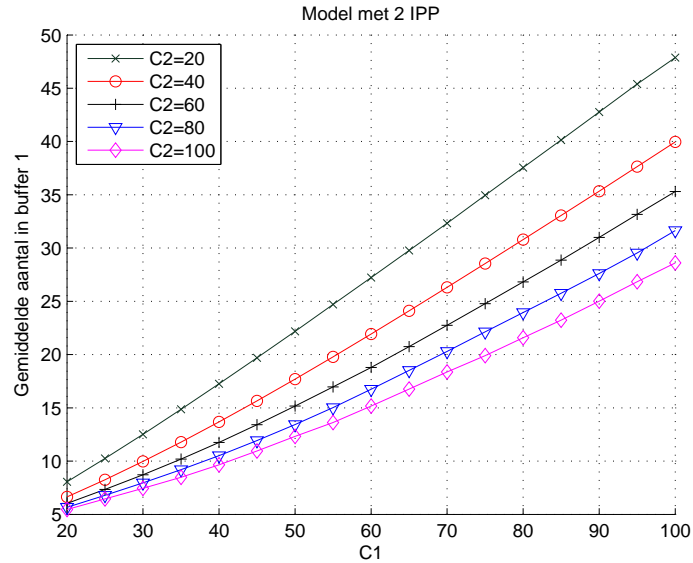
Kans dat één van beide buffers leeg is

Qua stock-out-kans weergegeven in figuur 4.16, stellen we vast dat deze het hoogst is bij een 2IPP-model. De uitleg gegeven bij figuur 4.11 is hier ook van toepassing. Voor het 2IPP-model zal de bufferbezetting in de actieve periodes nog sneller toenemen doordat zowel component 1 als component 2 aan productieonderbrekingen onderhevig zijn. Bij toenemende buffercapaciteit zullen de stock-out-kans en de verschillen tussen de drie modellen afnemen.

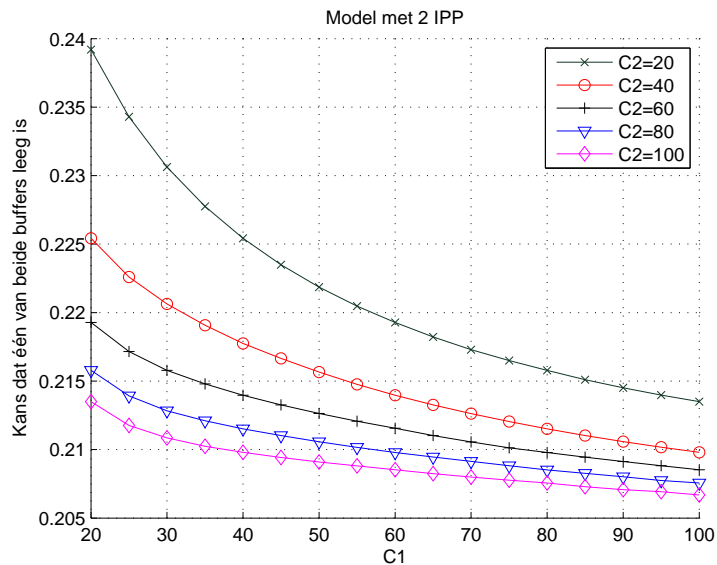
Kans dat buffer 1 vol is

Figuur 4.17 toont de kans dat buffer 1 vol is bij variërende C_1 en een 80% actief percentage voor beide onderdelen. Zoals in de vorige grafiek neemt deze kans de hoogste waarden aan bij het 2IPP-model. Daarnaast is deze grafiek

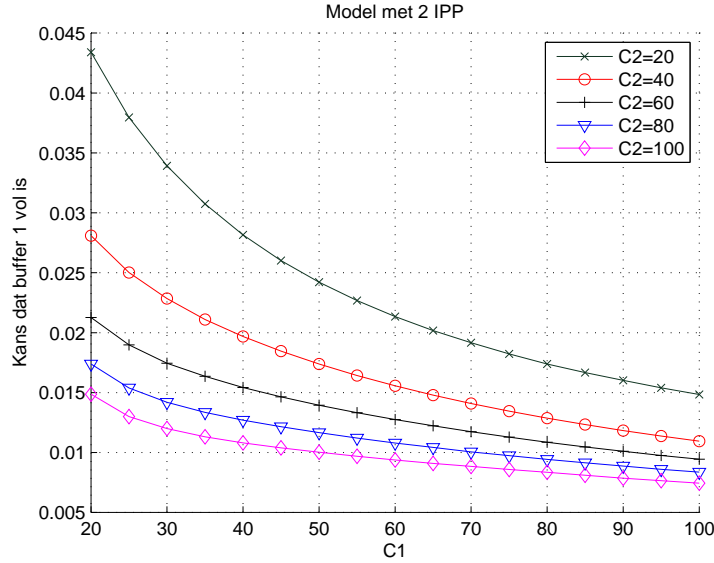
identiek aan die van buffer 2 omdat het een symmetrische situatie is: de twee buffers worden door hetzelfde aankomstproces aangevuld. Bij het 1IPP-model is dit niet het geval en bleek de productieonderbrekingen vooral invloed te hebben op de kans dat de buffer 2 vol is. Het modelleren van tijdelijke productieonderbrekingen bij component 2 doet de kans dat buffer 1 vol is, verhogen.



Figuur 4.15: 2IPP-model: Gemiddelde bezetting in buffer1 bij variërende C_1 en $\sigma_1 = \sigma_2 = 0,8$

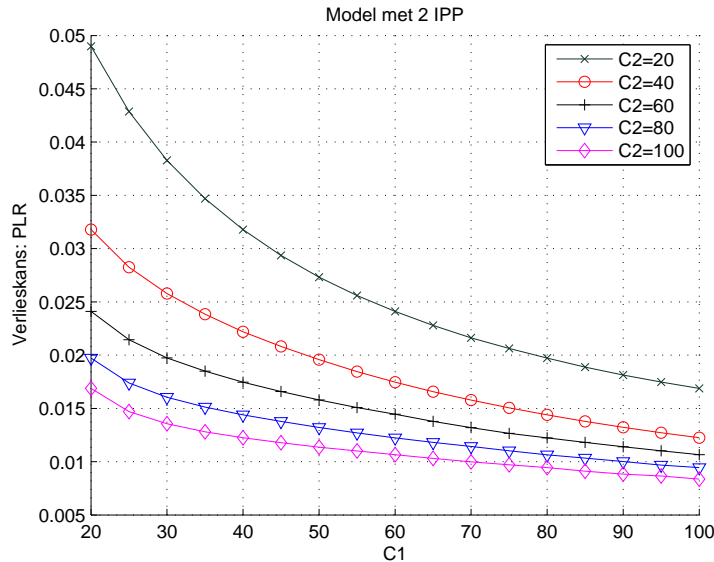


Figuur 4.16: 2IPP-model: Kans dat één van beide buffers leeg is bij variërende C_1 en $\sigma_1 = \sigma_2 = 0,8$



Figuur 4.17: 2IPP-model: Kans dat buffer 1 vol is bij variërende C_1 en $\sigma_1 = \sigma_2 = 0,8$

Verlieskans voor beide buffers



Figuur 4.18: 2IPP-model: Verlieskans voor beide buffers bij variërende C_1 en $\sigma_1 = \sigma_2 = 0,8$

Figuur 4.20 toont de verlieskans voor beide buffers in het 2IPP-model. Opnieuw blijkt dat deze kans het grootst is bij het 2IPP-model en het verschil met de twee andere modellen vermindert naarmate de capaciteit groter wordt. Als $C_1 = C_2 = 100$ dan ligt de verlieskans voor alle drie modellen tussen 0,005 en 0,01.

We merken ook dat de verlieskans enigszins hoger ligt dan de kans dat buffer 1 (en 2) vol is, wat niet het geval is bij het basis- en 1IPP-model. Voor het basismodel zijn de drie kansen gelijk aan elkaar en voor het 1IPP-model is de verlieskans gelijk aan de kans dat buffer 2 vol is. De reden is dat in het 2IPP-model beide aankomstprocessen

als een IPP gemodelleerd worden en dit aankomstproces de PASTA-eigenschap niet bezit. Dit betekent dat de stochastische eigenschappen van het proces afhangen van het tijdstip waarop het proces geobserveerd wordt: op aankomsttijdstippen zijn de eigenschappen niet hetzelfde als op andere willekeurige tijdstippen.

