

ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics

Bachelorscriptie Econometrie en Operationele Research

Een clustertechniek voor het Liner Shipping Network Design Problem

Naam student: Hans Veenhof

Studentnummer: 501807

Begeleider: N. Milovanovic

Tweede beoordelaar: R. Dekker

Datum definitieve versie: 20-06-2021

Samenvatting

In dit verslag wordt het Liner Shipping Network Design Problem (LSNDP) bekeken. In dit probleem wordt er een netwerk voor lijnvaart schepen gemaakt op zo'n manier dat de kosten worden geminimaliseerd. Het netwerk wordt geconstrueerd onder een aantal aannames en voorwaarden. Om dit netwerk te construeren zijn er een aantal typen constraints die in verschillende combinaties in drie verschillende modellen worden gebruikt, om zo het sterkste model te zoeken. De typen constraints zijn: routeconstraints, stroomconstraints, overslagconstraint, scheepsvlootconstraints, eliminatie van subtours constraints, symmetrie brekende constraints en innerlijke representatie constraints. Echter kunnen deze modellen tot slechts 10 havens tot optimaliteit worden opgelost. Daarom wordt er later in dit verslag een clustertechniek ontwikkeld. Met deze techniek worden voor verzamelingen met een groot aantal havens clusters gemaakt. Deze clusters kunnen vervolgens wel met het LSNDP worden opgelost.



Het geschreven in deze scriptie is de opvatting van de auteur en niet noodzakelijk die van de begeleider, tweede beoordelaar, Erasmus School of Economics of Erasmus Universiteit Rotterdam.

Inhoudsopgave

1	Introductie	3
2	Literatuur	4
3	Data	5
4	Methodologie	8
4.1	Model voor het Liner Shipping Network Design Problem	8
4.1.1	Probleem beschrijving	8
4.1.2	Berekenen van het aantal routes voor elke scheepstype	12
4.1.3	Versterking van het model	13
4.2	Clusteren voor het Liner Shipping Network Design Problem	14
5	Resultaten	15
5.1	Exacte resultaten voor het Liner Shipping Network Design Problem	15
5.2	Resultaten voor de clustertechniek	18
5.3	Prestatie van de clustertechniek	19
6	Conclusie	21
A	Kaarten en routes	23
B	Verbeteringen Model	41
C	Pseudocode voor het berekenen van het maximum aantal routes	44
D	Pseudocode van de clusteringtechniek	47
E	Bestanden uit de codebijlage	51

1 Introductie

Zeevaart is het belangrijkste transportmiddel van de internationale handel. Het aandeel van Zeevaart in de totale wereldhandel wordt geschat op ongeveer 85% door Drewry (2014). Deze Zeevaart wordt verdeeld in drie verschillende categorieën door Lawrence (1972): bedrijfsrederijen (industrial), koopvaardij (tramp) en lijnvaart (liner). De bedrijfsrederijen worden gekenmerkt door het feit dat de ladingeigenaar en scheepseigenaar hetzelfde zijn. In de koopvaardij worden schepen als een soort taxi's gebruikt om de lading te vervoeren. In tegenstelling tot de lijnvaart waar de schepen als een soort bus worden gebruikt. In dit artikel ligt de focus op de lijnvaart en hoe hiervoor een zo goedkoop mogelijk netwerk voor ontworpen kan worden waarin alle wekelijkse vracht wordt vervoerd en aan enkele andere voorwaarden wordt voldaan. De onderzoeksvraag is dan ook: *"Hoe kan een routeschema voor de lijnvaart worden gemaakt die de kosten minimaliseert?"*

Om de goedkoopste routes voor de lijnvaart te bepalen wordt het Liner Shipping Network Design Problem (LSNDP) opgelost. In dit probleem worden routes tussen de havens gemaakt en met deze routes worden vraagparen getransporteerd. Deze routes worden vervolgens toegewezen aan bepaalde schepen. Het doel is om deze routes op zo'n manier te maken dat de kosten (aanlegkosten voor een bezoek aan een haven, overslagkosten en vaarkosten) worden geminimaliseerd. Een vereiste in dit probleem is dat iedere haven met een regelmatige frequentie bezocht wordt. Deze frequentie moet ten minste één keer per week zijn, zodat het wekelijkse aanbod van de vraag kan worden opgehaald. Als een route langer dan één week duurt, worden er meerdere schepen op deze route ingezet. Alle wekelijkse vraag moet altijd verscheept worden, echter maakt het niet uit hoe lang het duurt totdat lading op de plaats van bestemming aankomt. Ook hoeft de lading niet direct van de oorsprongshaven naar de bestemmingshaven te gaan, het is toegestaan om van schip te wisselen. Dit wordt ook wel een overslag genoemd. Hier zijn wel kosten aan verbonden: de overslagkosten. Door gebruik te maken van een tweelaags netwerk, kunnen havens twee keer bezocht worden en ontstaan er zogenaamde vlinderroutes. Voor het LSNDP is van tevoren bekend wat de vraag is, waar de havens zijn en er is een afstandsmatrix beschikbaar.

Echter kan het LSNDP model slechts tot 10 havens binnen redelijke tijd worden opgelost volgens Ameln e.a. (2019). Om deze reden wordt in dit verslag ook een clustertechniek ontwikkeld. Met deze techniek worden havens die dichtbij elkaar liggen geclusterd. Als alle havens zich in een cluster bevinden, wordt voor elk cluster een hub aangewezen. Voor de aanwijzing van de hub in een cluster wordt de hoeveelheid vraag en de afstand tot andere havens voor iedere haven buiten het clusters met elkaar vergeleken. Deze hubs worden daarna ook weer met elkaar

geclusterd. Het aanwijzen van hubs en het clusteren van hubs gaat door, totdat er nog maar één hub over is. Op dat moment kunnen er namelijk geen nieuwe clusters meer gevormd worden. Na het clusteren van de havens zijn er kleinere problemen ontstaan die wel met het LSNDP kunnen worden opgelost.

