



UHASSELT

KNOWLEDGE IN ACTION

Faculteit Bedrijfseconomische Wetenschappen

master handelsingenieur in de beleidsinformatica

Masterthesis

Een eventlog-based quantile graph methode: inzichten in procesdynamiek

Gythe Stans

Scriptie ingediend tot het behalen van de graad van master handelsingenieur in de beleidsinformatica

PROMOTOR :

Prof. dr. Benoit DEPAIRE



UHASSELT

KNOWLEDGE IN ACTION

www.uhasselt.be

Universiteit Hasselt

Campus Hasselt:

Martelarenlaan 42 | 3500 Hasselt

Campus Diepenbeek:

Agoralaan Gebouw D | 3590 Diepenbeek

2023
2024



Faculteit Bedrijfseconomische Wetenschappen

master handelsingenieur in de beleidsinformatica

Masterthesis

Een eventlog-based quantile graph methode: inzichten in procesdynamiek

Gythe Stans

Scriptie ingediend tot het behalen van de graad van master handelsingenieur in de beleidsinformatica

PROMOTOR :

Prof. dr. Benoit DEPAIRE

Een *eventlog-based quantile graph* methode: inzichten in procesdynamiek

Gythe Stans

Abstract

Deze paper introduceert een innovatief algoritme voor *process mining* dat zich richt op het analyseren van volledige processystemen, in tegenstelling tot traditionele technieken die focussen op individuele procesinstanties. Met de voorgestelde *eventlog-based quantile graph* methode worden waardevolle inzichten verkregen in het gehele processysteem, zoals of een proces geleidelijk evolueert of abrupte veranderingen vertoont. De methode levert zowel lijngrafieken als netwerken op. In de lijngrafieken worden drie belangrijke aspecten besproken voor optimale inzichten. In de netwerken worden drie cruciale structuren geïdentificeerd, die een dieper inzicht bieden in de transities binnen het processysteem. Dit onderzoek toont aan dat het algoritme significant potentieel heeft om de dynamiek van volledige processystemen te onthullen.

Keywords: *Process mining, eventlog, tijdreeks, procesdynamiek, quantile graphs*

1 Inleiding

Process mining richt zich op het analyseren van *end-to-end* processen om *real-life* processen te ontdekken, te monitoren en te verbeteren [1]. Traditionele technieken focussen op afzonderlijke procesinstanties, waardoor inzicht verkregen wordt in de individuele segmenten en minder in het gehele processysteem.

Er is een behoefte aan uitgebreide inzichten in de dynamiek van volledige processystemen om processen te optimaliseren. Verandert het proces geleidelijk aan? Hoe kan een proces geïdentificeerd worden als een al dan niet geleidelijk veranderend proces? Traditionele methoden bieden niet het holistische beeld dat kan helpen bij het beantwoorden van die vragen omdat ze zich richten op individuele procesinstanties. Deze studie vult deze leemte aan door een innovatief algoritme te introduceren dat volledige processystemen analyseert voor een beter begrip van hun toestanden en dynamiek.

Deze paper richt zich op het uitleggen en toepassen van een algoritme voor de analyse van volledige processystemen. De focus ligt op twee hoofdonderwerpen: een uitgebreide bespreking en toelichting van het algoritme met voorbeelden, en de toepassing ervan op een *case study*. Verschillende analysemogelijkheden worden belicht en resultaten geïllustreerd aan de hand van een praktische toepassing.

Deze paper draagt bij aan *process mining* door een methode te presenteren voor het analyseren van volledige processystemen. Het algoritme verbetert het begrip van procesdynamiek en biedt praktische inzichten voor analyse en interpretatie, wat een krachtig hulpmiddel is voor het optimaliseren vanuit een holistisch perspectief.

De paper start met een bespreking van gerelateerd onderzoek, gevolgd door een uitleg van de belangrijkste concepten om een grondig begrip van het algoritme te vergemakkelijken. Dan wordt het algoritme gedetailleerd uitgelegd en toegepast op een *case study*, waarbij de belangrijkste resultaten en analysemogelijkheden worden besproken.

2 Gerelateerd werk

2.1 Procesdynamiek

De stabiliteit en dynamiek van processen werd voorheen in enkele onderzoeken geanalyseerd en besproken. In de paper van Li et al. (2017) wordt een methode ontwikkeld om de flexibiliteit en verandering van processen efficiënt te analyseren en beheren.

Ze doen dit door situaties te identificeren waarin het proces verandert of is veranderd, bekend als *concept drifts*. Eerst worden die situaties gedetecteerd en gelokaliseerd, waarbij traditionele technieken worden toegepast in combinatie met een nieuwe *feature*, voorgesteld in de paper. Daarnaast worden traditionele *process mining*-methoden uitgebreid met nieuwe technieken om de procesmodellen aan te passen aan de gedetecteerde *concept drifts* [2]. De paper focust dus vooral op het lokaliseren van de veranderingen en op het aanpassen van de bestaande procesmodellen aan de hand van die gevonden resultaten. Het doel van deze paper is niet om veranderingen te kunnen lokaliseren, maar om meer inzicht te verkrijgen in het verloop van die veranderingen. Daarnaast spelen *concept drifts* zich af op procesinstantie-niveau, men focust op specifieke, individuele gevallen. Deze paper zal zich daarentegen afspelen op processysteem-niveau. Aangezien de traditionele *process mining*-technieken uit de paper van Li et al. (2017) [2] focussen op procesinstanties, zal in deze paper gebruik gemaakt worden van een andere methode, namelijk de *quantile graph* methode.

In een paper van Takemoto en Arizono (2020) focust men op de stabiliteit van processen. Een stabiel proces blijft binnen bepaalde onder- en bovengrenzen. Indien deze overschreden worden, is het proces niet langer stabiel. De auteurs introduceren een methode gebaseerd op de Bayesiaanse theorie om deze stabiliteit te onderzoeken, waarbij ze kansverdelingen van het procesgemiddelde vergelijken om te bepalen of een proces al dan niet stabiel is. Ze richten zich op de productiesector, specifiek op de overgang naar massaproductie. Hoewel hun onderzoek de stabiliteit over het hele processysteem kan beoordelen, biedt deze geen inzicht in de verdere dynamiek van het proces. Net als deze paper richten Takemoto en Arizono (2020) zich op het volledige processysteem, maar deze paper streeft ernaar verder te gaan door ook inzicht te bieden in de aard van eventuele veranderingen [3].

2.2 *Quantile graph* methode

De methode die wordt toegepast in deze paper om de dynamiek van een processysteem te analyseren, is gebaseerd op de *quantile graph* methode. Die methode werd voorheen reeds toegepast in enkele andere contexten, voornamelijk de gezondheidssector.

In een paper van Pineda et al. (2019) wordt de methode toegepast om de ziekte van Alzheimer te kunnen detecteren. Men maakt gebruik van een tijdreeks die bestaat uit de resultaten van een hersenscan. De resultaten geven de elektrische activiteit in de hersenen weer. De paper toont aan dat de methode zeer goed presteert in het onderscheid tussen individuen met verschillende gezondheidstoestanden [4].

Daarnaast wordt de *quantile graph* methode ook in een andere paper toegepast, eveneens gerelateerd aan de gezondheidssector. Pineda en Rodrigues (2021) passen de *quantile graph* methode toe op de resultaten van een elektrocardiogram, of ECG. Die test registreert de elektrische signalen die de hartspier laten samentrekken. De methode toont in die paper een betrouwbare techniek te zijn om oudere patiënten te onderscheiden van jonge patiënten op basis van de resultaten van hun ECG. Ook wanneer het ECG behoorde tot een ongezonde jonge patiënt, zoals bijvoorbeeld een jonge roker, kan de methode dat onderscheid maken [5].

De paper van Campanharo et al. (2011) richt zich op de beschrijving en gebruik van de methode, met een mindere focus op een specifiek probleem. De methode wordt toegepast in de gezondheidssector, waarbij de hartslag van gezonde en ongezonde mensen wordt vergeleken. Daarnaast wordt de methode ook kort toegepast op de stofwisseling van een specifieke plantensoort en op het internet in 1997 [6].

Deze drie papers delen dus het gemeenschappelijke kenmerk dat het onderzoek zich afspeelt in de gezondheidssector, behalve de derde studie die zich ook kort in het milieu- en onlinegedrag bevindt.

Daarnaast vertrekken deze drie onderzoeken telkens van een beschikbare tijdreeks. Dit is een belangrijk aspect van hun methodologie, aangezien de resultaten van de tests die ze gebruiken, automatisch worden opgeslagen en geïdentificeerd als tijdreeks.

Deze paper toont aan dat de methode niet enkel nuttig kan zijn voor de gezondheidssector en het identificeren van gezondheidsproblemen, maar ook kan helpen bij het analyseren van processen en procesdynamieken. Daarom zal de methode toegepast worden binnen *process mining*. De sector of context zal dus niet alleen vernieuwend zijn, maar ook de manier van toepassing zal veranderen. Aangezien het startpunt van *process mining* bij een *eventlog* ligt, is er dus geen tijdreeks beschikbaar voor een rechtstreekse toepassing van de *quantile graph* methode.

3 Voorbespreking

Om het algoritme te begrijpen, moeten enkele essentiële begrippen worden uitgelegd. Eerst wordt een ***eventlog*** toegelicht, een essentieel concept in *process mining*. Daarna wordt een **tijdreeks** beschreven en geïllustreerd aan de hand van een voorbeeld, omdat de *eventlog* wordt omgezet naar een tijdreeks. Tot slot worden **netwerken**, of netwerkgrafen, uitgelegd.

3.1 *Eventlog*

Process mining richt zich op het analyseren van *end-to-end* processen, met als doel het ontdekken, monitoren en verbeteren van echte processen [1]. Deze techniek biedt inzicht en transparantie in processen, wat vervolgens wordt gebruikt om ze te optimaliseren [7].