Samengevat blijken de grafieken te bevestigen dat de impact van productieonderbrekingen van de ene component vooral effect heeft op de kansen van de andere buffer. Echter, bij een 80% is er weinig invloed te merken bij de gemiddelde bufferbezetting en zijn variantie. We zien opnieuw dat het productieonderbrekende effect wegvalt wanneer de capaciteit van beide buffers voldoende groot is. We bespreken nu het 2IPP-model als $\sigma = 0,4$ en beschrijven kort de verschillen met dit model.

4.3.4 2IPP-model als $\sigma_1 = \sigma_2 = 0,4$ en $\kappa_1 = \kappa_2 = 10$

We bepalen het 2IPP-model waarin component 1 en 2 zich in 40% van de tijd in het proces kunnen aandienen. Opnieuw zorgen we ervoor dat beide aankomstprocessen identiek en onafhankelijk zijn van elkaar.

Ingangparameters

2IPP-model	
Aankomstintensiteit in buffer 1 (actieve periode)	$\lambda_1^* = [0 \quad 2 \quad 0 \quad 2]$
Aankomstintensiteit in buffer 2 (actieve periode)	$\lambda_2^* = [0 \quad 0 \quad 2 \quad 2]$
Generatormatrix	$A = \begin{bmatrix} 0 & 0,1667 & 0,1667 & 0 \\ 0,25 & 0 & 0 & 0,1667 \\ 0,25 & 0 & 0 & 0,1667 \\ 0 & 0,25 & 0,25 & 0 \end{bmatrix}$
Verwerkingsintensiteit	$\mu = [1 \quad 1 \quad 1 \quad 1]$

Tabel 4.4: Ingangparameters voor het 2IPP-model als $\sigma_1 = \sigma_2 = 0,4$ en $\kappa_1 = \kappa_2 = 10$

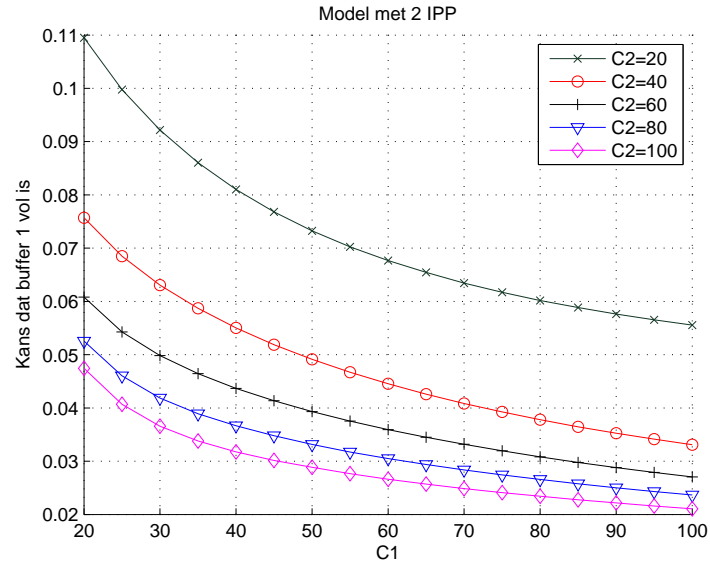
Grafieken

Kans dat buffer 1 vol is

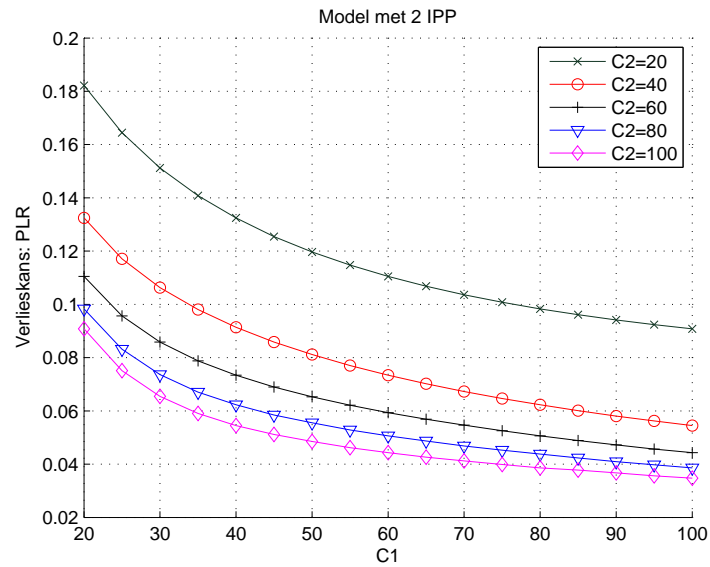
We bekijken de kans dat buffer 1 vol is in figuur 4.19. Voor $C_1 = C_2 = 20$ is deze kans gelijk aan 0,045 als $\sigma = 0,8$ terwijl deze gelijk is aan 0,11 wanneer $\sigma = 0,4$. De kans dat buffer 1 vol is, is dus groter naarmate het actieve percentage kleiner is. Naarmate de capaciteit groter wordt verkleint dit verschil.

Verlieskans voor beide buffers

Figuur 4.20 toont de verlieskans voor beide buffers. Zoals in het 1IPP-model geldt: als $\sigma = 0,4$ dan is deze kans groter dan wanneer $\sigma = 0,8$. We leiden uit de grafieken voor kansbuffers af dat deze het grootste zijn in een 2IPP-model met een klein actief percentage.



Figuur 4.19: 2IPP-model: Kans dat buffer 1 vol is bij variërende C_1 en $\sigma_1 = \sigma_2 = 0,4$



Figuur 4.20: 2IPP-model: Verlieskans voor beide buffers bij variërende C_1 en $\sigma_1 = \sigma_2 = 0,4$

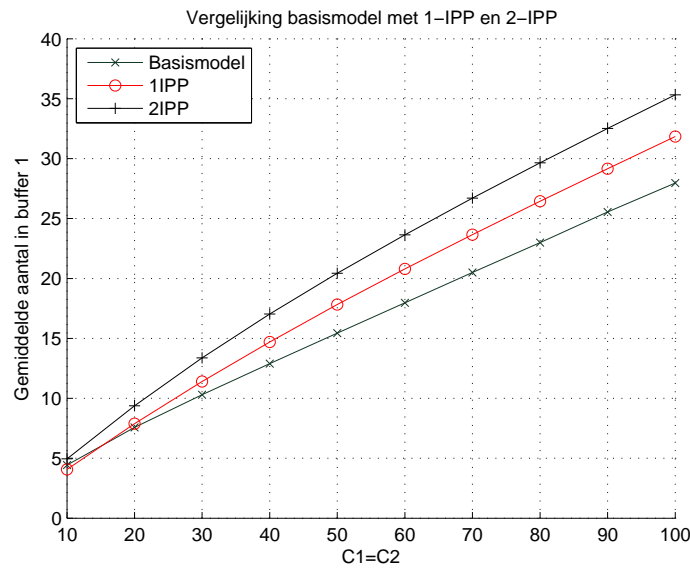
4.3.5 Modellen samen als $\sigma = 0,4$ en $\kappa = 10$

Naast aparte grafieken per model, hebben we ook grafieken gemaakt waarbij de drie modellen samen getekend zijn. We laten de capaciteit van beide buffers variëren op de x-as. Als we een actief percentage en werklast van 80% veronderstellen dan blijken de verschillen in prestatie-uitkomsten tussen de modellen zeer klein te zijn. Voor de drie modellen is het gemiddelde aantal in de buffers ongeveer gelijk en de kansen streven naar eenzelfde waarde als C_1 en C_2 voldoende groot zijn.

Om de verschillen tussen de modellen duidelijker te maken, intensifiëren we daarom de aankomsten. Zoals voor de modellen in de delen 4.3.2 en 4.3.4, behouden we een 80% werklast door de actieve periode korter te maken. We beschouwen voor de productieonderbrekende component dat $\sigma = 0,4$ en $\lambda^* = 2$. De waarden die we aan de parameters gegeven hebben staan in tabel 2 in de bijlage A.3.

Voor de meeste figuren blijkt de volgorde in uitkomst (van groot naar klein) als volgt te gaan: eerst het 2IPP- dan het 1IPP- en ten laatste het basismodel. Deze regel geldt voor buffer 1 maar niet altijd voor buffer 2.