Voor het LSNDP worden drie modellen getest: het standaard model, het basis model en het versterkte model. Het standaard model bestaat uit routeconstraints, stroomconstraints, overslagconstraint, scheepsvlootconstraints en eliminatie van subtours constraints. Het basis model bestaat uit dezelfde typen constraints als het standaard model en bovendien worden hier enkele symmetrie brekende constraints aan toegevoegd. Als laatste bestaat het versterkte model uit dezelfde constrainttypen als het basis model met hieraan toegevoegd nog meer symmetrie brekende constraints en innerlijke representatie constraints. Na deze modellen getest te hebben op verschillende verzamelingen van havens en vraagparen in het Baltische gebied, blijkt het versterkte model het beste model te zijn. De clustertechniek blijkt een hele snelle oplossingsmethode te zijn om een netwerk met een groter aantal havens op te lossen. Het enige nadeel van deze techniek is dat er veel overslagkosten zijn om de lading tussen verschillende clusters over te slaan.

De rest van het verslag heeft de volgende structuur: in sectie 2 is een analyse van de bestaande literatuur te vinden. De volgende sectie, sectie 3, geeft een korte samenvatting van de beschikbare data voor dit probleem. In het eerste deel van sectie 4 wordt het model behorende bij het LSNDP en de versterking hiervan besproken. In het tweede deel wordt de clustertechniek voor dit probleem besproken. Waarna in sectie 5 de resultaten voor zowel het model als de clustertechniek worden gepresenteerd. In de laatste sectie, sectie 6, wordt een conclusie gebaseerd op de resultaten weergegeven en worden suggesties voor verder onderzoek gegeven.

2 Literatuur

Het doel van de Liner Shipping Network Design Problem (LSNDP) is om de kosten van de lijnscheepvaart te minimaliseren. Ameln e.a. (2019) heeft hiervoor een model ontwikkeld. Dit model bestaat uit verschillende onderdelen. De eerste versie van het LSNDP is gemaakt door Agarwal en Ergun (2008). Hierin was sprake van een heterogene vloot, een wekelijkse frequentie voor de havenbezoeken, verschillende routes en lading overslag handelingen. Echter zijn de overslagkosten nog buiten beschouwing gelaten. Ook mag een haven meerdere keren worden bezocht, zolang dit op verschillende dagen gebeurt.

Ameln e.a. (2019) maakt gebruik van een tweelaags netwerk, om het LSNDP op te lossen. Dit netwerk is nodig om havens twee keer te kunnen bezoeken en wordt geïntroduceerd door Jacobsen en O. Madsen (1980) en O. B. Madsen (1983) als het twee-level locatie-route probleem. Als gevolg van het tweelaags netwerk kunnen vlinderroutes ontstaan. Vlinderroutes zijn routes waarin één enkele haven twee keer wordt bezocht (Reinhardt en Pisinger (2011)). Plum, Pisinger en Sigurd (2014) werkt een twee-level netwerk (met vlinderroutes) uit voor het LSNDP.

Voor de LSNDP notatie die in dit artikel zal worden gebruikt, is het van belang om vooraf het maximum aantal routes te bepalen. De grootte van het model is afhankelijk van het aantal maximum routes dat vooraf bepaald is. Sinds problemen met minder constraints makkelijker op te lossen zijn dan problemen met meer constraints, is het belangrijk om het maximum aantal routes niet te hoog in te stellen. Het maximum aantal routes kan worden bepaald door het veeg algoritme van Laporte e.a. (2000). In het veeg algoritme worden toegestane clusters van havens voor een route bepaald door een straal te laten roteren om een hub haven.

Uit Ameln e.a. (2019) blijkt dat de LSNDP formulatie tot maar 10 havens optimaal opgelost kan worden. Om een goede benadering te maken voor problemen met meer dan 10 havens kan dit probleem opgesplitst worden in meerder kleinere clusters van havens. Welke bestaan uit slechts enkele havens en vraagparen. Hierdoor kan een hub-and-spoke netwerk ontstaan, zoals beschreven in Ameln e.a. (2019). In Chen (2007) wordt besproken hoe het best bepaald kan worden welke haven als hub dient te worden gebruikt.

3 Data

Brouer e.a. (2014) heeft data beschikbaar voor verschillende gebieden in de wereld. Hiervan wordt de data voor havens in twee gebieden gebruikt: het Baltische gebied en het Middellandse Zeegebied. In het Baltische gebied zijn 12 havens en in het middellandse Zeegebied zijn 39 havens. Om meerdere en verschillende verzamelingen van havens en vraagparen te hebben, wordt het Baltische gebied gesplitst in 5 verschillende groepen, met verschillende aantal havens en aantal vraagparen. In deze sectie wordt alleen de relevante data voor dit artikel besproken. In Brouer e.a. (2014) is de volledige beschrijving van de data te vinden. Naast de havens zijn ook de vaarroutes tussen twee havens, de vraag en schepen beschikbaar in de data.

Voor de havens zijn verschillende parameters beschikbaar. De eerste is de United Nations Code for Trade and Transport Locations (UNLocode), dit is een internationale vijfletterige code voor de plaats waar de haven zich bevindt. Ook de naam van de haven is beschikbaar. Daarnaast

is ook het land en de regio waarin de haven zich bevindt beschikbaar. Bovendien bestaat de data uit de lengtegraad en breedtegraad van de haven. Ook de cabotage regio, cabotage is het vervoer van goederen en/of passagiers binnen één land door een vervoersmaatschappij van een ander land, is beschikbaar. De volgende parameter is de maximale scheepsdiepte (in meters). De laatste parameters van de haven bestaan uit verschillende kosten voor de schepen voor de desbetreffende haven (allemaal in USD): de laad- en loskosten per FFE (Forty Foot Equivalent), overslagkosten per FFE, vaste aanlegkosten per FFE en variabele aanlegkosten per FFE. In tabel 1 staat een voorbeeld van de data voor een haven. In het eerste deel van tabel 3 en in het eerste deel van tabel 4 staan de karakteristieken van de havens van respectievelijk het Baltische gebied en het Middellandse Zeegebied. In bijlage A1 staan figuur A1 en figuur A2 waarin de kaarten met de havens van respectievelijk het Baltische gebied en het Middellandse Zee worden gepresenteerd.