Het vertrekpunt van *process mining* zijn de *eventlogs*, die bestaan uit *event* data. Elke rij in een *eventlog* vertegenwoordigt een *event*, of gebeurtenis, in het proces. Zo'n gebeurtenis verwijst naar een *case*, een activiteit en een specifiek tijdstip. De *case* wordt weergegeven aan de hand van een identificatiecode. De activiteit betreft de handeling die op de *case* wordt uitgevoerd, waarbij het tijdstip kan aangeven wanneer de activiteit begon of eindigde, afhankelijk van de *eventlog* [1].

Aan een *event* is een *case* gekoppeld, die ook wel een *process instance* wordt genoemd. Daarnaast kan extra informatie worden opgeslagen in de *eventlog*, zoals de *resource* die de activiteit heeft uitgevoerd, of de kost van de activiteit. Deze aanvullende informatie kan ook worden benut bij *process mining* [1]. Een voorbeeld van een *eventlog* wordt gegeven in Tabel 1.

Tabel 1: Voorbeeld van een *eventlog*.

Case ID	Activiteit	Tijdstip	Resource
1001	Bestelling ontvangen	24/03/2024 11:15:57	Persoon A
1002	Verzenden bestelling	24/03/2024 11:37:23	Persoon B
1003	Kwaliteitscontrole	24/03/2024 12:23:48	Persoon C
1002	Verzenden bestelling	24/03/2024 12:24:01	Persoon B

3.2 Tijdreeks

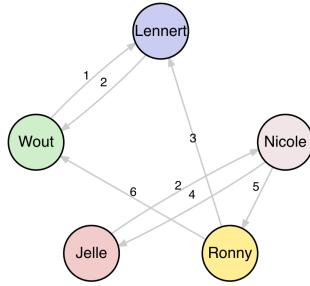
Deze paper richt zich niet alleen op *process mining*, maar ook op tijdreeksanalyse. Een tijdreeks bestaat uit kwantitatieve observaties gerangschikt in chronologische volgorde, zoals te zien is in Tabel 2. Door tijdreeksen te analyseren, kunnen regelmatigheden worden gedetecteerd, patronen worden geïdentificeerd en voorspellingen worden opgesteld voor de toekomst [8].

Tabel 2: Voorbeeld van een tijdreeks.

Tijdstip	Aantal klanten
11:00	15
11:01	26
11:02	7

3.3 Netwerken

Het algoritme maakt gebruik van netwerken, of netwerkgrafen. Een netwerk kan veel betekenissen hebben. Veel realistische systemen kunnen omschreven worden als een netwerk met onderlinge verbindingen, bijvoorbeeld vriendschappen, organisaties of infrastructuren [9]. De *nodes* stellen de centrale punten voor, bijvoorbeeld de mensen in een vriendschap. De *edges* in een netwerk stellen de verbindingen voor, bijvoorbeeld de vriendschap tussen twee mensen of *nodes*. Die *edges* kunnen op verschillende manieren voorgesteld worden. Ze kunnen namelijk enerzijds *directed* of *undirected* zijn, en anderzijds *weighted* of *unweighted* zijn. Een *directed* netwerk bevat verbindingen die de verbanden in een specifieke richting aantonen. Een telefoongesprek van persoon A naar persoon B kan voorgesteld worden als een *directed edge* van A naar B. Wanneer een *edge weighted* is, wordt een numerieke waarde toegekend aan de *edge*, namelijk het gewicht. Een *directed*, en *weighted* netwerk is weergegeven in Figuur 1. Stel dat het netwerk telefoongesprekken voorstelt. De *nodes* geven de personen weer, de *edges* geven weer naar wie gebeld wordt. De waarden bij de *edges* geven weer hoe vaak gebeld wordt. Zo belde Ronny bijvoorbeeld drie keer naar Lennert.

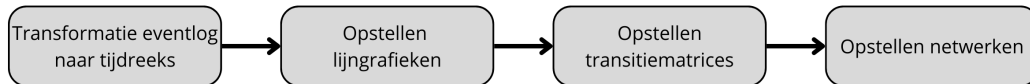


Figuur 1: Het netwerk stelt telefoongesprekken voor. De *nodes* zijn de personen, waarbij de *directed* en *weighted edges* de telefoongesprekken weergeven. Zo belde Lennert bijvoorbeeld twee keer naar Wout.

4 Algoritme

Het algoritme dat in deze paper besproken wordt betreft de werking van een *quantile graph* methode, toegepast op een *eventlog* uit *process mining*. Aan de hand van dit algoritme is het mogelijk om op basis van een *eventlog* inzicht te krijgen in de toestanden en de dynamiek van een processysteem.

Het *high-level* stappenplan wordt weergegeven in Figuur 2. De stappen worden in detail beschreven in het vervolg van deze sectie. De methode begint met het transformeren van de *eventlog* naar een tijdreeks. Daarna wordt de tijdreeks gevisualiseerd met lijngrafieken om het proces te begrijpen. Vervolgens worden verschillende transitiematrices opgesteld, die de basis vormen voor de netwerken. Het algoritme maakt gebruik van zes parameters, die worden beschreven bij de relevante stappen en samengevat in Tabel 16.



Figuur 2: Een *high-level* overzicht van de stappen in de *quantile graph* methode.

Voorbeeld Het algoritme wordt telkens toegelicht aan de hand van een voorbeeld, namelijk een eenvoudige *eventlog* weergegeven in Tabel 3.

Case ID	Activiteit	Tijdstip
1	A	24/03/2024 00:00:00
2	B	24/03/2024 00:11:24
3	C	24/03/2024 00:14:48
2	B	24/03/2024 00:42:01
4	D	25/03/2024 01:01:51
5	A	25/03/2024 01:12:50
1	B	25/03/2024 01:18:03
3	D	25/03/2024 01:31:01

Tabel 3

4.1 Transformatie *eventlog* naar tijdreeks

De eerste stap van het algoritme omvat de transformatie van de *eventlog* naar een tijdreeks, waardoor de toestand op verschillende momenten gemeten kan worden aan de hand van geselecteerde toestandsvariabelen. Deze omzetting omvat verschillende stappen. Eerst wordt de *eventlog* opgedeeld in tijdvensters. Vervolgens worden toestandsvariabelen berekend binnen deze tijdvensters om de toestand op dat moment te bepalen. Ten slotte worden verschillende zones vastgesteld op basis van de waarden van deze toestandsvariabelen. Op deze manier wordt de toestand weergegeven met zowel continue waarden van de toestandsvariabele als discrete zones.

4.1.1 Segmentatie *eventlog* volgens tijdvensters

Eerst wordt de *eventlog* gesegmenteerd volgens tijdvensters. Om toe te laten dat de toestand van het proces op bepaalde momenten gemeten wordt, is het nodig om eerst te bepalen welke momenten dat zijn. Is het bijvoorbeeld nodig om elk uur of elke minuut het proces te meten? Die beslissing beïnvloedt de eerste parameter in het algoritme, namelijk t . Die parameter t definieert de grootte van de tijdvensters, bijvoorbeeld één uur of één minuut. Elk tijdstip in de *eventlog* wordt toegewezen aan een tijdvenster.

Zo wordt de *eventlog* omgevormd naar één lange tijdreeks van alle data. Op basis van de onderzoeksdoelen kan een cyclus worden gedefinieerd, zoals bijvoorbeeld 24 uur, een week, een maand, of op basis van de seizoenen. Deze cyclus bepaalt de opsplitsing van de tijdvensters. Bij een cyclus van 24 uur worden de dagen met elkaar vergeleken, bijvoorbeeld maandag met dinsdag. Door de opsplitsing komt dan bijvoorbeeld het eerste uur op maandag overeen met het eerste uur op dinsdag enzovoort.

Voorbeeld Stel dat t gelijk is aan tien minuten, dan wordt de tijdreeks wordt opgesplitst in tijdvensters van elk tien minuten. Kolom 5 "Tijdvenster" in Tabel 4 geeft weer in welk tijdvenster de rij zich bevindt.

Dan wordt de cyclus bepaald. In dit voorbeeld zal de cyclus 24 uur zijn. De tijdreeks zal gesegmenteerd worden per dag, weergegeven in de zesde kolom.

Case ID	Activiteit	Tijdstip	Tijdvenster	Cyclus
1	A	24/03/2024 00:00:00	1	1
2	B	24/03/2024 00:11:24	2	1
3	C	24/03/2024 00:14:48	2	1
2	B	24/03/2024 00:42:01	5	1
4	D	25/03/2024 01:01:51	151	2
5	A	25/03/2024 01:12:50	152	2
1	B	25/03/2024 01:18:03	152	2
3	D	25/03/2024 01:31:01	154	2

Tabel 4

4.1.2 Berekening toestandsvariabelen

De volgende stap in de transformatie is het berekenen van de toestandsvariabelen. De toestandsvariabelen worden gedefinieerd en vervolgens berekend per tijdvenster.

Een toestandsvariabele geeft een bepaalde meting weer van de toestand van het proces. Het doel van die variabele is om een beter inzicht in het proces of bijvoorbeeld de hoeveelheid werk te krijgen op een bepaald moment. Er zijn veel mogelijke toestandsvariabelen. In deze paper worden er twee besproken, namelijk de bezettingsgraad en het aantal herwerkingen. De bezettingsgraad geeft in percentage weer hoeveel van de beschikbare capaciteit, of *resources*, effectief gebruikt wordt op een bepaald moment. In dit onderzoek wordt de bezettingsgraad berekend op de werknemers. Er wordt een vergelijking gemaakt tussen het aantal werknemers dat in het betreffende tijdvenster een activiteit heeft uitgevoerd, en het aantal werknemers dat in het betreffende tijdvenster beschikbaar is. Een werknemer wordt als beschikbaar geteld indien hij in de halve dag van het relevante tijdvenster een activiteit heeft uitgevoerd. Wanneer een werknemer een activiteit heeft uitgevoerd tussen het tijdstip 00:00:00 en 11:59:59, wordt hij als beschikbaar beschouwd voor alle tijdvensters tussen die uren. Dezelfde redenering geldt voor de werknemers die tussen 12:00:00 en 23:59:59 een activiteit hebben uitgevoerd. Indien de beschikbaarheid van 14 uur tot 15 uur wordt gemeten, wordt de werknemer die om 12 uur of 18 uur een taak heeft uitgevoerd als beschikbaar beschouwd.