Gemiddelde bezetting in buffer 1



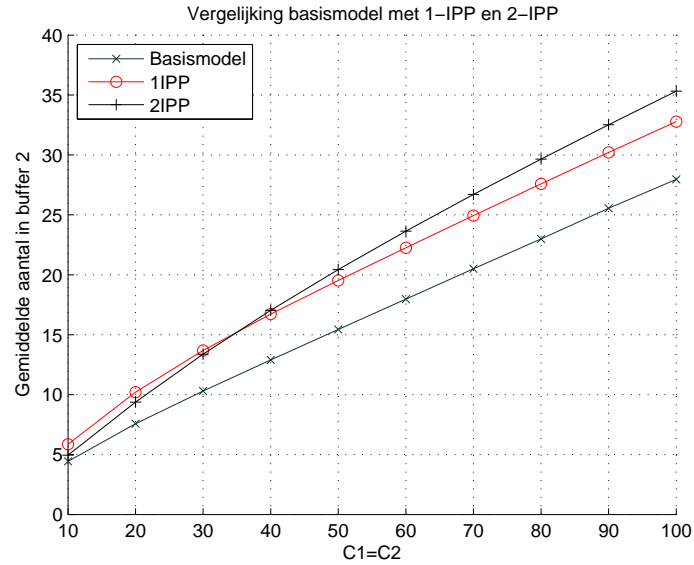
Figuur 4.21: Modellen samen: Gemiddelde bufferbezetting in buffer 1 bij variërende $C_1 = C_2$ en $\sigma = 0,4$

In figuur 4.21 schetsen we de gemiddelde bezetting in buffer 1 van de drie modellen. Voor $\sigma = 0,4$ blijken de verschillen tussen de drie modellen alsmate groter te worden naarmate de capaciteit stijgt.

Gemiddelde bezetting in buffer 2

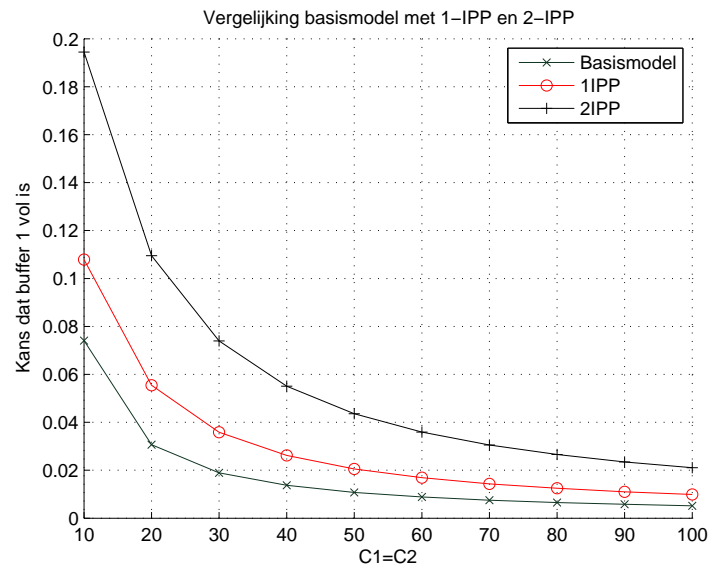
In figuur 4.22 merken we dat als $C_1 = C_2 = 35$ de twee IPP-modellen elkaar kruisen. Bij een kleinere capaciteit blijkt het 1IPP-model een hogere gemiddelde te hebben, daarna is de waarde het hoogst in het 2IPP-model.

Het verschil tussen de twee IPP-modellen blijkt zich te versterken naarmate de capaciteit groter wordt. Het verschil van deze modellen met het basismodel blijft ongeveer constant.



Figuur 4.22: Modellen samen: Gemiddelde bezetting in buffer 2 bij variërende $C_1 = C_2$ en $\sigma = 0,4$

Kans dat buffer 1 vol is

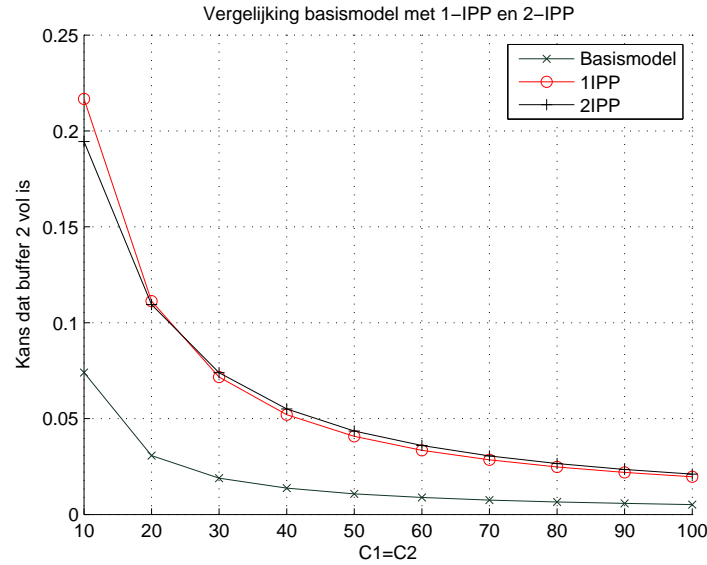


Figuur 4.23: Modellen samen: Kans dat buffer 1 vol is bij variërende $C_1 = C_2$ en $\sigma = 0,4$

Kans dat buffer 2 vol is

Een eerste vaststelling in figuur 4.24 is dat een onderbreking in de productie van component 1 een duidelijk negatieve impact heeft op de kans dat buffer 2 vol is. De 1IPP-lijn bevestigt het feit dat onderbrekingen in de productie van de ene component invloed heeft op de andere buffer.

Daarnaast blijkt dat het toevoegen van interrupties in component 2 de kans niet significant groter maakt: de twee

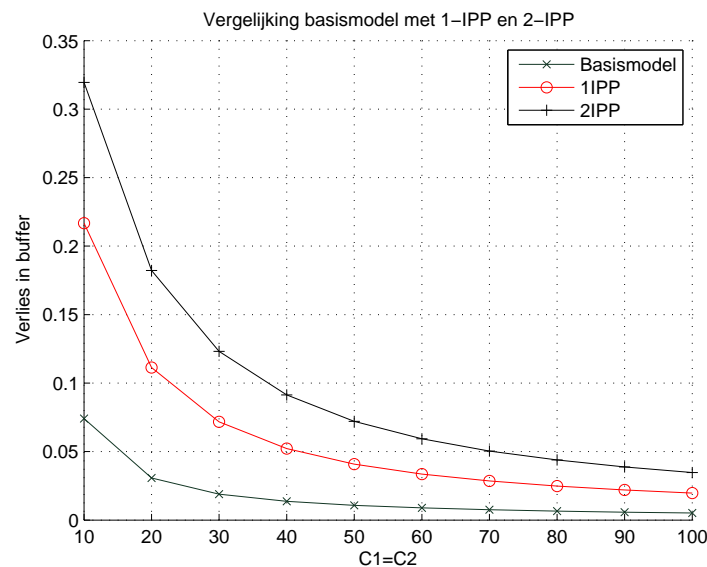


Figuur 4.24: Modellen samen: Kans dat buffer 2 vol is bij variërende $C_1 = C_2$ en $\sigma = 0,4$

IPP-lijnen zijn bijna identiek. Dit betekent dat van zodra we onderbrekingen aan de tweede component toevoegen, de kans dat zijn buffer vol is niet significant zal verhogen. Dit is wel het geval voor de kans dat buffer 1 vol is (zie figuur 4.23).

In tegenstelling tot het gemiddelde, stellen we tenslotte vast dat de drie modellen naar eenzelfde waarde streven naarmate de capaciteit groter wordt, namelijk nul.

Verlieskans voor beide buffers



Figuur 4.25: Modellen samen: Verlieskans voor beide buffers bij variërende $C_1 = C_2$ en $\sigma = 0,4$

Voor de verlieskans bijken de drie modellen opnieuw naar de waarde nul te convergeren. In sectie 4.4 laten we de aankomstintensiteit variëren in plaats van de capaciteit.

4.4 Numerieke voorbeelden bij variërende werklast

In deze laatste sectie vergelijken we het 1IPP- met het basismodel bij variërende werklast, i.e. de gemiddelde intensiteit over de actieve en inactieve periode (supra, p.48). Voor het basismodel hebben we als enige ingangparameter $\mu = 1$. Voor het 1IPP-model nemen we opnieuw een 40% actief percentage zodat de verschillen met het basismodel duidelijker zijn (zie tabel 4.5). We bespreken de gemiddelde bufferbezetting en de verlieskans voor beide buffers. Uit alle grafieken stellen we vast dat de werklast duidelijk invloed heeft op de prestatie van het proces.