Tabel 1: Voorbeeld data voor een haven.

UNLocode	Naam	Land	Cabotage	Regio	Lengtegraad	Breedtegraad
DEBRV	Bremerhaven	Duitsland	Duitsland	Noord Europa	8.58	53.55
UNLocode	Diepgang ¹	Laad- en loskst ²	overslag kst ²	vaste aanlegkst ²	variabele aanlegkst ³	
DEBRV	13.5	199.00	121.00	11795.00	14.00	

¹ In meters

² In US Dollars

³ In US Dollars per FFE

Voor de vaarroutes tussen twee havens is een afstandsmatrix (in Zeemijlen) beschikbaar. Voor iedere afstand is de starthaven en de eindhaven bekend. Voor elke vaarroute tussen twee havens is ook bekend of deze door het Panamakanaal of door het Suezkanaal gaan. Hiervoor is een dummy variabele beschikbaar, welke gelijk is aan 0 als de route tussen de starthaven en de eindhaven niet door respectievelijk het Panama- of Suezkanaal gaat en is gelijk aan 1 in alle andere gevallen. Deze gegevens zijn belangrijk voor de tol die hier betaald moet worden. Als laatst is de diepgang (in meters) van de route tussen de start- en eindhaven bekend, als deze relevant is. In het tweede deel van tabel 3 en in het tweede deel van tabel 4 staan de belangrijkste statistieken van de afstanden voor respectievelijk het Baltische en Middellandse Zeegebied gepresenteerd.

Voor de vraag is bekend wat de oorsprong van het aanbod en de bestemming van de vraag is, respectievelijk de oorsprongshaven en de bestemmingshaven. Ook is bekend hoeveel FFE deze per week bedraagt. In het derde deel van tabel 3 en het derde deel van tabel 4 staan hiervan de relevante statistieken voor respectievelijk het Baltische en Middellandse Zeegebied.

In het Baltische gebied varen 2 verschillende typen schepen de Feeder 450 en de Feeder 800. Deze scheepstypen varen ook in het Middellandse Zeegebied, hier vaart ook de Panamax 1200. De belangrijkste kenmerken van deze scheepstypen staan weergegeven in tabel 2.

Tabel 2: Belangrijkste statistieken van de scheepstypen.

Scheepstype	Capaciteit ³	Operatie kosten ²	Diepgang ¹	Minimale snelheid ⁴	Maximale snelheid ⁴
Feeder 450	450	5000	8.0	10	14
Feeder 800	800	8000	9.5	10	17
Panamax 1200	1200	11000	12.0	12	19
Scheepstype	Ontwerp snelheid ⁴	Verbruik ⁵	Stationair verbruik ⁶	Panama tol ²	Suez tol ²
Feeder 450	12	18.8	2.4	64800	175769
Feeder 800	14	23.7	2.5	115200	218445
Panamax 1200	18	52.5	4.0	172800	267217

¹ In meters² In US Dollars³ In FFE⁴ In knopen⁵ In ton per dag bij ontwerpssnelheid⁶ In ton per dag bij het binnengaan van een haven

Tabel 3: Belangrijkste statistieken van de data voor het Baltische gebied.

	Gemiddelde	Mediaan	Maximale waarde	Minimale waarde	Standaard afwijking
Lengte graad	14.50	11.08	30.27	5.32	8.70
Breedte graad	58.18	58.56	62.49	53.55	2.90
Diepgang ¹	10.58	10.25	13.50	8.00	1.86
Laad- en loskosten ²	275.00	240.00	684.00	84.00	162.02
Overslagkosten ²	95.58	107.00	203.00	2.00	58.50
Vaste aanlegkosten ²	13555.42	14648.00	26838.00	722.00	10337.12
Variabele aanlegkosten ²	71.42	44.50	180.00	7.00	62.39
Afstand ³	628.58	592.00	1291.00	70.00	318.26
Panamakanaal	0.00	0.00	0.00	0.00	0.00
Suezkanaal	0.00	0.00	0.00	0.00	0.00
Diepgang ^{1, 5}	n.v.t.	n.v.t.	n.v.t.	n.v.t.	n.v.t.
Vraag ⁴ (3,4) ⁶	692.50	628.50	1215.00	298.00	382.45
Vraag ⁴ (5,8) ⁶	330.38	324.50	365.00	310.00	19.41
Vraag ⁴ (7,10) ⁶	166.90	130.00	456.00	6.00	155.93
Vraag ⁴ (7,12) ⁶	381.33	283.00	1215.00	7.00	326.64
Vraag ⁴ (9,16) ⁶	298.00	209.00	1215.00	7.00	317.20

¹ In meters² In US Dollars³ In Zeemijlen⁴ In FFE per week⁵ 0 observaties⁶ (Aantal havens, aantal vraagparen)

Tabel 4: Belangrijkste statistieken van de data voor het Middellandse Zeegebied.

	Gemiddelde	Mediaan	Maximale waarde	Minimale waarde	Standaard afwijking
Lengte graad	13.42	10.74	41.68	-16.25	17.24
Breedte graad	37.11	36.80	46.45	28.10	4.56
Diepgang ¹	10.93	11.00	13.50	8.00	2.21
Laad- en loskosten ²	208.79	202.00	597.00	4.00	145.87
Overslagkosten ²	126.18	136.00	342.00	0.00	54.57
Vaste aanlegkosten ²	13752.87	3124.00	118592.00	32.00	29520.08
Variabele aanlegkosten ²	21.05	12.00	119.00	0.00	28.72
Afstand ³	1242.64	1200.00	3072.00	20.00	684.27
Panamakanaal	0.00	0.00	0.00	0.00	0.00
Suezkanaal	0.00	0.00	0.00	0.00	0.00
Diepgang ^{1, 5}	n.v.t.	n.v.t.	n.v.t.	n.v.t.	n.v.t.
Vraag ⁴ (39, 365) ⁶	20.67	12.00	266.00	1.00	28.76

¹ In meters² In US Dollars³ In Zeemijlen⁴ In FFE per week⁵ 0 observaties⁶ Aantal havens, aantal vraagparen

4 Methodologie

In deze sectie wordt eerst een beschrijving van het Liner Shipping Network Design Problem gepresenteerd met bij behorende aannames. Dit probleem, deze aannames en het model zijn gebaseerd op Ameln e.a. (2019). Op hun model zijn 3 kleinere en grotere verbeteringen gedaan. Daarna wordt de clustertechniek voor dit probleem gepresenteerd. Als laatst wordt de clustertechniek geëvalueerd.