Naast de bezettingsgraad, wordt ook het aantal herwerkingen berekend per tijdvenster. Een herwerking betreft een activiteit die meerdere keren op éénzelfde *case* wordt uitgevoerd. Indien een *case* een bepaalde activiteit doorstaat, wordt elke herhaling van de activiteit voor die *case* geteld als een herwerking. Stel dat een bepaalde factuur op maandag is goedgekeurd door de manager. Als de manager op woensdag dezelfde factuur moet goedkeuren, door een foutieve handtekening, is dat een herwerking. Indien die factuur een derde keer goedgekeurd moet worden omdat het document niet correct was, is dat ook een herwerking en zijn dat in totaal twee herwerkingen.

Naast de bezettingsgraad en het aantal herwerkingen kunnen nog veel andere toestandsvariabelen gebruikt worden voor een inzicht te verkrijgen in de toestand van het proces. De keuze voor de toestandsvariabele ligt bij de gebruiker van het algoritme.

Voorbeeld De toestandsvariabele is het aantal *events*. Per tijdvenster wordt geteld hoeveel *events* zich afspelen binnen dat tijdvenster.

Dit resulteert in de tijdreeks in Tabel 5, berekend aan de hand van Tabel 4.

Tijdvenster	Aantal <i>events</i> (toestandsvariabele)
1	1
2	2
5	1
151	1
152	2
154	1

Tabel 5

4.1.3 Bepaling van de zones

Ten slotte worden de zones bepaald. De vorige stappen resulteren in een tijdreeks met continue waarden die de toestand van het proces beschrijven. De zones geven de toestand weer op een discrete manier. Ze worden bepaald op basis van de waarden van de toestandsvariabelen. Elke zone bevat een gelijke hoeveelheid van alle mogelijke waarden. Parameter z , het aantal zones, bepaalt hoeveel procent van de mogelijke data een zone bevat. Indien men kiest voor vijf zones, dan omvat elke zone twintig procent van alle mogelijke waarden. De zones stellen de 5-kwantielen voor.

De onder- en bovengrenzen van de zones worden berekend aan de hand van de waarden. Om de zones te bepalen wordt er gebruik gemaakt van alle mogelijke waarden aanwezig

in de volledige tijdreeks. Op die manier worden algemene zones bepaald zodat deze niet verschillen bij de vergelijkingen. Stel dat de zones bijvoorbeeld per dag bepaald zouden worden, dan wordt de vergelijking tussen de dagen op basis van de overgangen tussen zones vermoeilijkt. Daarom worden alle mogelijke waarden in de tijdreeks individueel gerangschikt naar grootte, om vervolgens te definiëren waar de kwantilen gelegen zijn. Stel dat de waarden in de tijdreeks van één tot en met tien lopen, dan loopt de eerste zone van nul tot en met twee, aangezien dit twintig procent van de waarden omvat. Zone 2 bevat de waarden drie en vier, enzovoort.

Voorbeeld Stel dat er meer data verwerkt is en de resultaten van de vorige stappen leiden tot een tijdreeks met de waarden zoals weergegeven in Tabel 6.

Tijdvenster	1	2	5	151	152	154	165	166	167	180
Aantal <i>events</i>	3	7	10	2	25	12	16	10	20	9

Tabel 6

In dit voorbeeld wordt gekozen voor vijf zones. Elke zone omvat dus 20 procent van de mogelijke waarden. De waarden worden gerangschikt naar grootte om de grenzen te bepalen. Tabel 7 geeft deze stap weer. De tweede kolom geeft weer hoeveel procent van de data zich onder die waarde bevindt. Voor de zonebepaling zijn de waarden nodig waarvan 20 procent, 40 procent, 60 procent, 80 procent en 100 procent van de data kleiner is.

# <i>Events</i>	2	3	7	9	10	10	12	16	20	25
	10%	20%	30%	40%	60%	60%	70%	80%	90%	100%

Tabel 7

Tabel 8 geeft een overzicht weer van de zones en hun grenzen, en een toewijzing aan de tijdvensters.

Zone	Bereik	Tijdvenster	Zone
1	[0;3]	1	1
2	]3;9]	2	2
3	]9;10]	5	3
4	]10;16]	151	1
5	]16;25]	...	...

Tabel 8

4.2 Opstellen lijngrafieken

De volgende stap in het algoritme betreft het opstellen van de lijngrafieken. Eerst worden de waarden van de toestandsvariabelen gesegmenteerd en in de segmenten vervolgens geaggregeerd. Die resultaten worden gevisualiseerd aan de hand van lijngrafieken.

4.2.1 Segmentatie en aggregatie

Bij de transformatie van de *eventlog* naar een tijdreeks wordt een cyclus c bepaald om de tijdreeks te segmenteren. Deze segmentatie groepeerde de tijdreeks in bepaalde periodes om aggregaties mogelijk te maken, zodat verschillende segmenten vergeleken kunnen worden. Bijvoorbeeld, de tijdreeks kan opgesplitst worden in weken om vakantieweken met normale weken te vergelijken.

De volledige tijdreeks wordt dus opgedeeld volgens c . Bijvoorbeeld, bij een cyclus van 24 uur wordt de tijdreeks opgesplitst in 365 groepen als het een jaar betreft. Bij een cyclus van zeven dagen, wordt de tijdreeks opgesplitst in 52 groepen.

Vervolgens worden overeenkomstige waarden uit verschillende groepen geaggregeerd. Stel dat een tijdreeks van 1 januari tot en met 31 december is gesegmenteerd in 365 groepen, namelijk elke dag stelt één groep voor. Indien er bijvoorbeeld vraag is naar een vergelijking van het proces tussen de verschillende weekdays, dan is het nodig om alle waarden van de 52 maandagen samen te nemen om een beeld te krijgen van het proces op maandag in dat jaar. Zo worden de waarden per weekday geaggregeerd, zodat vervolgens de verschillende weekdays vergeleken kunnen worden.

Er zijn verschillende mogelijkheden om de waarden van de groepen te aggregeren. In deze paper worden twee mogelijkheden gebruikt, namelijk de som van de waarden en het gemiddelde van de waarden. Om de som te bekomen van bijvoorbeeld alle maandagen, worden de waarden van het eerste tijdvenster opgeteld. Het resultaat is het totaal van de 52 individuele waarden, één waarde voor elke maandag in dat jaar. Het gemiddelde wordt op dezelfde manier berekend, enkel wordt in dat geval de som gedeeld door 52 om het gemiddelde te bekomen.

Voorbeeld: segmentatie De segmentatie volgens twee verschillende cycli van een tijdreeks van enkele dagen in januari wordt weergegeven in Tabel 9. Enerzijds wordt de tijdreeks opgesplitst volgens een cyclus van 24 uur, en anderzijds volgens een cyclus van zeven dagen.

Dag	Aantal <i>events</i>	Cyclus: 24 uur	Cyclus: 7 dagen
Maandag 1 januari	6	1	1
Dinsdag 2 januari	9	2	1
Woensdag 3 januari	15	3	1
Maandag 8 januari	7	8	2
Dinsdag 9 januari	3	9	2
Woensdag 10 januari	2	10	2

Tabel 9

Voorbeeld: aggregatie De aggregatie van de twee verschillende cycli wordt weergegeven in Tabel 10. Voor de cyclus van 24 uur wordt gekozen voor de som van de segmenten, op basis van de weekdays. Voor de cyclus van zeven dagen wordt gekozen voor het gemiddelde, op basis van de weeknummers.

Dag	Som	Week	Gemiddelde
Maandag	13	1	10
Dinsdag	12	2	4
Woensdag	17		

Tabel 10

4.2.2 Visualisatie

Op basis van de segmentatie en aggregatie in de vorige stap, wordt de lijngrafiek gevisualiseerd. Op welke manier die lijngrafiek kan bijdragen aan het verwerven van inzichten, en naar welke punten gekeken moeten worden om die inzichten optimaal te bekomen wordt besproken in het onderdeel [5.1 Lijngrafieken](#).

De lijngrafiek wordt opgesteld aan de hand van twee variabelen in de tijdreeks. De x-as is gebaseerd op het tijdvenster, de y-as geeft de geaggregeerde waarden in dat tijdvenster weer van de betreffende toestandsvariabele. De lijngrafiek zal dus bijvoorbeeld de gemiddelde bezettingsgraad weergeven op bepaalde tijdstippen op maandag.

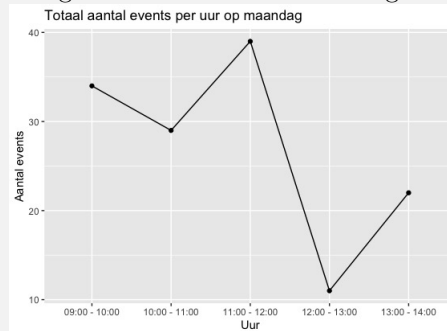
De lijngrafiek geeft een beeld van het verloop van het proces en de toestand van het proces op bepaalde momenten. Stel dat een lijngrafiek wordt opgesteld voor alle verschillende weekdays in een week, dan kan de lijngrafiek helpen bij de vergelijking tussen het proces op een maandag en op een dinsdag. Specifieke analysemogelijkheden worden opgesomd in [5.1 Lijngrafieken](#).