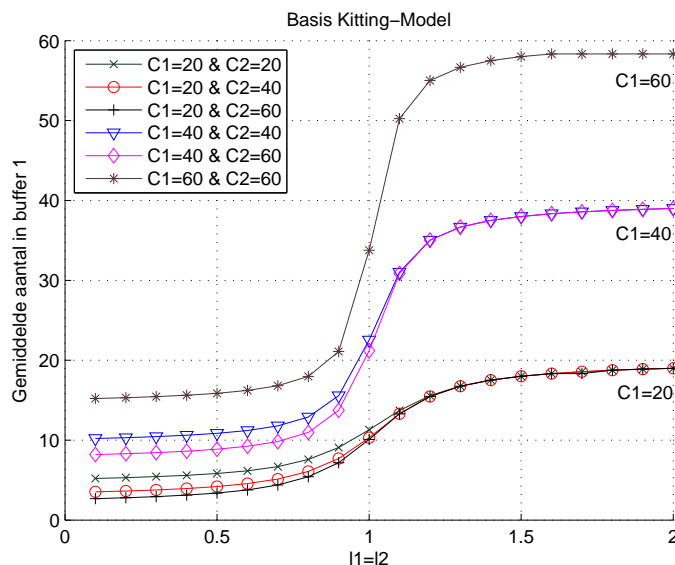
Ingangparameters

1IPP-model	
Overgangsintensiteit (uit naar aan)	$\beta = 0,1667$
Overgangsintensiteit (aan naar uit)	$\alpha = 0,25$
Verwerkingsintensiteit	$\mu = 1$

Tabel 4.5: Ingangparameters voor het 1PP-model als $\sigma = 0,4$

4.4.1 Basismodel

Gemiddelde bezetting in buffer 1



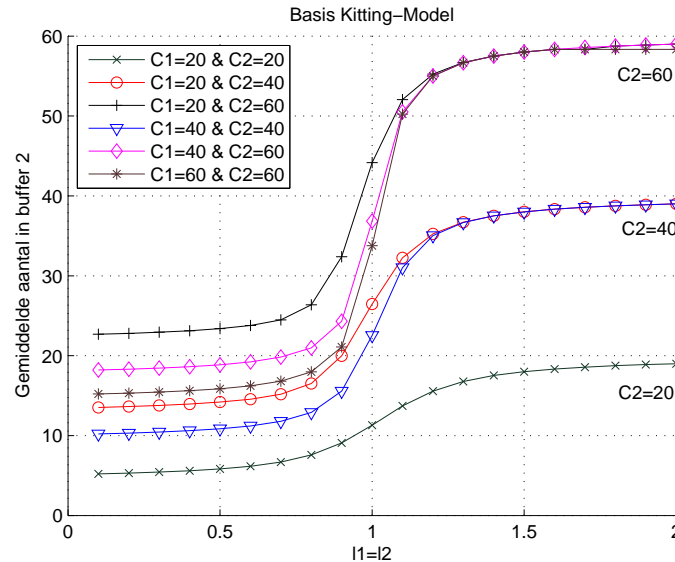
Figuur 4.26: Basismodel: Gemiddelde aantal in buffer 1 bij variërende $\lambda_1 = \lambda_2$

Figuur 4.26 toont het gemiddeld aantal in buffer 1. Deze stijgt heel sterk voor een werklast gelijk aan één. Doordat de verwerkingsintensiteit μ gelijk is aan één, bevinden we ons in een situatie van overbelasting, i.e. $\frac{\lambda}{\mu}$ is groter of gelijk aan één. Dit effect wordt versterkt naarmate C_1 groter is. Wanneer λ_1 en λ_2 ongeveer de waarde 1,2 bereiken, streeft de curve naar de buffercapaciteit C_1 . Dit betekent dat als de belasting voldoende hoog is, de buffer bijna altijd vol zal zijn.

We zien ook dat bij eenzelfde C_1 , de lijnen ongeveer dezelfde trend volgen. Voor een gegeven C_1 en werklast kleiner dan één, is de gemiddelde bezetting van buffer 1 groter naarmate C_2 kleiner is.

Gemiddelde bezetting in buffer 2

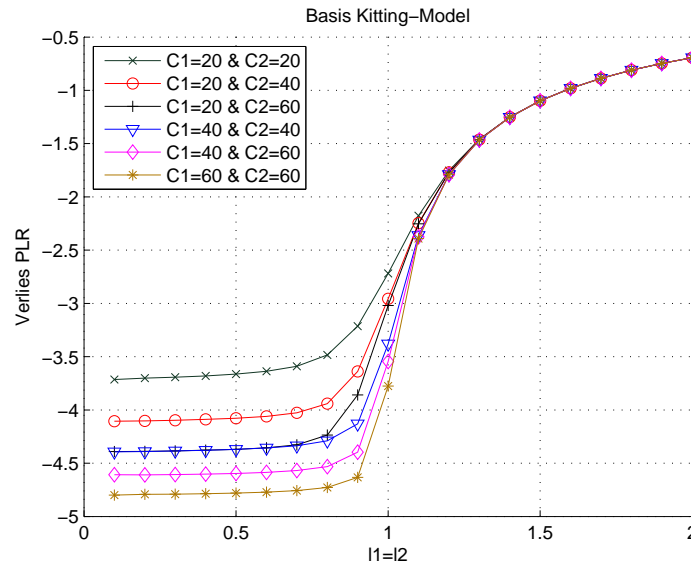
Voor het gemiddelde in buffer 2 in figuur 4.27 stellen we opnieuw dezelfde trend vast als voor buffer 1, waarbij dit keer voor een gegeven C_2 (en niet C_1) de lijnen naar eenzelfde waarde C_2 convergeren. Zoals in de vorige grafiek streven de lijnen naar de capaciteit van de onderzochte buffer.



Figuur 4.27: Basismodel: Gemiddelde aantal in buffer 2 bij variërende $\lambda_1 = \lambda_2$

Verlieskans voor beide buffers

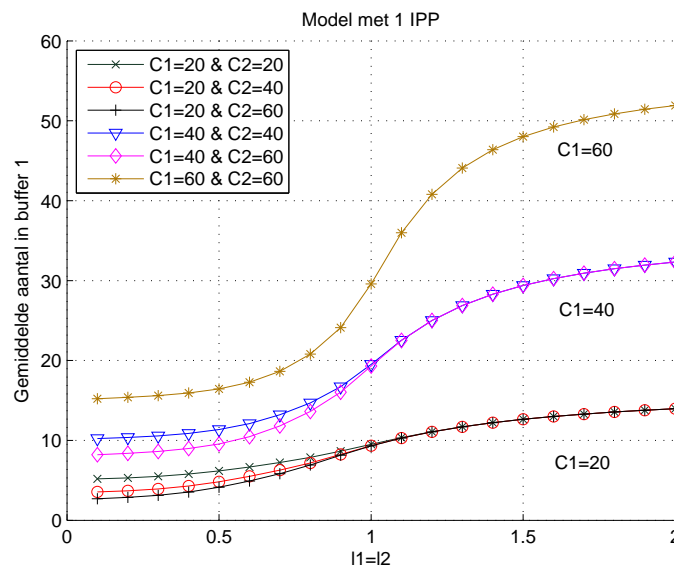
In verband met de verlieskans, gebruiken we voor de y-as in figuur 4.28 een logaritmische schaal. Dit laat ons toe de verschillen tussen de lijnen beter te zien. Het valt op dat, zoals het gemiddelde, de kans begint te stijgen wanneer $\lambda_1 = \lambda_2 = 0,8$. Deze stijging vermindert naarmate de werklast groter wordt. Als de belasting hoog genoeg is, dan zal de buffer bijna altijd vol zijn waardoor componenten bijna altijd zullen geweigerd worden.