4.1 Model voor het Liner Shipping Network Design Problem

4.1.1 Probleem beschrijving

Het Liner Shipping Design Network Problem, waarin P het aantal havens is, kan met een graaf $G = (\mathcal{N}, \mathcal{A})$ worden weergegeven. Hierin is $\mathcal{N} = \{1, \dots, 2P\}$ de verzameling van de (knoop)punten. In deze verzameling zijn alle havens twee keer ingevoegd, één keer in de bovenste laag ($\mathcal{N}^U = \{1, \dots, P\}$) en één keer in de onderste laag ($\mathcal{N}^L = \{P + 1, \dots, 2P\}$). Hierdoor ontstaat het tweelaags netwerk. In dit netwerk wordt iedere haven p door de punten p (in de bovenste laag) en $P + p$ (in de onderste laag) vertegenwoordigd. De set van lijnen bevat alle lijnen tussen alle havens binnen één laag (de bovenste of de onderste laag) voor beide lagen, met daarop aanvullend de lijnen die van haven p naar haven $P + p$ gaan en vice versa voor alle havens. Hierdoor kan je alleen tussen de lagen wisselen door naar dezelfde haven in de andere laag te gaan. De set van lijnen wordt als volgt gedefinieerd: $\mathcal{A} = \{(i, j) | i, j \in \mathcal{N}^U\} \cup \{(i, j) | i, j \in \mathcal{N}^L\} \cup \{(i, P + i) | i \in \mathcal{N}^U\} \cup \{(P + i, i) | i \in \mathcal{N}^U\}$. Door deze tweelaags structuur kunnen alle havens twee keer worden bezocht, behalve de havens waar van laag wordt gewisseld. Hierdoor kunnen ook vlinderroutes ontstaan.

Elk vraagpaar zit in de verzameling $\mathcal{D}$ en wordt weergegeven als (o, d) , waar o de oorsprongshaven is en d de bestemmingshaven. De wekelijkse vraag wordt weergegeven met D_{od} in Forty Foot Equivalent (FFE). Voor de stroom van de lading van de vraag wordt aangenomen dat deze niet mag terugkeren naar de oorsprongshaven en ook niet mag vertrekken uit de bestemmingshaven. De vraag kan worden verscheept met verschillende typen schepen. De verzameling scheepstypen wordt weergegeven met $\mathcal{K}$. Elk type schip heeft een capaciteit van Q_k en er varen N_k schepen van type k rond. De vaartijd van haven i naar haven j wordt weergegeven door T_{ijk} in uren. Om T_{ijk} te berekenen wordt de volgende formule gebruikt: $T_{ijk} = \frac{\text{afstand}}{\text{snelheid}}$, de afstand is gemeten in Zeemijlen en de snelheid is in knopen en is gelijk aan de ontwerpsnelheid van het schip. Een verzameling van alle routes (leeg of gevuld met havens) voor scheepstype k wordt weergegeven met $\mathcal{R}_k$ en moet van tevoren worden bepaald. De maximale duur van een route

is gelijk aan $\bar{T}$. In dit probleem wordt $\bar{T}$ gelijk gesteld aan oneindig, omdat er geen maximale reistijd voor de routes is.

In het probleem zijn er verschillende soorten kosten, alle kosten zijn afhankelijk van het scheepstype. Alle kosten zijn in US Dollars (USD). De eerste zijn de aanlegkosten bij een haven, weergegeven met C_{pk}^P . Deze wordt berekend met de volgende formule: $C_{pk}^P = \text{vaste aanlegkosten} + \text{variabele aanlegkosten} * \text{capaciteit schip} (FFE)$. De volgende kosten zijn de vaste en variabele vaarkosten, weergegeven met respectievelijk C_k^S en C_k^V . De vaste vaarkosten zijn gelijk aan de operatie kosten van een schip vermenigvuldigd met 7, om zo de kosten van één week te krijgen. De variabele vaarkosten worden met de volgende formule berekend: $C_k^V = \left(\frac{\text{afstand}}{\text{ontwerpsnelheid}} \right)^3 * \text{vaarsnelheid} * \text{brandstofprijs}$. Zoals eerder geïntroduceerd, wordt in dit verslag aangenomen dat de vaarsnelheid gelijk is aan de ontwerpsnelheid. Er is ook sprake van een vaste brandstofprijs, deze is gelijk aan 600 USD per ton.

Hieronder is het volledige model voor dit probleem inclusief beschrijving weergegeven, met dit model worden routes voor de lijnschepen zodanig gevormd dat de kosten worden geminimaliseerd. Hierin zijn de volgende beslissingsvariabelen gebruikt: y_{kr} is een binaire variabele die gelijk is aan 1 als route r door ten minste één schip van scheepstype k wordt gevaren en anders gelijk aan 0. De binaire variabele x_{ijkr} is gelijk aan 1 als een schip van scheepstype k route r vaart en hierbij gebruik maakt van lijn (i, j) en is anders gelijk aan 0. De volgende variabele is f_{ijodkr} en is een continue variabele en geeft de stroom van de vraag (o, d) weer die is vervoerd door een schip van scheepstype k over route r over lijn (i, j) . De volgende is t_{iodkr} , deze is continu en geeft de hoeveelheid vraag (o, d) die overgeslagen wordt in haven i op route r door een schip van scheepstype k weer. De beslissingsvariabele w_{kr} geeft het aantal schepen aan die een route r varen. Deze beslissingsvariabele is discreet en neemt een waarde tussen 0 en $\bar{T}$ aan. De laatste twee beslissingsvariabelen zijn toegevoegd voor de subtour eliminatie van dit probleem. De eerste hiervan is s_{ikr} en geeft aan de hoeveelste haven in de reeks havens van route r van scheepstype k haven i is. De laatste beslissingsvariabele is d_{ikr} , deze is binair. Het is gelijk aan 1 als haven i het depot is van route r van scheepstype k en gelijk aan 0 in alle andere gevallen.