Voorbeeld Stel dat men na de segmentatie en aggregatie van een *eventlog* de waarden in Tabel 11 bekomt. De som geeft in dit geval weer dat voor elke individuele maandag in de *eventlog* het aantal *events* is berekend per tijdvenster, en vervolgens die verschillende waarden zijn opgeteld per tijdvenster.

Dag	Tijdvenster	Som aantal <i>events</i>
Maandag	09:00 - 10:00	34
Maandag	10:00 - 11:00	29
Maandag	11:00 - 12:00	39
Maandag	12:00 - 13:00	11
Maandag	13:00 - 14:00	22

Tabel 11

De lijngrafiek in Figuur 3 geeft weer hoe die data zou gevisualiseerd worden.



Figuur 3: Voorbeeldvisualisatie van de lijngrafiek.

4.3 Opstellen transitie matrices

Op basis van de tijdreeks, de bijhorende waarden van de toestandsvariabelen en de bepaalde zones, worden er verschillende transitie matrices opgesteld. Eerst worden die matrices opgesteld op basis van de parameter k , of het aantal sprongen tussen twee tijdvensters. Vervolgens worden de matrices getransformeerd. Er wordt een grens lim_1 bepaald, waarbij frequenties onder die grens verwijderd worden. Daarnaast worden de waarden omgevormd naar conditionele kansen om de interpretatie te vergemakkelijken.

4.3.1 Opstelling transitie matrices op basis van k

Bij het opstellen van de transitie matrices moet eerst de parameter k bepaald worden. Die k geeft weer welke transitie geanalyseerd wordt, bijvoorbeeld een éénstaptransitie of bijvoorbeeld een tweestaptransitie. Een éénstaptransitie geeft de transitie weer tussen aangrenzende tijdvensters, een tweestaptransitie slaat één tijdvenster over, enzovoort. Een éénstaptransitie wil zeggen dat men spreekt over twee aangrenzende tijdvensters, er is slechts één sprong nodig van het eerste tijdvenster naar het volgende tijdvenster. Een tweestaptransitie betekent twee stappen of sprongen. Er wordt dus één tijdvenster overgelaten. Stel dat de tijdvensters opgedeeld zijn in uren, waarbij één tijdvenster dus van negen tot tien uur loopt. Het volgende venster loopt van tien uur tot elf uur enzovoort. Bij een éénstaptransitie kijkt men naar de transitie tussen negen uur tot tien uur, en tien uur tot elf uur. Bij een tweestaptransitie gaat het over het venster van negen uur tot tien uur, en het venster van elf uur tot twaalf uur. Een kleine k -waarde betekent dus dat er gekeken wordt naar de verandering in toestand op korte tijd. Hoe groter k wordt, hoe groter de tijd tussen de transities. Bij een grote k -waarde kijkt men naar transities op lange termijn.

Vervolgens moet aan de hand van k bepaald worden welke transitie gebeurt. Zo moet bijvoorbeeld bij $k = 1$, of een éénstaptransitie, berekend worden in welke zone het volgende tijdvenster zich bevindt.

Aan de hand van die resultaten kan de transitie matrix opgesteld worden. In die matrix geven de rijen en de kolommen de verschillende zones weer. De rij geeft de zone weer van

het huidige tijdvenster, de kolom geeft de zone weer van het tijdvenster over k sprongen. Bijvoorbeeld, indien de waarde in de tweede rij en de vijfde kolom zeven is, dan is de transitie van zone 2 naar zone 5 zeven keer voorgekomen in de tijdreeks.

Voorbeeld: k Stel dat men op basis van meer data Tabel 6 heeft kunnen uitbreiden naar Tabel 12. De zones uit Tabel 8 werden toegepast. De vierde kolom geeft de éénstaptransitie weer, de vijfde kolom geeft de tweestaptransitie weer.

Tijdvenster	Aantal events	Zone	$k = 1$	$k = 2$
1	3	1	2	5
2	7	2	5	4
3	20	5	4	3
4	12	4	3	1
5	10	3	1	5
151	2	1	5	1
152	25	5	1	...
153	2	1	...	...
...	...	...	...	...

Tabel 12

Voorbeeld: transitiematrix Op basis van Tabel 12 wordt een transitiematrix opgesteld voor $k = 1$. De matrix geeft weer dat bijvoorbeeld één transitie van zone 1 naar zone 2 is geteld in Tabel 12.

Zone	1	2	3	4	5
1	0	1	0	0	1
2	0	0	0	0	1
3	1	0	0	0	0
4	0	0	1	0	0
5	1	0	0	1	1

Tabel 13

4.3.2 Transformatie transitiematrices

Daarna moeten twee transformaties worden uitgevoerd op de transitiematrices.

De eerste transformatie betreft het instellen van de ondergrens lim_1 voor de transitiefrequenties. Dit betekent dat een minimumaantal observaties voor een transitie vereist is om deze in de analyse op te nemen. Frequenties onder lim_1 worden op nul gezet. Deze parameter bepaalt het detailniveau van het onderzoek. Als elke transitie moet worden geanalyseerd, kan lim_1 op nul worden gezet. Indien het slechts nodig is dat een frequentie in één van de vier gevallen voorkomt, kan lim_1 op 25 procent worden gezet.

De tweede transformatie betreft het omzetten van absolute frequenties naar conditionele kansen. Voor elke rij, die een zone in het huidige tijdvenster voorstelt, worden de conditionele kansen voor de transities naar andere zones berekend. De som van een rij is gelijk aan één. Elke waarde geeft de kans weer op een transitie naar de zone in de betreffende kolom. Bijvoorbeeld, als de tweede rij en vijfde kolom een waarde van zeven procent hebben, is de kans op een transitie van zone 2 naar zone 5 zeven procent. Bij honderd transities vanuit zone 2 zullen er zeven naar zone 5 zijn.

Voorbeeld Aangezien er meer data nodig is voor het toepassen van een limiet, wordt voor deze stap gebruik gemaakt van een willekeurige transitie matrix, weergegeven in Tabel 14.

Zone	1	2	3	4	5
1	90	100	70	50	10
2	100	90	50	70	10
3	70	100	10	90	50
4	10	90	50	70	100
5	2	10	90	50	70

Tabel 14

Twee transformaties worden uitgevoerd. Enerzijds wordt de ondergrens, lim_1 toegepast, anderzijds worden de waarden omgerekend naar conditionele kansen. De eerste matrix in Tabel 15 geeft het toepassen van $lim_1 = 10$ procent weer. De tweede matrix geeft de afgeronde conditionele kansen weer. Indien de waarde zich op een bepaald moment in zone 1 bevindt, heeft een transitie naar de tweede zone 32 procent kans, een lichte stijging in de waarde.

Zone	1	2	3	4	5	Zone	1	2	3	4	5
1	90	100	70	50	0	1	29%	32%	23%	16%	0%
2	100	90	50	70	0	2	32%	29%	16%	23%	0%
3	70	100	0	90	50	3	23%	32%	0%	29%	16%
4	0	90	50	70	100	4	0%	29%	16%	23%	32%
5	0	0	90	50	70	5	0%	0%	43%	24%	33%

Tabel 15

4.4 Opstellen netwerken

Zoals de lijngrafieken een visueel beeld vormen van de segmentatie en aggregatie van de tijdreeksen, zijn de netwerken een visualisatie van de transitie matrices. De lijngrafieken vertellen meer over het verloop van het proces, terwijl de netwerken meer informatie bieden over de toestanden in het proces, en de transities tussen de toestanden.

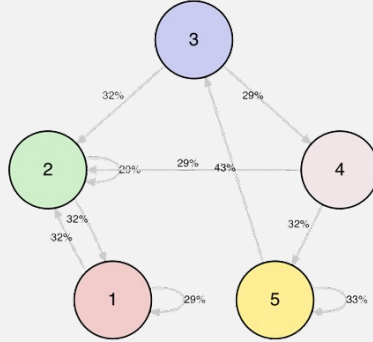
De netwerken geven weer hoe de toestand van het proces verandert. Daarin worden de zones weergegeven, met daarbij de kans op een transitie naar een andere zone. Een netwerk bevat **nodes** en **edges**. Het onderdeel 5.2 Netwerken biedt meer informatie over de analyse mogelijkheden van de netwerken, en op welke manier de netwerken optimaal gebruikt kunnen worden bij het verwerven van inzichten in de dynamiek van het processysteem.

De **nodes** in het netwerk stellen de verschillende zones voor. Stel dat men kiest voor vijf zones, de 5-kwantielen, dan zijn vijf *nodes* aanwezig in het netwerk. Die *nodes* zijn de discrete toestanden op basis van de toestandsvariabelen. Stel dat de tijdreeksen elke waarde tussen één en tien exact één keer bevat, en gekozen is voor vijf zones, dan stelt de eerste *node* de eerste zone voor, met de waarden één tot en met twee.

De **edges** in het netwerk stellen de transities voor uit de transitie matrix. Één rij in de transitie matrix stelt de verschillende uitgaande *edges* voor bij de zone van die rij. Bijvoorbeeld, de eerste rij in de transitie matrix heeft de waarden 0,1; 0,2; 0,3 en 0,4. Er is in dat geval tien procent kans op een transitie van zone 1 naar zone 1. De *edge* van zone 1 naar zone 1, een *selfloop*, zal dus de waarde 0,1 krijgen. De *edge* van zone 1 naar zone 2 zal de waarde 0,2 krijgen enzovoort. De som van de waarden van de uitgaande *edges* van een bepaalde zone bedraagt één.

Voor de *edges* wordt eventueel gebruik gemaakt van een ondergrens, lim_2 . Die ondergrens kan helpen bij het duidelijker visualiseren van de netwerken. Opnieuw is die ondergrens afhankelijk van de nood aan detail in het onderzoek. Indien men elke transitie wil bekijken, dan kan de ondergrens de waarde nul aannemen. Maar indien men bijvoorbeeld enkel de transities wil bekijken die minstens één keer voorkomen in tien gevallen, dan kan lim_2 de waarde 0.1 of tien procent aannemen.