Figuur 4.28: Basismodel: Verlieskans voor beide buffers bij variërende $\lambda_1 = \lambda_2$

4.4.2 1IPP-model

Gemiddelde bezetting in buffer 1



Figuur 4.29: 1IPP-model: Gemiddelde aantal in buffer 1 bij variërende $\lambda_1 = \lambda_2$ en $\sigma = 0,4$

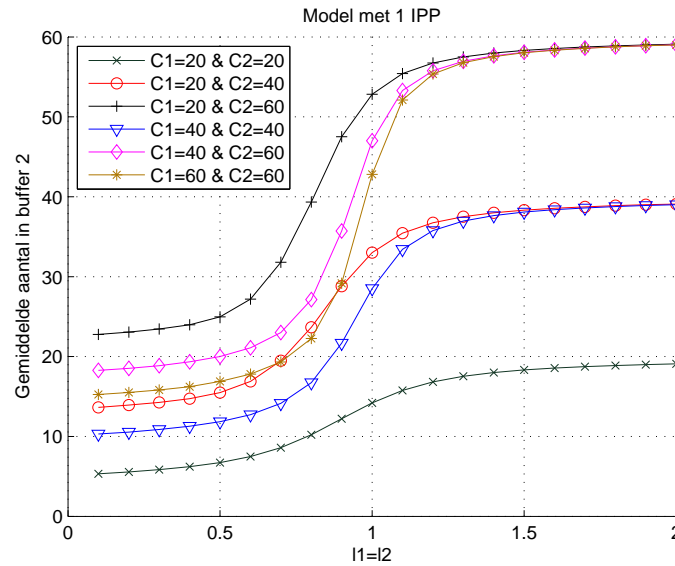
Uit figuur 4.29 en 4.26 blijkt dat het gemiddelde in buffer 1 voor het 1IPP-model dezelfde trend volgt als in het basismodel. Wanneer de werklast groter is dan 1,2 merken we dat het 1IPP-model lagere waarden heeft dan het basismodel. Dit betekent dat in een situatie van overbelasting, het gemiddelde trager naar C_1 convergeert als in het basismodel. In een situatie van onderbelasting geldt het omgekeerde: de waarden liggen enigszins hoger bij het

1IPP-model.

Gemiddelde bezetting in buffer 2

Wat betreft het gemiddelde in buffer 2, zijn de verschillen nauwelijks te merken (zie figuur 4.27 en 4.30). De waarden convergeren even snel naar C_2 .

Net zoals in het basismodel geldt: als de werklast kleiner is dan 1,2, dan blijkt de gemiddelde bezetting in buffer 2 groter te zijn bij de combinatie $C_1 = 20$ en $C_2 = 60$ dan wanneer $C_1 = 40$ en $C_2 = 60$. Dit doordat er minder kits kunnen verwerkt worden wanneer de capaciteit van buffer 1 kleiner is.

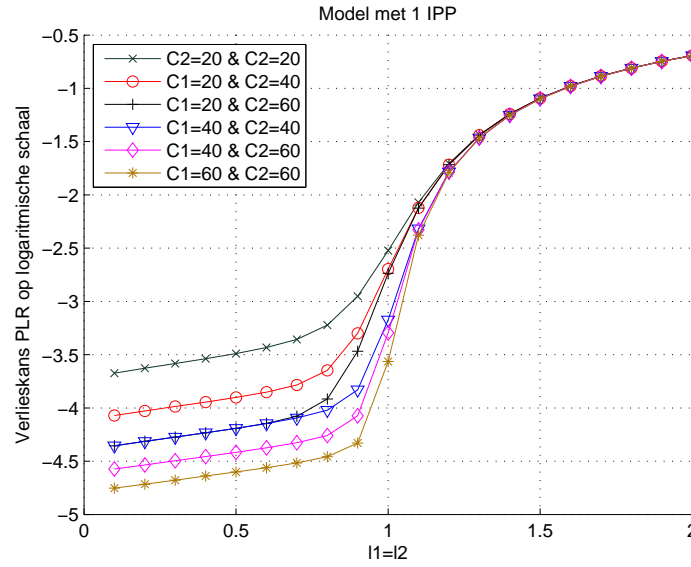


Figuur 4.30: 1IPP-model: Gemiddelde aantal in buffer 2 bij variërende $\lambda_1 = \lambda_2$ en $\sigma = 0,4$

Verlieskans voor beide buffers

Als we de verlieskans van het 1IPP- met het basismodel in de figuren 4.28 en 4.31 vergelijken, dan blijkt deze vooral groter te zijn als $\lambda = 1$. Als $\lambda = 2$ is de verlieskans in beide modellen ongeveer gelijk aan $10^{-0,5}$. Voor beide modellen gaat de kans dat één van beide buffers leeg is naar nul omdat er bijna altijd geproduceerd wordt. Bijgevolg wanneer $\lambda \gg 1$, dan is de doorvoer gelijk aan μ zodat **PLR** gelijk is aan $1 - \frac{\mu}{\lambda}$ (supra, p.37).

Samengevat, wanneer we het basis- met het 1IPP-model bij variërende werklast vergelijken, dan blijkt dat dit keer het effect van onderbreking in de productie vooral te merken is bij buffer 1. Bij steeds grotere overbelasting streeft het gemiddelde trager naar zijn capaciteitsbeperking dan bij buffer 2. De verlieskans convergeert bij beide modellen naar eenzelfde waarde.



Figuur 4.31: 1IPP-model: Verlieskans voor beide buffers bij variërende $\lambda_1 = \lambda_2$ en $\sigma = 0,4$

Hoofdstuk 5

Conclusies en verder onderzoek

5.1 Conclusies

Het doel van deze thesis was de impact van productieonderbrekingen op een kittingproces te onderzoeken. We bestudeerden een proces waarbij twee verschillende componenten, vooraleer deze samen te groeperen in een kit, opgeslagen worden in hun respectievelijke buffers. We veronderstelden een beperkte capaciteit voor de onderdelen-buffers: zodra een component zich in het proces aandient en zijn buffer vol is, gaat deze component verloren.

Om na te gaan wat de gevolgen zijn van inefficiëntie in de productie hebben we drie kittingmodellen opgesteld: een model waarbij het kittingproces niet aan productieonderbrekingen onderhevig is (i.e. het basismodel); een model waarbij component 1 aan interrupties in de productie lijdt (i.e. het 1IPP-model) en een model waarbij dit voor beide componenten het geval is (i.e. het 2IPP-model). Voor dit laatste veronderstelden we twee gelijkaardige en onafhankelijke aankomstprocessen. We modelleerden de onderbrekingen in de productie door de aankomst van de componenten in de tijd te beperken: ze kunnen enkel in een actieve productieperiode toekomen. We beschouwden een 80% werklast en voor de IPP-modellen een 80% en 40% actieve productieperiode. Eenzelfde werklast werd behouden door de aankomstintensiteit in het tweede geval te verdubbelen. De gebruikte parameters staan in de bijlage A.3. “Ingangparameters”. Aan de hand van prestatiematen vergeleken we de drie modellen (supra, p.34).

In de numerieke voorbeelden, die de prestatie-uitkomsten weergaven, lieten we onder andere de capaciteit van nul tot honderd variëren. Bij bijvoorbeeld een 2IPP-model waar de capaciteit van beide buffers gelijk is aan zestig, is het aantal op te lossen vergelijkingen gelijk aan 14884, wat zeer omvangrijk is. Om de onbekenden van dit stelsel te kunnen bepalen, maakten we gebruik van de GMRES-methode (Generalized Minimum Residual). Deze methode is iteratief en laat ons toe de prestatiematen op een snelle en nauwkeurige manier te berekenen.

Uit deze numerieke voorbeelden stellen we vooreerst vast dat voor een gegeven capaciteit en werklast, het basismodel het best presteert, gevolgd door het model waar component 1 aan productieonderbrekingen onderhevig is en tenslotte het model waar beide onderdelen dit ervaren. Het verschil in prestatie vermindert echter naarmate de capaciteit groter wordt, vooral als de capaciteit van de productieonderbrekende component vergroot. Er is voldoende ruimte in de buffer waardoor de kans kleiner is dat er componenten verloren gaan. Voor een 80% productieperiode en als de buffercapaciteit voldoende groot is dan convergeren de drie modellen naar eenzelfde waarde. Dit is niet het geval voor een 40% actieve periode. Hoewel we eenzelfde werklast in de twee scenario's veronderstellen, is de impact van productieonderbreking veel groter in het tweede dan in het eerste geval.

Daarnaast stellen we ook vast dat bij variërende capaciteit, inefficiënties in de productie van de ene component vooral te merken is op de andere buffer. Dit komt doordat de tweede buffer frequenter leeg in de actieve en vaker vol zal zijn in de inactieve periodes. Bij een 1IPP-model is de kans dat buffer 2 vol is hoger dan die voor buffer 1. Wanneer component 2 ook aan productieonderbrekingen onderhevig is, blijkt de kans dat buffer 1 vol is te verhogen en gelijk te zijn aan de kans dat buffer 2 vol is. Bijgevolg hebben productieonderbrekingen bij de ene component bij variërende capaciteit vooral een negatieve impact op de prestatie van de andere buffer.