$$\begin{aligned}
\text{minimize} \quad & \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}_k} C_k^V w_{kr} + \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N} | (i,j) \in A} \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}_k} C_k^S T_{ijk} x_{ijkr} + \sum_{i \in \mathcal{N}^U} \sum_{j \in \mathcal{N}^U | (i,j) \in A} \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}_k} C_{ik}^P (x_{ijkr} + x_{P+i,P+j,kr}) \\
& + \sum_{i \in \mathcal{N}^U} \sum_{(o,d) \in \mathcal{D}} \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}_k} C_i^T (t_{iodkr} + t_{P+i,odkr})
\end{aligned} \tag{1a}$$

$$\text{subject to} \quad \sum_{i \in \mathcal{N}^U} \sum_{j \in \mathcal{N}^U | (i,j) \in A} 2(x_{ijkr} + x_{P+i,P+j,kr}) + \sum_{i \in \mathcal{N}^U} (x_{i,P+i,kr} + x_{P+i,ikr}) - 6y_{kr} \geq 0 \quad k \in \mathcal{K}, r \in \mathcal{R}_k, \tag{1b}$$

$$\sum_{i \in \mathcal{N} | (i,j) \in A} x_{ijkr} - y_{kr} \leq 0 \quad j \in \mathcal{N}, k \in \mathcal{K}, r \in \mathcal{R}_k, \tag{1c}$$

$$\sum_{i \in \mathcal{N} | (i,j) \in A} x_{ijkr} - \sum_{i \in \mathcal{N} | (i,j) \in A} x_{jikr} = 0 \quad j \in \mathcal{N}, k \in \mathcal{K}, r \in \mathcal{R}_k, \tag{1d}$$

$$\sum_{i \in \mathcal{N}^U | (i,j) \in A} \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}_k} (f_{ijodkr} + f_{P+i,P+j,odkr} - f_{jiodkr} - f_{P+j,P+i,odkr}) = 0 \quad j \in \mathcal{N}^U, (o,d) \in \mathcal{D}, j \neq o, j \neq d, \tag{1e}$$

$$\sum_{j \in \mathcal{N}^U} \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}_k} (f_{ojodkr} + f_{P+o,P+j,odkr}) = D_{od} \quad (o,d) \in \mathcal{D}, \tag{1f}$$

$$\sum_{i \in \mathcal{N}^U} \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}_k} (f_{idodkr} + f_{P+i,P+d,odkr}) = D_{od} \quad (o,d) \in \mathcal{D}, \tag{1g}$$

$$\sum_{(o,d) \in \mathcal{D}} f_{ijodkr} \leq Q_k x_{ijkr} \quad (i,j) \in \mathcal{A}, k \in \mathcal{K}, r \in \mathcal{R}_k, \tag{1h}$$

$$\sum_{j \in \mathcal{N} | (i,j) \in A} (f_{ijodkr} - f_{jiodkr}) \leq t_{iodkr} \quad i \in \mathcal{N}, (o,d) \in \mathcal{D}, i \neq o, i \neq d, k \in \mathcal{K}, r \in \mathcal{R}_k, \tag{1i}$$

$$\sum_{i \in \mathcal{N}^U} \sum_{j \in \mathcal{N}^U | (i,j) \in \mathcal{A}} \frac{T_{ijk}}{168} (x_{ijk} + x_{P+i, P+j, kr}) \leq w_{kr} \quad k \in \mathcal{K}, r \in \mathcal{R}_k, \quad (1j)$$

$$\sum_{r \in \mathcal{R}_k} w_{kr} \leq N_k \quad k \in \mathcal{K}, \quad (1k)$$

$$\sum_{i \in \mathcal{N}} d_{ikr} = y_{kr} \quad k \in \mathcal{K}, r \in \mathcal{R}_k, \quad (1l)$$

$$s_{ikr} - s_{jkr} + (|\mathcal{N}| - 1)x_{ijk} + (|\mathcal{N}| - 3)x_{jkr} \leq |\mathcal{N}| - 2 + |\mathcal{N}|d_{jkr} \quad (i, j) \in \mathcal{A}, k \in \mathcal{K}, r \in \mathcal{R}_k, \quad (1m)$$

$$x_{ijk} \in \{0, 1\} \quad (i, j) \in \mathcal{A}, k \in \mathcal{K}, r \in \mathcal{R}_k, \quad (1n)$$

$$y_{kr} \in \{0, 1\} \quad k \in \mathcal{K}, r \in \mathcal{R}_k, \quad (1o)$$

$$d_{ikr} \in \{0, 1\} \quad i \in \mathcal{N}, k \in \mathcal{K}, r \in \mathcal{R}_k, \quad (1p)$$

$$w_{kr} \in \{0, 1, \dots, \bar{T}\} \quad k \in \mathcal{K}, r \in \mathcal{R}_k, \quad (1q)$$

$$f_{iodkr} \geq 0 \quad (i, j) \in \mathcal{A}, (o, d) \in \mathcal{D}, k \in \mathcal{K}, r \in \mathcal{R}_k, \quad (1r)$$

$$t_{iodkr} \geq 0 \quad i \in \mathcal{N}, (o, d) \in \mathcal{D}, k \in \mathcal{K}, r \in \mathcal{R}_k, \quad (1s)$$

$$s_{ikr} \geq 0 \quad i \in \mathcal{N}, k \in \mathcal{K}, r \in \mathcal{R}_k \quad (1t)$$

Merk op dat er een aantal verschillen zijn met dit model en het originele model van Ameln e.a. (2019). Als eerst is constraint (1j) omgedraaid ten opzichte van de originele constraint. In deze constraint is ook een tijd schalingsfactor toegevoegd. Ook is er een extra aanname gemaakt: de stroom van de vraag mag niet terug naar de oorsprongshaven gaan en de stroom van de vraag mag ook niet vertrekken uit de bestemmingshaven. Als laatst is de notatie van het model aangepast. Zie voor verdere details bijlage B

De doelfunctie is weergegeven in (1a). Deze minimaliseert de kosten van het LSNDP, bestaande uit de vaste kosten voor het gebruiken van een schip, de vaarkosten van schepen op bepaalde routes, havenvergoedingen voor het aanleggen in een haven en overslagkosten.