Voorbeeld Op basis van de getransformeerde transitie matrix in Tabel 15 kan een netwerk opgesteld worden. Als voorbeeld wordt gekozen voor een minimumgrens lim_2 van 25 procent. Figuur 4 geeft het netwerk weer. De *nodes* stellen de zones voor, de *edges* stellen de kans op de transitie voor. Zo wordt duidelijk dat bij een waarde in zone 1, er 32 procent kans is op een transitie naar zone 2.



Figuur 4: Voorbeeldvisualisatie van een netwerk.

5 Case study

Om de methode te demonstreren en aan te tonen hoe de methode gebruikt kan worden om inzichten in een proces te verwerven, werd een *case study* uitgevoerd. In dit onderdeel wordt aangehaald hoe de methode toegepast kan worden op een *eventlog* en welke analyses mogelijk zijn op de resultaten.

Bij deze methode kunnen twee soorten grafieken geanalyseerd worden. Enerzijds kunnen de lijngrafieken geanalyseerd worden, en anderzijds de netwerken. In beide grafieken zijn er enkele punten waarop gelet kan worden bij het helpen van inzicht verwerven in het proces. Die verschillende punten worden aangehaald in deze sectie. Daarnaast wordt de methode toegepast op een dataset om een voorbeeld te schetsen. Om de methode te kunnen toepassen, worden de waarden voor de verschillende parameters bepaald. De parameters, hun notatie en de waarden in de *case study* worden weergegeven in Tabel 16.

Tabel 16: Parameters en waarden in de *case study*

Parameter	Eenheid	Notatie	Waarde in <i>case study</i> .
Grootte tijdvenster	Tijd	t	10 minuten
Cyclus	Tijd	c	24 uur
Aantal zones	Integer	z	5
Aantal sprongen	Integer	k	1, 2, 3
Minimumgrens transitie matrix	Procent	lim_1	10%
Minimumgrens <i>edges</i>	Procent	lim_2	5%

De gebruikte dataset is de *eventlog* van de *BPI Challenge* in 2012. Die *eventlog* bevat het proces van een financiële instelling die de aanvragen behandelt van een persoonlijke lening of rekening-courantkrediet binnen een wereldwijde financieringsorganisatie. De data omvat enkele variabelen, waarvan de belangrijkste voor dit onderzoek de volgende zijn: de identificatiecode van de *case*, de identificatiecode van de activiteit, de identificatiecode van de *resource* en het tijdstip. Dat tijdstip stelt het startpunt van de activiteit voor [10].

De data in de *eventlog* is verdeeld volgens de waarden in Tabel 17. Uit die waarden wordt duidelijk dat de hoeveelheid data doorheen de week daalt. Zo bevat elke dag minder data dan de dag ervoor. Enkel donderdag wijkt af van deze trend en bevat ongeveer even veel data als woensdag. Daarbij wordt ook duidelijk dat zondag in vergelijking met de andere dagen maar een zeer klein deel van de data vertegenwoordigt, namelijk slechts 1.67 procent. Aangezien zondag dus zeer weinig observaties bevat, zal die data buiten

beschouwing gelaten worden in de analyses. In de analyse wordt de vergelijking gemaakt tussen de samengenomen weekdays en zaterdag. Er werd besloten om de gegevens van de weekdays samen te nemen voor de analyse aangezien die dagen vaak zeer gelijke patronen en structuren vertonen. Die gegevens worden geanalyseerd ten opzichte van de gegevens van de zaterdagen.

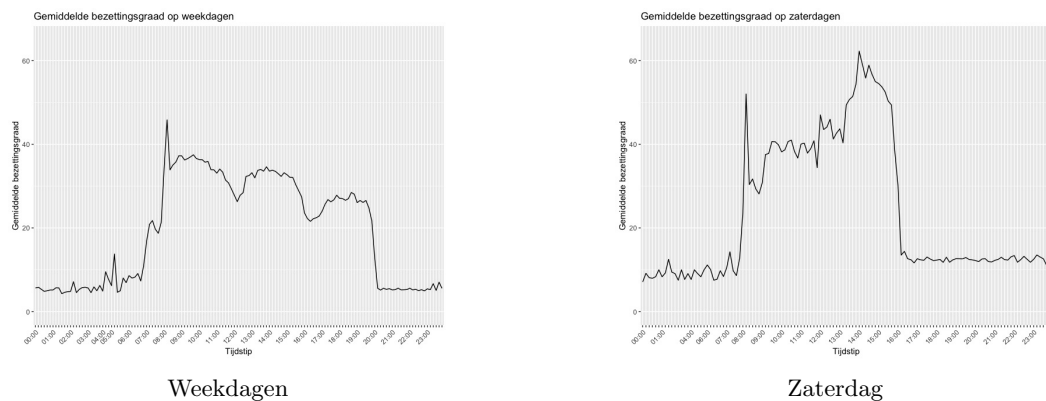
Voor elk onderdeel wordt het onderwerp eerst algemeen besproken en vervolgens toegelicht aan de hand van een toepassing op de dataset.

Tabel 17: Aantal *events* per weekday.

Weekdag	Aantal <i>events</i>	Relatieve frequentie
Maandag	54.774	20.89%
Dinsdag	49.091	18.72%
Woensdag	46.046	17.56%
Donderdag	46.721	17.82%
Vrijdag	40.584	15.48%
Zaterdag	20.614	7.86%
Zondag	4.370	1.67%

5.1 Lijngrafieken

Het toepassen van de methode leidt tot enkele lijngrafieken die de evolutie in het proces weergeven. Die lijngrafieken kunnen tot verschillende inzichten leiden in het proces. Om die inzichten te verwerven zijn er drie belangrijke punten om naar te kijken, namelijk de **startfase**, de **trends** en de **perioden** in de lijngrafiek. Bij het toepassen van deze drie onderdelen wordt er gebruik gemaakt van de lijngrafieken in Figuur 5.



Figuur 5: De gemiddelde bezettingsgraad op de weekdays en op zaterdag.

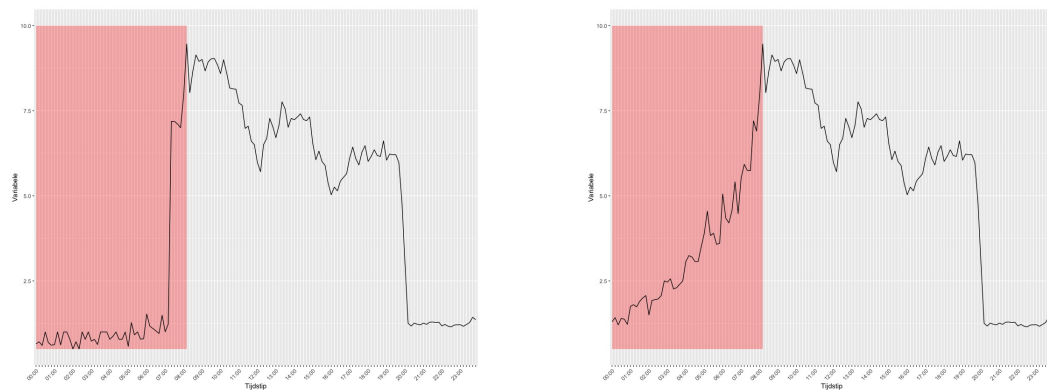
5.1.1 Startfase

Algemeen

Bij het analyseren van het proces biedt de startfase een eerste inzicht, waarbij de start van een werkdag wordt weergegeven, inclusief de opstart van het werk en de werknemers. Er worden twee soorten startfasen onderscheiden: een plotse start en een geleidelijke start. Deze twee soorten worden weergegeven in Figuur 6.

Een plotse start in een proces betekent dat er plotseling veel werk binnenkomt op een specifiek moment. Bijvoorbeeld, als een kledingwinkel om negen uur opent en er op dat moment een grote groep klanten in de rij staat, dan ervaart de winkel een plotse startfase doordat het werk plotseling sterk toeneemt door de klanten bij opening.

Een plotse startfase in een lijngrafiek wordt gekenmerkt door een snelle, abrupte stijging van de waarden binnen een korte tijd. Eerst blijft de lijngrafiek relatief stabiel met kleine veranderingen, maar plotseling is er een aanzienlijke toename die leidt tot een grote sprong tussen twee tijdstippen, of een bijna verticale lijn op de grafiek.



Plotse startfase

Geleidelijke startfase

Figuur 6: De twee lijngrafieken geven de twee soorten startfasen weer.

Naast een plotse start kan een proces ook een geleidelijke start kennen. Dit houdt in dat het werk niet abrupt, maar geleidelijk toeneemt over een langere periode. Bijvoorbeeld, in een kledingwinkel die om negen uur opent, kan in het eerste uur één klant langskomen, in het tweede uur twee klanten, in het derde uur vier, enzovoort. De hoeveelheid werk neemt dus gestaag toe tot een hoogtepunt. In het geval van een plotse start kan dit hoogtepunt in het eerste openingsuur zijn, terwijl bij een geleidelijke start dit bijvoorbeeld pas later plaatsvindt.

In tegenstelling tot een plotse start, vertoont een geleidelijke start geen abrupte, verticale lijn in de grafiek. De geleidelijke start laat een geleidelijk stijgende lijn zien vanaf het startpunt tot het begin van de activiteitenfase. In plaats van een specifiek punt van stijging, zoals bij een plotse stijging, laat de startfase over het algemeen een geleidelijke opwaartse trend zien.