Wanneer we echter de werklast laten variëren dan is de impact van inefficiëntie vooral te zien bij de buffer van de productieonderbrekende component. Het blijkt bijvoorbeeld dat voor een 40% actieve periode het gemiddelde in deze buffer veel trager zijn capaciteit bereikt dan in het basismodel. Dit is niet het geval voor de andere buffer. De invloed van productieonderbrekingen op de prestatie van het proces varieert dus in functie van de prestatie maat (op de y -as) en de afhankelijke variabele (op de x -as).

Als belangrijkste prestatie maat kiezen we om de verlieskans uitvoerig te bespreken. Een eerste interessante opmerking is dat voor de drie modellen de verlieskans voor de buffers apart en te samen dezelfde is. Dit komt doordat we eenzelfde werklast aan beide bronnen veronderstellen. Zou dit niet het geval zijn, dan zou het systeem bij grote capaciteit kunnen beschouwd worden als een wachtrij met slechts één buffer, namelijk degene die altijd vol zit. Daarnaast merken we op dat voor de numerieke voorbeelden van de IPP-modellen er een groot verschil is in verlieskans tussen een 80% en een 40% actieve periode en tussen een kleine en grote buffercapaciteit. Deze kans is het grootst bij korte productieperiodes en een kleine capaciteit voor beide buffers. Wanneer bovendien minstens één van de aankomstprocessen gemodelleerd wordt als een Poisson-proces, zoals in het basis- en 1IPP-model, dan blijkt deze kans gelijk te zijn aan de kans dat de buffer vol is. Voor het 1IPP-model geldt dat de verlieskans gelijk is aan kans dat buffer 2 vol is. In het 2IPP-model is dit niet het geval doordat het IPP voor beide bronnen geen PASTA-eigenschap heeft. De verlieskans zal hoger liggen dan de kans dat de buffer vol is. Zoals voor de andere prestatie maten blijkt de verlieskans, naast de capaciteit, ook afhankelijk te zijn van de werklast. Wanneer we het 1IPP- met het basismodel vergelijken dan zijn de verschillen voor deze prestatie maat enkel te merken bij een werklast kleiner dan 1. Beide modellen convergeren naar eenzelfde waarde naarmate de werklast groter wordt.

5.2 Verder onderzoek

Wachttijmodellen om de prestatie van kittingprocessen te bepalen worden momenteel weinig bestudeerd. Er is bijgevolg zeker ruimte voor verder onderzoek.

Vooreerst zouden de opgesomde veronderstellingen uit de inleiding geleidelijk kunnen verlicht of verwijderd worden. We beperkten ons tot twee onderdelen, terwijl het proces zich gemakkelijk tot meerdere onderdelen zou kunnen uitbreiden. Daarnaast gingen we ervan uit dat de buffers onderdeel per onderdeel worden aangevuld en dat kits één na één vertrekken. In werkelijkheid kunnen meerdere onderdelen tegelijkertijd toekomen en samengesteld worden. Dit is de combinatie van een batch- met een kittingproces welke besproken werd in hoofdstuk 2 (supra, p.4). Tenslotte beperkten we onze analytische studie tot een tachtig percent werklast en een omschakeltijd gelijk aan tien terwijl andere waarden ook mogelijk waren.

Na het opstellen van de modellen, werd de impact van productieonderbrekingen bepaald op basis van verschillen in prestatie-uitkomsten. De gekozen prestatiematen waren beperkt en enkel gericht op de onderdelenbuffers. We zouden ook analoge maten kunnen definiëren voor de kitbuffer. Daarnaast zou het interessant zijn om de *kit-vroegheid* (de kits zijn voorbereid maar ze zijn nog niet vereist aan de assemblagelijijn) en de *kit-traagheid* (de kits zijn vereist aan de assemblagelijijn maar ze zijn nog niet klaar) te bepalen.

Om de realiteit beter te benaderen, zouden we dit proces ook kunnen integreren in een productieproces. Bijkomende factoren die invloed hebben op de prestatie van het proces kunnen dan in rekening gebracht worden. Als bedrijven kittingactiviteiten gaan gebruiken, is het - naast de prestatie - bovendien relevant de kost van dit proces te berekenen. Een interessant onderzoek is de capaciteit van de buffers en/of de doorvoer van de onderdelen afstemmen op de kittingactiviteit om een algemene kostenminimalisatie te bekomen.

Hoewel de wachttijmodellen specifiek opgebouwd zijn voor kittingprocessen, zou deze analysetechniek tenslotte ook in andere domeinen toegepast kunnen worden, bijvoorbeeld voor buffers in telecommunicatiesystemen.

Bibliografie

- R. Ackerley. *Telecommunications Performance Engineering, Classical Microscopic Theory, Markov Modulated Poisson Processes*. The Institution of Electrical Engineers, p.17, London, U.K., 2003.
- Y.A. Bozer en L.F. McGinnis. Kitting versus line stocking: A conceptual framework and a descriptive model. *International Journal of Production Economics*, 28:1–19, 1992.
- H. Bruneel. *Wachtligntheorie, Wachtlignmodellen in continue tijd*. Faculteit Toegepaste Wetenschappen in de Universiteit Gent, Vakgroep TELIN, 2008.
- H. Bryznér en M.I. Johansson. Design and performance of kitting and order picking systems. *International Journal of Production Economics*, 41:115–125, 1995.
- O. Carlsson en B. Hensvold. Kitting in a high variation assembly line: a case study at caterpillar BCP-E, theoretical framework. Master's thesis, Luleå University of Technology, p.6-21, 2008.
- C. G. Cassandras en S. Lafortune. *Introduction to Discrete Event Systems, Second Edition*. Springer Science and Business Media, 2008.
- F.Y. Ding en P. Balakrishnan. Kitting in just-in-time production. *Production and Inventory Management Journal*, 31(4):25–28, 1990.
- R. Frederix. *Gids voor Management en Organisatie: aanvullende module*. De Boeck, p.13-14, 2008.
- J.M. Harrison. Assembly-like queues. *Journal Of Applied Probability*, 10(2):354–367, 1973.
- D. P. Heyman en M. J. Sobel. *Stochastic models in Operation Research: Stochastic processes and Operating Characteristics*. Mc Graw-Hill Book Company, p.112, 1982.
- W. J. Hopp en J. T. Simon. Bounds and heuristics for assembly-like queues. *Queueing Systems*, 4:137 – 156, 1989.
- B. Johansson en M.I. Johansson. High automated Kitting system for small parts: a case study from the Volvo uddevalla plant. In *Proceedings of the 23rd International Symposium on Automotive Technology and Automation*, p.75-82, Vienna, Austria, 1990.
- E. Johansson en M.I. Johansson. Materials supply systems design in product development projects. *International journal of Operations and Production Management*, 26:225–230, 2006.
- M.I. Johansson. Kitting systems for small size parts in manual assembly systems. In M. Pridham en C. Obrien, editors, *Production Research: Approaching the 21st century*. Taylor and Francis Ltd, p.225-230, 1991.
- G. Latouche. Queues with paired customers. *Journal of Applied Probability*, 18:684–696, 1981.

- V. Limère. A decision model for kitting. Technical report, Department of Industrial Management, The University of Ghent, p.1, 2007.
- E.H. Lipper en B. Sengupta. Assembly-like queues with finite capacity: bounds, asymptotics and approximations. *Queueing Systems: Theory and Applications*, 18:684, 1986.
- New Breed Logistics. Kitting and sub-assembly services increase manufacturing efficiency, aid lean manufacturing strategy, 2007, URL :<<http://www.newbreed.com>>. (04/11/2009).
- Fumiaki Machihara. A fractal poisson process and its input queue. *An International Journal Computers & Mathematics with applications*, 51:199–200, 2006.
- R. C. McKinnon. *Cause, Effect and Control of Accidental Loss with Accident Investigation Kit, Chapter 12: Business interruption*. CRC, p.159-161, 2000.
- L. Medbo. Assembly work execution and materials kit functionality in parallel flow assembly systems. *International Journal of Industrial Ergonomics*, 31:263 – 281, 2003.
- A. Popescu. *Tools for modern Techniques of Networking: Markov-Modulated Poisson Processes*. Departement of Telecommunication Systems, p.1, 2000.
- S. Ramachandran en D. Delen. Performance analysis of a Kitting process in stochastic assembly systems. *Computers & Operations Research*, 32(3):449 – 463, 2005.
- R. Ramakrishnan en A. Krishnamurthy. Analytical approximations for Kitting systems with multiple inputs. *Asia-Pacific Journal of Operations Research*, 25(2):187 – 216, 2008.
- A. Roubos. Call Centers, appendix: het Poisson Proces. 2009, URL: <<http://www.few.vu.nl/~aroubos/Diktaat-PoissonProces.pdf>>. (22/02/2010).
- F. C. Schoute. *Prestatie-analyse van Telecommunicatiesystemen*. Information and Communication Theory Group, Technische Universiteit Delft, Geert Awater and Dirk Sparreboom edition, 1995.
- G.F. Schwind. How storage systems keep kits moving. *Material Handling Engineering*, 47(12):43–45, 1992.
- P. Som, W. Wilhelm, en R. Disney. Kitting process in a stochastic assembly system. *Queueing Systems*, 17:471 – 490, 1994.
- OEE Industry Standard. Definition of several types of time, 2003, URL:<<http://www.oestandard.com>>. (21/02/2010).
- P. Tran-Gia. A class of Renewal Interrupted Poisson Process and applications to queueing systems. *Zeitschrift für Operations Research*, 32:231, 1988.
- R. Vujosevic. Lean kitting: A case study. Technical report, Department of Mechanical Engineering, The University of Texas at Austin, p.3, 1997.