De eerste drie constraints zijn route constraints. De eerste constraint, constraint (1b), heeft verschillende functies, de eerste is dat hij y_{kr} verplicht 0 te zijn op het moment dat er geen lijnen voor de route r worden gebruikt. Ook zorgt hij ervoor dat niet op één route de lijnen (i, j) en (j, i) gebruikt kunnen worden. Constraint (1c) zorgt ervoor dat elke haven (in elke laag) maximaal één keer bezocht mag worden. De laatste route constraint, constraint (1d) zorgt ervoor dat als een haven bezocht wordt dat er een ingaande en uitgaande lijn is.

De volgende vier constraints zijn stroom constraints. Constraint (1e) zorgt ervoor dat de lading wordt doorgevoerd als het niet de oorsprongs- of bestemmingshaven is. De volgende constraint, constraint (1f), zorgt dat alle lading alleen in de oorsprongshaven wordt ingeladen. Waar constraint (1g) er juist voor zorgt dat de lading alleen in de bestemmingshaven wordt gelost. De laatste constraint uit deze categorie is constraint (1h) en zorgt ervoor dat de capaciteit van het schip niet wordt overschreden.

Constraint (1i) is de overslagconstraint. Deze constraint zet de waarde van de overslag variabelen in de overslaghaven gelijk aan de vraag op de route.

De volgende 2 constraints zijn constraints voor de scheepsvloot. De eerste constraint, constraint (1j), berekent het aantal schepen van scheepstype k die nodig zijn voor route r om de wekelijkse frequentie tot stand te laten komen. De volgende, constraint (1k), zorgt ervoor dat er niet meer schepen dan het aantal beschikbare schepen wordt gebruikt.

De laatste categorie van constraints bestaat uit 2 constraints die zorgen voor een eliminatie van subtours. Constraint (1l) forceert dat elke route exact één depot heeft. Constraint (1m) geeft elke haven in een route een nummer, hierdoor kunnen er geen subtours ontstaan. De nummering begint in het depot.

Als laatst komen de variabele restricties. Deze zijn weergegeven in (1n) tot en met (1t).

4.1.2 Berekenen van het aantal routes voor elke scheepstype

Het maximum aantal routes dat een scheepstype kan varen, moet van tevoren worden bepaald. Dit kan gedaan worden door $\mathcal{R}_k$ een grote waarde mee te geven voor elk scheepstype. Echter is de grootte van het probleem gecorreleerd met $|\mathcal{R}_k|$. Dit komt doordat er meer variabelen en constraints zijn op het moment dat $|\mathcal{R}_k|$ groter wordt. Daarom wordt er gebruik gemaakt van een heuristiek om het maximale benodigde aantal routes te berekenen. Deze heuristiek is gebaseerd op het Feeder heuristiek van Ameln e.a. (2019).

In deze heuristiek wordt één haven aangewezen als hub haven. Dit is de haven waarvan de som van alle vraag met deze haven als oorsprongshaven en alle vraag met deze haven als bestemmingshaven het grootst is. Alle overige havens worden gesorteerd op de hoek tussen de lijn uit de hub verticaal naar boven en de lijn van de hub naar de haven, beginnend met de kleinste hoek. Vervolgens worden deze havens één voor één afgegaan en gekeken of alle vraag met deze haven als oorsprongshaven of bestemmingshaven nog mee kan op de huidige route. Als dat niet het geval is wordt het aantal benodigde routes met één opgehoogd. Als alle havens bezocht zijn, wordt de bovengrens voor het aantal routes ($|\mathcal{R}_k|$) gelijk gesteld aan het aantal benodigde routes. Een pseudocode voor deze heuristiek wordt gepresenteerd in bijlage C.

4.1.3 Versterking van het model

In Ameln e.a. (2019) wordt het LSNDP versterkt door een aantal constraints toe te voegen of te vervangen. Als eerst wordt uit het standaard model ((1a) - (1t)) het basis model gemaakt. In dit basis model wordt constraint (1l) vervangen door de volgende twee constraints:

$$\sum_{i \in N^U} d_{ikr} = y_{kr} \quad k \in \mathcal{K}, r \in \mathcal{R}_k \quad (2)$$

$$d_{ikr} = 0 \quad i \in N^L, k \in \mathcal{K}, r \in \mathcal{R}_k \quad (3)$$

Deze twee constraints zorgen ervoor dat er slechts één depot in een route is en dat deze in de bovenste laag zit. Aan het basis model worden nog drie extra constraints toegevoegd, constraint (4) tot en met constraint (6). Dit zijn alle drie symmetrie brekingsconstraints.

$$\sum_{i \in N^U} \sum_{j \in N^U} x_{ijkr} - \sum_{i \in N^L} \sum_{j \in N^L} x_{ijkr} \geq 0 \quad k \in \mathcal{K}, r \in \mathcal{R}_k \quad (4)$$

$$d_{ikr} + x_{i, P+i, kr} \leq 1 \quad i \in N^U, k \in \mathcal{K}, r \in \mathcal{R}_k \quad (5)$$

$$(|\mathcal{N}| - 1)(1 - d_{ikr}) \geq s_{ikr} \quad i \in \mathcal{N}, k \in \mathcal{K}, r \in \mathcal{R}_k \quad (6)$$

Constraint (4) forceert dat het aantal gebruikte lijnen in de bovenste laag ten minste zo groot is als het aantal gebruikte lijnen in de onderste laag. In constraint (5) wordt gezorgd dat lijnen uit een depot altijd in de bovenste laag liggen. Als laatst zorgt constraint (6) ervoor dat er geen symmetrie meer plaats kan vinden in de subtour eliminatie.