Toepassing

Zowel de weekdays als zaterdag vertonen in 5 eerder een plotse start. Zaterdag schiet op een korte tijd van een bezettingsgraad rond 10 procent, naar één van meer dan 50 procent. Dit is slechts een klein verschil met het maximum van ongeveer 60 procent. De weekdays vertonen een iets geleidelijkere stijging. De opstart op zaterdag levert dus een grotere hoeveelheid werk op dan op een weekday.

5.1.2 Trends

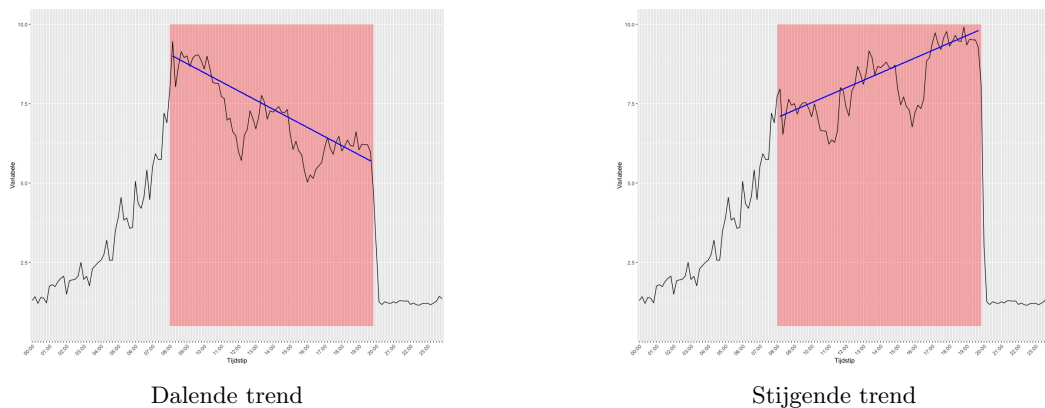
Algemeen

Naast de startfase kan ook de activiteitenfase, die volgt op de startfase, belangrijke inzichten bieden. De trend in de activiteitenfase kan dalend, stijgend of stabiel zijn. Voorbeelden van een dalende en stijgende trend worden getoond in Figuur 7.

Een **dalende** trend in de activiteitenfase, gemarkeerd als de rode zone in Fig 7, betekent dat de gemiddelde werklast gedurende de dag afneemt, met meer werk aan het begin dan aan het einde van de werkdag. Bijvoorbeeld, een koffiezaak zal doorgaans drukker zijn tijdens de ochtenduren wanneer mensen koffie halen, maar minder druk naar het einde van de dag toe.

Bij een dalende trend is de belangrijkste indicator dat de trendlijn, zoals de blauwe lijn in Figuur 7, een dalende lijn volgt. Hoewel er stijgingen en stabiele periodes kunnen voorkomen, blijft de algemene trend neerwaarts. Ondanks de pieken en plateaus, neemt de hoeveelheid werk af over het algemeen en keert niet terug naar het hoge niveau waar de activiteitenfase mee begon.

Een **stijgende** trend in de activiteitenfase duidt op toenemende werklast gedurende de dag, met meer werk aan het einde dan aan het begin. Bijvoorbeeld, in een restaurant zullen



Figuur 7: Een weergave van een dalende of stijgende trend in de activiteitenfase.

er aanvankelijk weinig klanten zijn, maar neemt het aantal toe bij de lunch en piekt het tijdens het avondeten.

Een stijgend proces kan verschillende vormen aannemen, maar de trendlijn in Figuur 7 bevestigt of er sprake is van een stijgende trend. Hoewel er fluctuaties en zelfs afnames kunnen zijn in de activiteitenfase (de rode zone), toont de trendlijn (de blauwe lijn) een duidelijke stijging in de hoeveelheid werk gedurende de dag.

Een **stabiel** proces betekent dat de hoeveelheid werk constant blijft gedurende de dag, zonder een dalende of stijgende trend. Bijvoorbeeld, een tandartsenpraktijk kan gedurende de dag een constante werklast hebben, omdat elke afspraak één patiënt betreft en de agenda meestal volgeboekt is.

Een stabiel proces kan fluctuaties vertonen, maar wordt gekenmerkt door een horizontale trendlijn in Figuur 7. De hoeveelheid werk blijft gedurende de dag gelijk.

Toepassing

Beide activiteitenfasen in Figuur 5 vertonen een duidelijke trend. In de activiteitenfase van de weekdays daalt de hoeveelheid werk duidelijk, aangezien een dalende trend op te merken is. Het tweede lokale maximum ligt lager dan het eerste lokale maximum. Het derde lokale maximum ligt vervolgens opnieuw lager. Op zaterdag is een stijgende trend aanwezig. Dat wil dus zeggen dat de hoeveelheid werk op een zaterdag stijgt doorheen de dag, terwijl de hoeveelheid op een weekday net daalt naarmate het einde van de dag nadert.

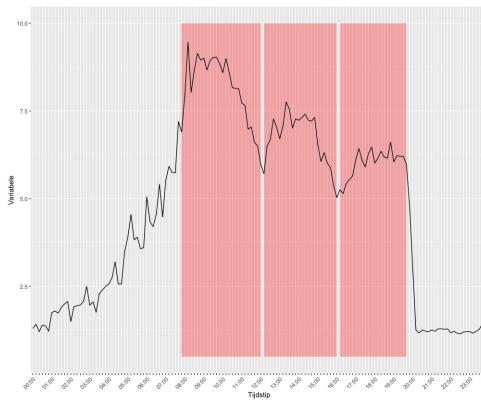
5.1.3 Perioden

Algemeen

Naast trends kunnen ook perioden worden geïdentificeerd in de activiteitenfase, die herhalingen of patronen aangeven. Deze perioden worden vastgesteld door terugkerende patronen te herkennen op basis van tijdvensters in de grafiek. Hoewel er veel verschillende soorten herhalingen kunnen zijn, zijn er geen vaste categorieën om op te letten. Afhankelijk van het proces kan het echter handig zijn om te letten op bepaalde shiftindelingen. Bijvoorbeeld, patronen kunnen zich voordoen per shift of tussen elke pauze. Om deze herhalingen te identificeren, is het nuttig om te kijken naar deze tijdstippen. Bijvoorbeeld, als de werkdag van acht uur 's ochtends tot vier uur 's middags loopt, met een pauze na twee uur werk, dan zou het zinvol zijn om te zoeken naar herhalingen tussen acht en tien uur, tien en twaalf uur, twaalf en twee uur, en twee en vier uur. Figuur 8 illustreert een voorbeeld van perioden in een proces.

Toepassing

Een zeer duidelijk voorbeeld van perioden is duidelijk in de lijngrafiek van de gemiddelde bezettingsgraad van de weekdays, in Figuur 5. Net zoals in het voorbeeld in Figuur 8 is een soort bergstructuur op te merken. Er zijn twee dalen op te merken rond half één en



Figuur 8: Een voorbeeld van perioden in een proces. De drie rode zones geven éénzelfde patroon weer, namelijk een soort bergstructuur. De dalpunten tussen de herhalingen zijn ongeveer om twaalf uur 's middags en vier uur in de namiddag, wat bijvoorbeeld kan wijzen op de shifts.

half zeven. Deze momenten kunnen wijzen op bijvoorbeeld pauzes, het einde van een shift of het wisselen van shifts.

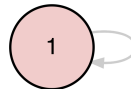
5.2 Netwerken

Naast de lijngrafieken, kunnen ook structuren in netwerken inzicht bieden in het proces. Drie identificeerbare structuren zijn **selfloops**, de **cirkelstructuur** en de **sterstructuur**. *Selfloops* geven stabiliteit aan, terwijl cirkel- en sterstructuren veranderingen in het proces aanduiden.

5.2.1 Selfloops

Algemeen

Selfloops duiden op stabiliteit in het procesnetwerk, waarbij geen opmerkelijke veranderingen optreden in de waarden. De hoeveelheid werk blijft consistent en grote fluctuaties komen zelden voor. Deze *selfloops* zijn te herkennen als *directed edges* die van een zone naar dezelfde zone terugkeren, waardoor een lus ontstaat. Figuur 9 toont een voorbeeld van een *selfloop*. Bijvoorbeeld, indien zeven klanten aanwezig zijn in een winkel en het bereik van zone 1 van nul klanten tot tien klanten gaat, dan zal in het volgende tijdvenster het aantal klanten opnieuw lager dan tien zijn.



Figuur 9: Een voorbeeld van een *selfloop* bij zone 1.

Toepassing

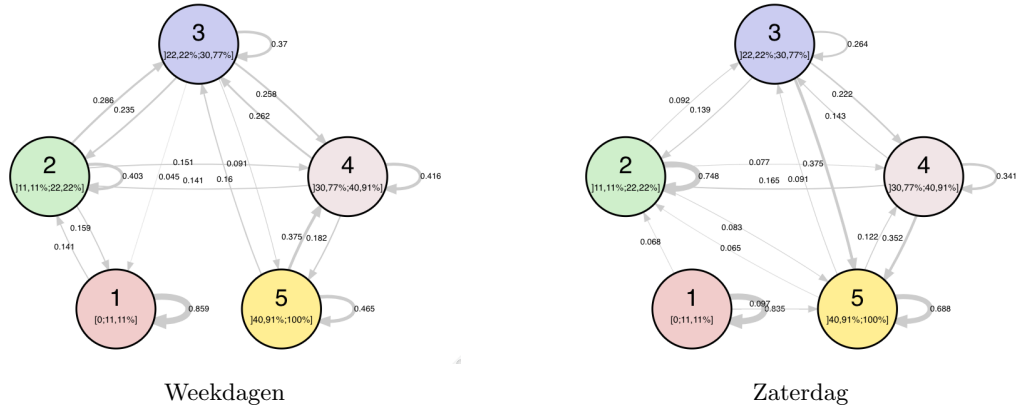
De *selfloops* in de netwerken in Figuur 10 bieden inzicht in de procesdynamiek, waarbij de zones en hun bereik op weekdays en op zaterdag worden weergegeven. De k -waarde voor deze netwerken is één.