Index

- 1IPP-model, 43
- 2IPP-model, 45

- aankomstintensiteit, 12, 21
- aankomstproces, 11
- aankomsttoestand, 43, 45
- actieve periode, 13, 43

- basismodel, 22
- belasting, 39

- capaciteit, 21

- forward Chapman-Kolmogorov vergelijking, 16

- gemiddelde bezetting in de buffer, 34
- generatormatrix, 16, 24, 46
- gmres functie, 30

- i.i.d., 12
- inactieve periode, 13, 43
- ingangparameters, ix
- IPP, 12

- kans dat één van beide buffers leeg is, 36
- kans dat de buffer vol is, 36
- kansmassafunctie, 34
- kit, 4
- kitting, 5
- kittingproces, 6
- kitverwerkingsproces, 11

- lean kitting, 6
- lean manufacturing, 6

- Markov-eigenschap, 11
- Markov-keten met continue-tijdparameter, 13
- Markov-proces, 11
- MATLAB, 28

- normeringsvoorwaarde, 25
- numerieke voorbeelden, 38, 49, 65

- overgangsimpensiteit, 13, 43, 46

- PASTA-eigenschap, 12
- Poisson-proces, 12
- prestatiepaten, 27
- productieonderbreking, 8

- sparse functie, 29
- sparse-methode, 28
- stationaire waarschijnlijkheidstoestand, 17, 24
- stationaire waarschijnlijkheidsvector, 17, 25
- symbolen, v

- toestand, 22, 43, 46
- toestandsdiagram, 23
- toestandsruimte, 13
- totale probabiliteitswet, 14
- transitiefuctie, 14
- transitiematrix, 15, 30

- variantie, 35
- verlieskans, 37
- verstreken tijd, vii
- verwerkingsintensiteit, 21
- Volvo Europa Trucks, 5

- werklast, 48

Bijlage A

Bijlagen

A.1 Lijst van de gebruikte symbolen

Kittingmodel

C_1 : maximum capaciteit van buffer 1,

C_2 : maximum capaciteit van buffer 2,

X_1 : bezetting van buffer 1 op een willekeurig moment,

X_2 : bezetting van buffer 2 op een willekeurig moment.

Toestand

(X_1, X_2) : toestand in een basismodel,

(i, X_1, X_2) : toestand in een 1IPP-model ($i = 0$ in een uit-toestand en $i = 1$ in een aan-toestand),

(i, X_1, X_2) : toestand in een 2IPP-model ($i = 1$: beide aankomstprocessen zijn in een uit-toestand, $i = 2$: aan-toestand voor component 1 en uit-toestand voor component 2, $i = 3$: uit-toestand voor component 1 en aan-toestand voor component 2 en $i = 4$: beide aankomstprocessen zijn in een aan-toestand).

Wachtlijntheorie

Q : generatormatrix,

q_i : kansmassafunctie van buffer i ,

π : stationaire waarschijnlijkheidsvector,

$\pi_{i,j}$: waarschijnlijkheidstoestand, i.e. waarschijnlijkheid om zich in toestand (i, j) te bevinden.

Intensiteit

q_{ij} : algemene notatie voor de intensiteit (intensiteit om van toestand i naar toestand j te evolueren),

α_i : intensiteit om van een actieve naar een inactieve periode te gaan voor component i ($i = 1$ voor het 1IPP-model en $i = 1$ of 2 voor het 2IPP-model),

β_i : intensiteit om van een inactieve naar een actieve periode te gaan voor component i ($i = 1$ voor het 1IPP-model en $i = 1$ of 2 voor het 2IPP-model),

λ_i : werklust van component i (gemiddelde aankomstintensiteit over de actieve en inactieve periode),

λ_i^* : aankomstintensiteit van component i over de actieve periode in het 1IPP-model,

$\lambda_i^* = \begin{bmatrix} \lambda_{i1} & \lambda_{i2} & \lambda_{i3} & \lambda_{i4} \end{bmatrix}$: vector van aankomstintensiteiten voor het 2IPP-model. Het geeft de aankomstintensiteit over de actieve periode weer voor respectievelijk aankomsttoestand 1, 2, 3 en 4,

μ : verwerkingsintensiteit tot een kit bij het basis- en 1IPP-model,

$\mu = \begin{bmatrix} \mu_1 & \mu_2 & \mu_3 & \mu_4 \end{bmatrix}$: vector van verwerkingsintensiteiten voor het 2IPP-model voor respectievelijk aankomsttoestand 1, 2, 3 en 4,

A: matrix voor een 2IPP-model die alle intensiteiten weergeeft om vanuit een gegeven aankomsttoestand te evolueren naar een nieuwe aankomsttoestand (ook genoemd als een generatormatrix).

Actieve periode in IPP-modellen

κ_i : omschakeltijd; het is gelijk aan de som van de gemiddelde lengte van een inactieve en actieve periode voor component i ($i = 1$ voor het 1IPP-model en $i = 1$ of 2 voor het 2IPP-model),

σ_i : actief percentage; het is gelijk aan de fractie van de tijd dat de toestand voor component i actief is, i.e. de periode waarin componenten geproduceerd worden ($i = 1$ voor het 1IPP-model en $i = 1$ of 2 voor het 2IPP-model).

Kansen

K_1 : stock-out-kans, i.e. de kans dat één van beide buffers leeg is,

K_2 : kans dat buffer 1 vol is,

K_3 : kans dat buffer 2 vol is,

PLR: verlieskans voor beide buffers,

PLR₁: verlieskans voor buffer 1,

PLR₂: verlieskans voor buffer 2,

A.2 Verstreken tijd

Eenmalige berekening

$C_1 = 10$ en $C_2 = 10$		
Basismodel	Verstreken tijd	Percentage
Generatormatrix	0,002631 sec.	7,87%
Stationaire waarschijnlijkheidsvector	0,027907 sec.	83,48%
Prestatiematen	0,002893 sec.	8,65%
Totaal	0,033431 sec	100%

1IPP-model	Verstreken tijd	Percentage
Generatormatrix	0,021297 sec.	15,59%
Stationaire waarschijnlijkheidsvector	0,089267 sec.	65,37%
Prestatiematen	0,026000 sec.	19,04%
Totaal	0,136564 sec.	100%

2IPP-model	Verstreken tijd	Percentage
Generatormatrix	0,075498 sec.	37,63%
Stationaire waarschijnlijkheidsvector	0,095276 sec.	47,49%
Prestatiematen	0,029863 sec.	14,88%
Totaal	0,200637 sec.	100%

Tabel A.1: Verstreken tijd bij een eenmalige berekening van prestatiematen

Numerieke voorbeelden

$\sigma = 80\%$	Verstreken tijd
Basismodel	344,95 sec.
1IPP-model	960,68 sec.
2IPP-model	3024,62 sec.
Modellen samen	567,73 sec.

$\sigma = 40\%$	Verstreken tijd
Basismodel	344,95 sec.
1IPP-model	1215,45 sec.
2IPP-model	4293,62 sec.
Modellen samen	598,09 sec.

Tabel A.2: Verstreken tijd wanneer de capaciteit varieert

	Verstreken tijd
Basismodel	72,56 sec.
1IPP-model	327,84 sec.