In Ameln e.a. (2019) worden aan dit basis model verschillende combinaties van constraints toegevoegd en vergeleken welke combinatie van constraints voor het sterkste model zorgde. Hieruit bleek dan onder andere dat constraints (7) tot en met constraint (9) kunnen worden

toegevoegd voor een betere werking. Hieruit ontstaat het versterkte model. Voor constraint (9) is extra notatie nodig. Als eerst is $\mathcal{N}^S$ een subset van $\mathcal{N}$, op zo een manier dat of beide punten van één haven (het punt in de bovenste laag en het punt in de onderste laag van het netwerk) in de subset zitten of beide niet in de subset zitten. Vervolgens is $\bar{\mathcal{N}}^S$ het complement van $\mathcal{N}^S$. Hierbij hoort ook een verzameling van lijnen, die is als volgt gedefinieerd: $\mathcal{A}^S = \{(i, j) | i \in \mathcal{N}^S, j \in \bar{\mathcal{N}}^S\} \cup \{(P + i, P + j) | i \in \mathcal{N}^S, j \in \bar{\mathcal{N}}^S\}$.

$$\sum_{(i,j) \in \mathcal{A}} t_{ijk}(x_{ijk,r} - x_{ijk,r-1}) \leq 0 \quad k \in \mathcal{K}, r \in \mathcal{R}_k, r > 1 \quad (7)$$

$$y_{kr} - y_{k,r-1} \leq 0 \quad k \in \mathcal{K}, r \in \mathcal{R}_k, r > 1 \quad (8)$$

$$\sum_{k \in \mathcal{K}} Q_k \left(\sum_{(i,j) \in \mathcal{A}^S} \sum_{r \in \mathcal{R}_k} x_{ijk,r} \right) \geq \sum_{o \in \mathcal{N}^S} \sum_{d \in \bar{\mathcal{N}}^S} D_{od} \quad \mathcal{N}^S \subset \mathcal{N} \quad (9)$$

Constraint (7) en constraint (8) zorgen er samen voor dat er ook geen symmetrie in de verzameling van routes zit, deze constraints zorgen er namelijk voor dat de tijdsduur van de eerste route tenminste net zo groot is als de tijdsduur van de tweede route, enzovoort. Met constraint (9) wordt de capaciteit constraint versterkt.

4.2 Clusteren voor het Liner Shipping Network Design Problem

Uit Ameln e.a. (2019) volgt dat het model beschreven in sectie 4.1 voor slechts maximaal tien havens kan worden opgelost. Om toch een netwerk voor een groter aantal havens te kunnen ontwikkelen, wordt hieronder een clustertechniek gepresenteerd om een benadering te maken. Met behulp van de clustertechniek worden de havens die bij elkaar in de buurt liggen in het netwerk met elkaar geclusterd, waarna voor elk cluster een hub wordt aangewezen. Daarna worden de hubs die bij elkaar liggen geclusterd, et cetera totdat alle clusters met elkaar verbonden zijn. Ieder cluster bestaat uit minimaal n havens en maximaal m havens. Na het clusteren kan voor ieder cluster het LSNDP worden opgelost.

De clustertechniek bestaat uit twee delen die na elkaar worden herhaald. In het eerste deel worden havens die dichtbij elkaar liggen in één cluster gedaan. In het tweede deel wordt de hub van het cluster aangewezen. Voor de eerste stap worden alle havens in een verzameling gedaan en gesorteerd op de gemiddelde afstand naar alle andere havens, beginnend met de haven met de grootste afstand tot alle andere havens. De haven met de grootste gemiddelde afstand wordt in het eerste cluster gedaan. Hierna wordt de dichtstbijzijnde haven bij deze haven ook in dit cluster toegevoegd. Waarna de haven die het dichtstbij één van de havens in het cluster ligt ook in dit cluster wordt gedaan, totdat het minimum aantal havens voor het cluster (n) is

bereikt. Daarna worden nog meer havens toegevoegd aan het cluster als de afstand kleiner is dan $\rho * \text{de langste afstand tussen havens in het cluster}$ en maximaal totdat het maximum aantal havens voor het cluster (m) bereikt is.

In het tweede deel van de clustertechniek wordt de hub voor het cluster aangewezen. Hiervoor wordt voor iedere haven in het cluster de volgende vergelijking berekend: $\frac{\text{vraag naar haven} - \text{vraag vanaf haven}}{2 * \text{totale vraag}} * \frac{\text{totale afstand van en naar alle havens}}{\text{totale afstand vanaf haven}}$. Waarna de haven met de grootste waarde als hub wordt aangewezen. Deze vergelijking is gebaseerd op Chen (2007). De volledige pseudocode van beide stappen voor de clustertechniek is te vinden in algoritme D1 in bijlage D.

Na deze twee stappen wordt stap één herhaald. Echter worden nu niet alle havens in een verzameling gedaan, maar alleen de hubs. Daarna wordt wederom in stap twee, voor de geconstrueerde clusters, een hub aangewezen. Stap één en stap twee worden herhaald, totdat er nog slechts één hub in de verzameling zit.

5 Resultaten

In deze sectie worden de resultaten van de modellen en clustertechniek uit de vorige sectie gepresenteerd. Alle resultaten zijn gegenereerd door een computer met een Intel Core i3-4170 CPU @3.70GHz met 2 cores en 4 threads. Ook is voor het berekenen van ieder resultaat een bovengrens van 10 uur ingesteld. In sectie 5.1 worden de exacte resultaten voor netwerken met minder dan tien havens gepresenteerd. Waarna in sectie 5.2 de resultaten voor een netwerk met meer havens wordt gepresenteerd, welke geconstrueerd is met de clustertechniek.