Zone 1 vertoont op beide dagen een sterke *selfloop*, ongeveer 86 procent op weekdays en 84 procent op zaterdag. Dit betekent dat indien de bezettingsgraad lager is dan 11,11 procent, de kans groot is dat deze na tien minuten nog steeds laag zal zijn. Het proces neigt een lage bezettingsgraad te behouden.

Zone 2 en 5 vertonen op zaterdag ook sterke *selfloops*, terwijl die van zone 3 en 4 lager zijn dan op weekdays. Dit impliceert dat op zaterdag het proces eerder geneigd is om éénzelfde bezettingsgraad te behouden, zowel bij zeer lage als zeer hoge niveaus. Bijvoorbeeld, als de bezettingsgraad 15 procent is, is op zaterdag de kans groter dat deze na 10 minuten vergelijkbaar blijft, net zoals bij bijvoorbeeld 50 procent.

De *selfloop* van zone 3 bedraagt op zaterdag slechts een waarde van ongeveer 26 procent. Dit betekent dat de bezettingsgraad in slechts één op de vier gevallen hetzelfde blijft na 10 minuten als deze tussen 22,22 procent en 30,77 procent ligt. Op zaterdag blijft de

bezettingsgraad dus niet lang in het midden hangen, maar zal eerder stijgen naar één van de hogere of lagere niveaus.



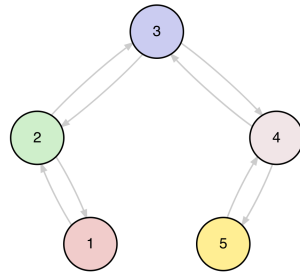
Figuur 10: De netwerken van de gemiddelde bezettingsgraad op weekdagen en op zaterdag, met een k -waarde gelijk aan 1.

5.2.2 Cirkelstructuur

Algemeen

Naast *selfloops* kan ook een cirkelstructuur in een netwerk worden waargenomen. In tegenstelling tot *selfloops* geeft de cirkelstructuur aan dat het proces niet stabiel blijft en de waarden stijgen of dalen naar aangrenzende zones. Veranderingen gebeuren geleidelijk en slaan geen zones over. Bijvoorbeeld, als elke zone telkens tien klanten meer bevat dan de vorige zone, zal een cirkelstructuur betekenen dat er nooit meer dan tien klanten bijkomen of weggaan in het volgende tijdvenster.

De cirkelstructuur kan worden geïdentificeerd door de geleidelijke vormgeving waarbij overgangen de omtrek van het netwerk volgen. Figuur 11 toont een voorbeeld. Belangrijk is op te merken dat overgangen tussen zone 1 en zone 5 niet deel uitmaken van de cirkelstructuur, ondanks het toebehoren aan de omtrek van het netwerk. In die overgang gaat het proces van één van de laagste hoeveelheden werk naar één van de grootste hoeveelheden werk, wat de interpretatie van de cirkelstructuur zou tegenspreken op basis van de geleidelijke veranderingen.



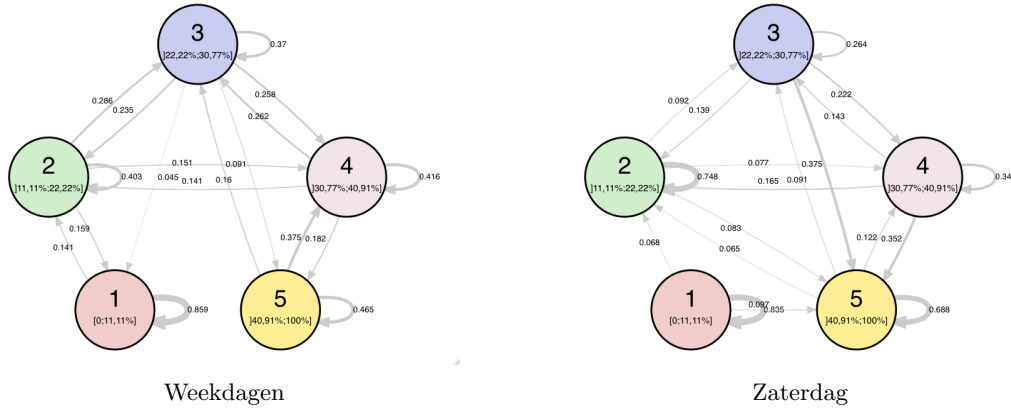
Figuur 11: Een voorbeeld van de cirkelstructuur in een netwerk. De cirkelstructuur betreft de *directed edges* tussen aangrenzende zones. De overgang tussen zone 1 en zone 5 maken geen deel uit van de cirkelstructuur.

Toepassing

De netwerken in Figuur 12 worden gebruikt als voorbeeld. De cirkelstructuur is duidelijk sterker aanwezig is op weekdagen dan op zaterdag, zoals te zien is aan het aantal verbindingen in de omtrek van elk netwerk. Zo valt de kans op een transitie van zone 2 naar zone 1 zelfs onder de minimumgrens van vijf procent (lim_2) op zaterdag, omdat de bijbehorende *directed edge* ontbreekt.

Bovendien zijn de kansen van de aanwezige transities in de cirkelstructuur op weekdays vaak hoger dan die op zaterdag, behalve de transitie van zone 4 naar zone 5. Over het algemeen is de kans op een transitie in de cirkelstructuur op een weekday ongeveer twee keer zo groot als die op zaterdag, behalve voor de transitie van zone 4 naar zone 5, waar de kans op een weekday slechts de helft is van die op zaterdag.

Dit suggereert dat het proces op weekdays doorgaans geleidelijker verloopt dan op zaterdag. Op zaterdag blijft de bezettingsgraad vaak stabiel, en wanneer dit niet het geval is, zijn de veranderingen doorgaans groter dan op weekdays.



Figuur 12: De netwerken van de gemiddelde bezettingsgraad op weekdays en zaterdag, met $k = 1$. De cirkelstructuur is op weekdays meer aanwezig dan op zaterdag.

5.2.3 Sterstructuur

Algemeen

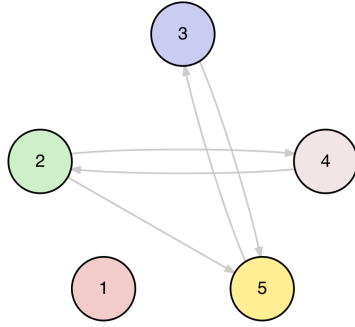
Een sterstructuur in het netwerk duidt op plotselinge grote wijzigingen in het proces, in tegenstelling tot de stabiele *selfloops* en de geleidelijke veranderingen van de cirkelstructuur. Deze wijzigingen gaan naar niet-aangrenzende zones, wat aangeeft dat de waarde op significante wijze stijgt of daalt en aangrenzende zones worden overgeslagen. Bijvoorbeeld, als elke zone telkens tien klanten meer bevat, zal een sterstructuur betekenen dat als er vijf klanten aanwezig zijn in zone 1, er in het volgende tijdvenster minstens 21 klanten aanwezig zullen zijn, omdat waarden tussen elf en twintig worden overgeslagen.

Bij het analyseren van een proces met een sterstructuur kunnen twee punten worden overwogen voor verdere inzichten: het **niveau** van de sterstructuur en het **aantal sprongen** dat nodig is om deze te veroorzaken.

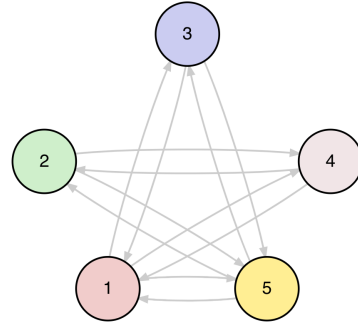
Een niveau kan worden toegewezen aan een sterstructuur, variërend van 0,5 tot 6. Dit niveau geeft de sterkte van de sterstructuur in het netwerk aan. Bijvoorbeeld, een niveau van 0 betekent dat er geen sterstructuur aanwezig is, terwijl een niveau van 6 aangeeft dat het netwerk volledig verbonden is. Het niveau wordt bepaald door het aantal *directed edges* tussen niet-aangrenzende zones: elke *directed edge* voegt 0,5 toe aan het niveau.

Bijvoorbeeld, als er slechts één *directed edge* is tussen zone 1 en zone 3, heeft de sterstructuur een niveau van 0,5. Hoe hoger het niveau, hoe gemakkelijker het proces zones overslaat en abrupt verandert. Een belangrijk punt om op te merken is dat overgangen tussen zone 1 en zone 5 deel uitmaken van de sterstructuur, ondanks dat ze visueel aangrenzende zones lijken. Dit is gebaseerd op de bereiken van de zones, niet op hun visuele nabijheid. Figuur 13 toont twee voorbeelden van sterstructuurniveaus.

Naast het niveau van de sterstructuur is het **aantal sprongen** waarbij deze verschijnt ook relevant. Een hoger aantal sprongen duidt op geleidelijkere wijzigingen. Bijvoorbeeld, als een proces een sterstructuur vertoont na drie sprongen, verandert het geleidelijker dan



Sterstructuur van niveau 2,5



Sterstructuur van niveau 6

Figuur 13: Twee voorbeelden van sterstructuren en hun niveau.

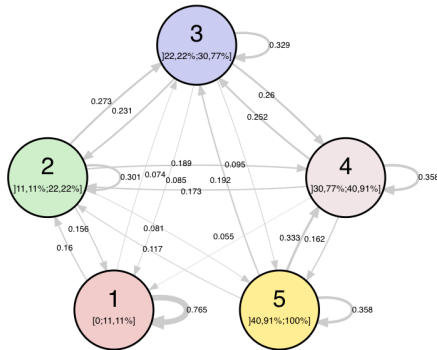
een proces waarbij de sterstructuur al na één sprong verschijnt. Grote veranderingen duren in het eerste geval langer dan in het tweede geval.