Tabel A.3: Verstreken tijd wanneer de aankomstintensiteit varieert

A.3 Ingangparameters

Tabel 1: actieve productieperiode van 80% en omschakeltijd gelijk aan 10.

Basismodel	
Aankomstintensiteit in buffer 1	$\lambda_1 = 0,8$
Aankomstintensiteit in buffer 2	$\lambda_2 = 0,8$
Verwerkingsintensiteit	$\mu = 1$
1IPP-model	
Aankomstintensiteit in buffer 1 (actieve periode)	$\lambda_1^* = 1$
Aankomstintensiteit in buffer 2	$\lambda_2 = 0,8$
Overgangsintensiteit (uit naar aan)	$\beta = 0,5$
Overgangsintensiteit (aan naar uit)	$\alpha = 0,125$
Verwerkingsintensiteit	$\mu = 1$
2IPP-model	
Aankomstintensiteit in buffer 1 (actieve periode)	$\lambda_1^* = [0 \quad 1 \quad 0 \quad 1]$
Aankomstintensiteit in buffer 2 (actieve periode)	$\lambda_2^* = [0 \quad 0 \quad 1 \quad 1]$
Generatormatrix	$A = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 \\ 0.125 & 0 & 0 & 0.5 \\ 0.125 & 0 & 0 & 0.5 \\ 0 & 0.125 & 0.125 & 0 \end{bmatrix}$
Verwerkingsintensiteit	$\mu = [1 \quad 1 \quad 1 \quad 1]$

Tabel A.4: Ingangparameters als $\sigma = 0,8$ en $\kappa = 10$

Tabel 2: actieve productieperiode van 40% en omschakeltijd gelijk aan 10.

Basismodel	
Aankomstintensiteit in buffer 1	$\lambda_1 = 0,8$
Aankomstintensiteit in buffer 2	$\lambda_2 = 0,8$
Verwerkingsintensiteit	$\mu = 1$

1IPP-model	
Aankomstintensiteit in buffer 1 (actieve periode)	$\lambda_1^* = 2$
Aankomstintensiteit in buffer 2	$\lambda_2 = 0,8$
Overgangsintensiteit (uit naar aan)	$\beta = 0,1667$
Overgangsintensiteit (aan naar uit)	$\alpha = 0,25$
Verwerkingsintensiteit	$\mu = 1$

2IPP-model	
Aankomstintensiteit in buffer 1 (actieve periode)	$\lambda_1^* = [0 \quad 2 \quad 0 \quad 2]$
Aankomstintensiteit in buffer 2 (actieve periode)	$\lambda_2^* = [0 \quad 0 \quad 2 \quad 2]$
Generatormatrix	$A = \begin{bmatrix} 0 & 0,1667 & 0,1667 & 0 \\ 0,25 & 0 & 0 & 0,1667 \\ 0,25 & 0 & 0 & 0,1667 \\ 0 & 0,25 & 0,25 & 0 \end{bmatrix}$
Verwerkingsintensiteit	$\mu = [1 \quad 1 \quad 1 \quad 1]$

Tabel A.5: Ingangparameters als $\sigma = 0,4$ en $\kappa = 10$

A.4 De generatormatrix in MATLAB voor het 2IPP-model

We bespreken hoe we de generatormatrix voor het 2IPP-model opgesteld hebben. Het totaal aantal toestanden in dit model is gelijk aan $(C_1 + 1) \cdot (C_2 + 1) \cdot 4$ waarbij 4 het aantal aankomsttoestanden weergeeft. In aankomsttoestand 1 kunnen beide onderdelen niet geproduceerd worden. Wanneer dit wel het geval is, zitten we in aankomsttoestand 4; wanneer enkel component 1 geproduceerd kan worden, in aankomsttoestand 2; en wanneer dit het geval is voor component 2 in aankomsttoestand 3. Een toestandsverandering wordt, zoals in het 1IPP-model, door zeven waarden bepaald: de drie eerste waarden bepalen de huidige toestand, i.e. het aantal componenten in buffer 1 en 2 en de aankomsttoestand, de drie volgende de nieuwe toestand en de laatste de intensiteit waarmee de huidige naar de nieuwe toestand evolueert. Dit geeft bijvoorbeeld een huidige toestand $(4, 0, 0)$ die naar een nieuwe toestand $(4, 1, 0)$ evolueert met intensiteit $\lambda_1 = 1$. Deze getallen worden naast elkaar weergegeven in de generatormatrix.

Om deze op te stellen bepalen we eerst het aantal veranderingen in de toestanden verbonden aan de intensiteiten λ_1 , λ_2 en μ voor elke aankomsttoestand. Dit zijn veranderingen die verbonden zijn aan wijzigingen in het aantal componenten in de buffers. Voor λ_1 zijn deze gelijk aan $C_1 \cdot (C_2 + 1)$, voor λ_2 aan $(C_1 + 1) \cdot C_2$ en voor μ aan $(C_1 \cdot C_2)$. Belangrijk is dat niet iedere aankomsttoestand gekenmerkt wordt door dezelfde intensiteiten. In aankomsttoestand 1 zijn aankomsten van component 1 en 2 niet mogelijk zodat de toestandsveranderingen enkel gekenmerkt wordt door μ . In aankomsttoestand 2 daarentegen kunnen componenten in buffer 1 toekomen zodat het aantal toestandsveranderingen gelijk is aan $(C_1 \cdot C_2) + C_2 \cdot (C_1 + 1)$, enzovoort voor de twee andere aankomsttoestanden. Daarnaast moeten ook de overgangen naar andere aankomsttoestanden bepaald worden: voor iedere aankomsttoestand is dit aantal gelijk aan $(C_1 + 1) \cdot (C_2 + 1) \cdot 3$.

Hoe gaan we nu in MATLAB te werk? We stellen eerst een nulmatrix op die de grootte van de gezochte generatormatrix heeft. Dit laat ons toe te controleren of alle waarden bepaald zijn en de procestijd in MATLAB te versnellen. Daarna maken we een for-lus waarbij we de aankomsttoestanden overlopen, gaande van één tot vier. In deze lus bepalen we de transities binnen aankomsttoestand i . Daarnaast bepalen we de overgangen naar nieuwe aankomsttoestanden. Zodoende stellen we binnen de i for-lus een bijkomende j for-lus. Deze waarde - gaande van nul tot twee - zal niet gelijk zijn aan de nieuwe aankomsttoestand omdat deze varieert volgens de waarde i . Voor bijvoorbeeld de aankomsttoestand 1 kan deze naar de nieuwe aankomsttoestanden 2, 3 en 4 evolueren, voor aankomsttoestand 2 naar de nieuwe toestanden 1, 3 of 4 enzovoort.

In ons programma zullen we op basis van i en j de eerste, tweede, derde, vijfde en zesde kolom kunnen bepalen. De vierde en zevende kolom zullen bepaald worden door de bijkomende waarde k . Deze stellen we gelijk aan de nieuwe aankomsttoestand. Om deze te kunnen bepalen zullen we gebruik maken van de formule: $k = i + j + 1$. Door een if-stelling daaraan toe te voegen, overlopen we alle mogelijke nieuwe aankomsttoestanden voor een huidige aankomsttoestand i . Het gaat als volgt te werk. Als $k \leq 4$ veranderen we deze waarde niet en is die gelijk aan de nieuwe aankomsttoestand. Echter als $k \geq 4$ dan vervangen we deze waarde door $k - 4$. Op die manier overlopen we alle nieuwe aankomsttoestanden. We geven kort een voorbeeld van de waarden van k wanneer $i = 3$. Tijdens de eerste lus is $j = 0$ zodat $k = 3 + 0 + 1 = 4$. Aangezien het gelijk is aan vier houden we $k = 4$. In de tweede lus, is $j = 1$ zodat $k = 3 + 1 + 1 = 5$. De waarde is hier groter dan de mogelijke aankomsttoestanden zodat we k vervangen door $k = 5 - 4 = 1$. In de laatste doorloop waarbij j gelijk is aan twee, is k gelijk aan zes. We stellen deze gelijk aan twee. Op die manier hebben we alle nieuwe aankomsttoestanden overlopen: toestand 4, 1 en 2.

Nadat de matrix \mathbf{Q} volledig bepaald is, zoals in de twee andere modellen, vervangen we de toestand van het proces tot een ééndimensionale waarde. De formule daarvoor is: $i = (\text{wachttijntoestand} - 1) \cdot (C_1 + 1) \cdot (C_2 + 1) + X_1 \cdot (C_2 + 1) + X_2 + 1$. Op het einde zetten we deze om in een sparse-matrix.