Het is belangrijk op te merken dat de structuur sterk afhankelijk is van de waarde van de minimumgrens lim_2 . Een lagere waarde kan leiden tot een hoger niveau of een eerdere verschijning van de structuur bij een lagere k -waarde. Een hogere waarde kan een tegengesteld effect veroorzaken.

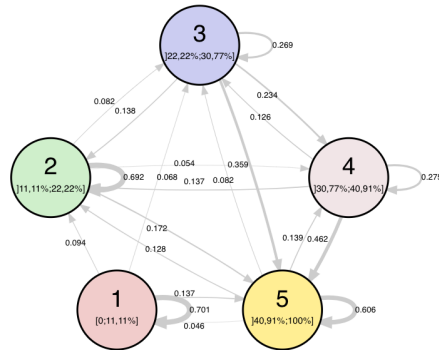
Toepassing

In Figuur 12, valt op dat het niveau van de sterstructuur op weekdays 2,5 is en op zaterdag 3. Belangrijk is op te merken dat het niveau zou kunnen veranderen als de minimumgrens lim_2 wordt aangepast. Deze niveaus suggereren dat veranderingen in de bezettingsgraad op zaterdag groter zijn dan op weekdays.

Door k te verhogen van 1 naar 3, resulteren de gegevens in de netwerken in Figuur 14. Hier zien we dat zowel de sterstructuur op weekdays als op zaterdag stijgt in niveau. Opmerkelijk is dat het niveau van weekdays met 2 stijgt naar niveau 4,5, terwijl zaterdag slechts met 1,5 stijgt, ook naar niveau 4,5. Op langere termijn vertonen beide dagen dus een even sterke sterstructuur, wat suggereert dat ze uiteindelijk even grillig veranderen.



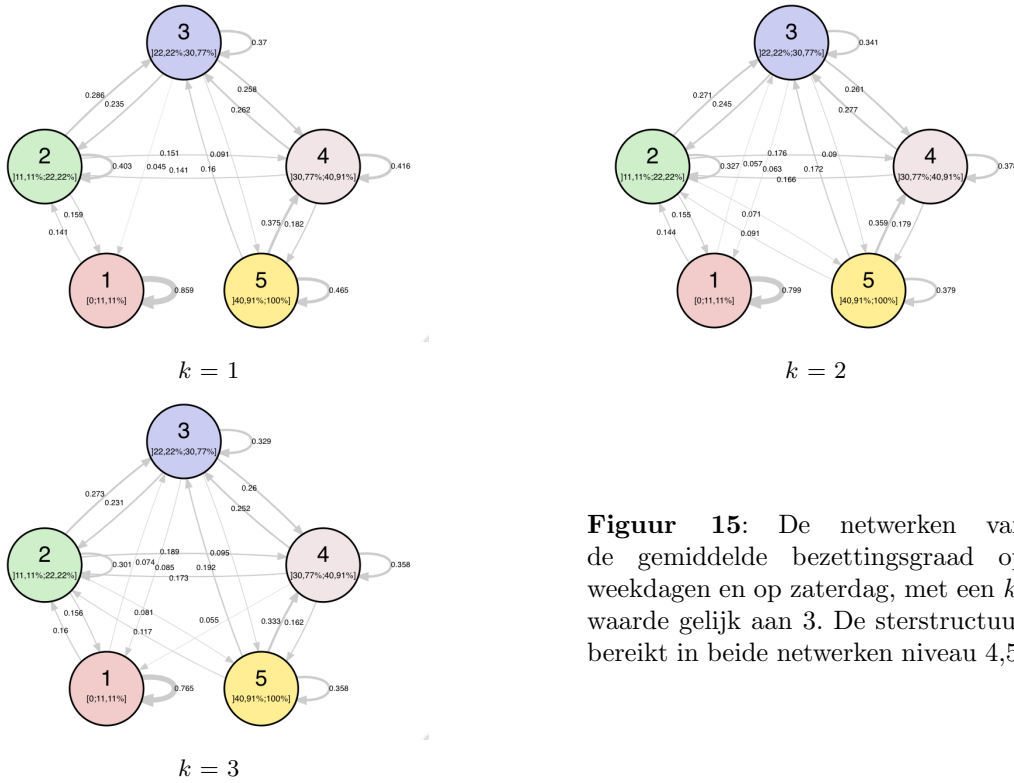
Weekdagen



Zaterdag

Figuur 14: De netwerken van de gemiddelde bezettingsgraad op weekdays en zaterdag, met een k -waarde gelijk aan 3. De sterstructuur bereikt in beide netwerken niveau 4,5.

Het niveau van de sterstructuur stijgt aanzienlijk van $k = 1$ naar $k = 3$ voor weekdays. Figuur 15 toont de evolutie van het netwerk voor $k = 1$, $k = 2$ en $k = 3$. Hieruit blijkt dat het niveau bij elke verhoging van k stijgt, van 2,5 naar 4 en uiteindelijk naar 4,5. Dit duidt op grotere veranderingen op lange termijn.



Figuur 15: De netwerken van de gemiddelde bezettingsgraad op weekdays en op zaterdag, met een k -waarde gelijk aan 3. De sterstructuur bereikt in beide netwerken niveau 4,5.

6 Conclusie

Process mining focust op het analyseren van *end-to-end* processen [1], waarbij gefocust wordt op specifieke procesinstanties. Het doel van deze paper is het ontwikkelen van een algoritme dat de mogelijkheid biedt om een volledig processysteem te analyseren, en dus niet enkel individuele procesinstanties. Dit is een methode om de dynamiek van een volledig processysteem bloot te leggen en de gebruiker inzichten te bieden in de toestand en dynamiek van het processysteem.

Het algoritme bestaat uit vier stappen: de *eventlog* wordt getransformeerd naar een tijdreeks, gevolgd door het opstellen van lijngrafieken. Vervolgens worden transitiematrices berekend om de toestanden en overgangen in het proces weer te geven. Ten slotte worden netwerken opgesteld. Zoals de lijngrafieken een visualisatie zijn van de tijdreeks, zijn de netwerken een visualisatie van de transitiematrices.

In de toepassing van het algoritme worden enkele punten toegelicht en toegepast die ervoor zorgen dat de inzichten optimaal verworven worden. In de lijngrafieken worden drie belangrijke onderdelen besproken: de startfase (plotse of geleidelijke start), de trends (stijgende, dalende, of stabiele trend), en de perioden in de activiteitenfase, zoals shifts.

Daarnaast worden drie belangrijke netwerkstructuren besproken: *selfloops*, de cirkelstructuur, en de sterstructuur. Een dominante *selfloop*structuur duidt op een stabiel proces met zeer consistente waarden. De cirkel- en sterstructuur wijzen daarentegen op veranderingen in het proces. Een cirkelstructuur duidt op geleidelijke veranderingen in het proces, terwijl een sterstructuur wijst op grotere, abrupte veranderingen. Hoe sterker of hoe sneller de sterstructuur voorkomt, hoe sneller en abrupter de veranderingen plaatsvinden.

Men kan stellen dat het ontwerp en de toepassing van het algoritme het aanzienlijk potentieel aantoont om de dynamiek in een volledig processysteem bloot te leggen. Door de gedetailleerde bespreking van de mogelijkheden van de analyse en de interpretatie van de resultaten zijn waardevolle resultaten ontdekt die de waarde van het algoritme benadrukken. Aan de hand van dit algoritme kan er inzicht verworven worden in de toestanden en dynamiek van een volledig processysteem aan de hand van een *eventlog*.

References

- [1] Aalst, W.M.P.: Process Mining: Overview and Opportunities. *ACM Transactions on Management Information Systems* **3**, 7–1717 (2012)
- [2] Li, T., He, T., Wang, Z., Zhang, Y., Chu, D.: Unraveling Process Evolution by Handling Concept Drifts in Process Mining. In: 2017 IEEE International Conference on Services Computing (SCC), pp. 442–449 (2017)
- [3] Takemoto, Y., Arizono, I.: A Study on Evaluation of Stability in Process Mean Using Bayesian Updating. In: 2020 Asia-Pacific International Symposium on Advanced Reliability and Maintenance Modeling (APARM), pp. 1–5 (2020)
- [4] Pineda, A.M., Ramos, F.M., Betting, L.E., Campanharo, A.S.L.O.: Use of Complex Networks for the Automatic Detection and the Diagnosis of Alzheimer’s Disease. In: Rojas, I., Joya, G., Catala, A. (eds.) *Advances in Computational Intelligence*, pp. 115–126. Springer, Cham (2019)
- [5] Pineda, A.M., Rodrigues, F.A.: Complex Networks to Differentiate Elderly and Young People. In: Lossio-Ventura, J.A., Valverde-Rebaza, J.C., Díaz, E., Alatrasta-Salas, H. (eds.) *Information Management and Big Data*, pp. 435–444. Springer, Cham (2021)
- [6] Campanharo, A., Sirer, M., Malmgren, R., Ramos, F., Amaral, L.: Duality between Time Series and Networks. *PloS one* **6**, 23378 (2011)
- [7] Pourbafrani, M.: Forward-Looking Process Mining : Data-Driven Simulation. PhD thesis, [object Object] (2023)
- [8] Kirchgässner, G., Wolters, J., Hassler, U.: *Introduction to Modern Time Series Analysis*, (2007)
- [9] Tulu, M.M., Hou, R., Younas, T.: Finding Important Nodes Based on Community Structure and Degree of Neighbor Nodes to Disseminate Information in Complex Networks, pp. 269–273 (2017)
- [10] Dongen, B.: BPI Challenge 2012. Eindhoven University of Technology (2012). https://data.4tu.nl/articles/dataset/BPI_Challenge_2012/12689204/1