5.1 Exacte resultaten voor het Liner Shipping Network Design Problem

In tabel 5 staan de resultaten van de oplossingen die genegeerd zijn met de modellen gepresenteerd in sectie 4.1. Er zijn vier Feeder 450 schepen en twee Feeder 800 schepen beschikbaar. In de eerste en tweede kolom staan respectievelijk het aantal havens en het aantal vraagparen waarvoor er een netwerk gevormd moet worden. Vervolgens staat in de derde kolom het model nummer waarmee de oplossing berekend is. Model 1 is het standaard model, zoals beschreven in sectie 4.1.1, en bestaat uit constraint (1a) - (1t). Model 2 is het basis model en model 3 is het verbeterde model, deze staan beide beschreven in 4.1.3. Model 2 bestaat uit constraints (1a) - (1k), (1m) - (1t) en (2) - (6). Model 3 bestaat uit constraint (1a) - (1k), (1m) - (1t) en (2) - (9). In kolom vier staat de gevonden doelfunctiewaarde en in de volgende kolom staat de tijd die het model erover gedaan heeft om deze te berekenen. Vervolgens is de waarde die de Feeder heuristiek heeft berekend voor de bovengrens van het aantal routes weergegeven. In de kolom daarna, in de zevende kolom, staat het werkelijke aantal routes in de oplossing gegeven en in

de kolom daarna het aantal uitgevoerde overslagen. Als laatst staat het nummer van de bijlage gepresenteerd, waarin de volledige oplossing van het probleem staat gepresenteerd.

Tabel 5: Resultaten met 4 Feeder 450 schepen en 2 Feeder 800 schepen.

Havens ¹	Vraagparen ¹	Model	Doelfunctiewaarde	Tijd ²	Heuristiek	$ \mathcal{R}_k ^3$	Routes ³	Overslag ¹	Bijlage
3	4	1	6493310.57	0.39	4/2	3/1	1		A.2.1
		2	6493310.57	0.42	4/2	3/1	1		A.2.1
		3	6493310.57	0.69	4/2	3/1	1		A.2.1
5	8	1	2969496.29	5.75	2/1	1/1	1		A.2.2
		2	2969496.29	5.36	2/1	1/1	1		A.2.2
		3	2969496.29	2.36	2/1	1/1	1		A.2.2
7	10	1	4967074.00	112.36	2/1	1/1	0		A.2.3
		2	4967074.00	116.78	2/1	1/1	0		A.2.3
		3	4967074.00	4.95	2/1	1/1	0		A.2.3

¹ Aantal

² In seconde

³ Feeder 450/ Feeder 800

Voor een netwerk bestaande uit 7 havens en 12 vraagparen en een bestaande uit 9 havens en 16 vraagparen werd na 10 uur geen oplossing gevonden als er 4 Feeder 450 schepen en 2 Feeder 800 schepen zijn. Daarom zijn voor deze netwerken 8 Feeder 450 schepen en 4 Feeder 800 schepen beschikbaar. De resultaten hiervan staan weergegeven in tabel 6. Deze tabel is hetzelfde opgebouwd als tabel 5, echter is in deze tabel de doelfunctiewaarde gesplitst in drie subkolommen, wanneer de tijdslimiet bereikt is. Dit zijn de volgende kolommen: beste ondergrens en beste bovengrens, welke respectievelijk de minimale doelfunctiewaarde en maximale doelfunctiewaarde weergeven en het gat welke wordt berekend met de volgende formule: $\frac{\text{maximum}-\text{minimum}}{\text{minimum}} * 100\%$.

Tabel 6: Resultaten met 8 Feeder 450 schepen en 4 Feeder 800 schepen.

Hav. ¹	Vra. ¹	Mod.	Onder.	Obj. Boven.	Gat	Tijd ²	Heur.	$ \mathcal{R}_k ^3$	Rou. ³	Over. ¹	Bij.
7	12	1	8727239.43	8418598.90	3.54%	36000.00	6/2	n.v.t.	n.v.t.		A.2.4
		2	8758501.57	5427527.24	3.78%	36000.00	6/2	n.v.t.	n.v.t.		A.2.4
		3		5706277.00		737.14	6/2	4/2	2		A.2.4
9	16	1	9566079.86	9164862.26	4.19%	36000.00	6/2	n.v.t.	n.v.t.		A.2.5
		2	9727046.43	9130282.84	6.14%	36000.00	6/2	n.v.t.	n.v.t.		A.2.5
		3		9566079.86		18064.73	6/2	4/2	3		A.2.5

¹ Aantal

² In seconde

³ Feeder 450/ Feeder 800

Uit tabel 5 en tabel 6 volgt dat model 3 het sterkste model is. Dit volgt uit het feit dat dit model altijd in de kortste tijd de oplossing voor het probleem heeft gevonden en voor grotere netwerken als enige model een oplossing vindt. Echter heeft dit model wel iets meer tijd nodig voor het kleinste probleem door de extra constraints, dit is te zien in tabel 5, waar voor 3 havens en 4 vraagparen model 3 het traagst is. Model 1 en model 2 zijn qua tijd in tabel 5 min of meer gelijk aan elkaar. Uit tabel 6 volgt dat de gaten van model 1 twee keer net iets beter zijn dan van model 2. Alles bij elkaar genomen, is model 1 net iets beter dan model 2.

Om de resultaten te berekenen is de heuristiek gepresenteerd in sectie 4.1.2 gebruikt, met behulp hiervan wordt $|\mathcal{R}_k|$ meegegeven aan het model. Uit tabel 5 en tabel 6 volgt dat deze heuristiek een krappe bovengrens geeft, want deze grens wordt 4 keer van de 10 keer bereikt. Om te controleren of de bovengrens niet te krap is, is het model ook uitgevoerd met een vaste waarde voor $|\mathcal{R}_k|$. Deze waarden en de uitkomst van het model staan weergegeven in tabel 7. Alle resultaten zijn berekend met model 3. Naast de verzamelingen van havens en vraagparen die gepresenteerd staan in deze tabel wordt ook voor de verzameling van 9 havens en 16 vraagparen een bovengrens van $|\mathcal{R}_k|$ bereikt. Echter was dit probleem na meer dan 24 uur nog niet opgelost met de volgende instellingen: $|\mathcal{R}_{\text{Feeder 450}}| = 8$ en $|\mathcal{R}_{\text{Feeder 800}}| = 4$.

Tabel 7: Resultaten met grotere bovengrens voor het aantal routes.

Havens ¹	vraagparen ¹	$ \mathcal{R}_k $ ³	Doelfunctiewaarde	Tijd ²	routes ³
5	7	4/3	2568210.29	3.69	0/2
7	10	4/3	4967074.00	97.13	1/1
7	12	8/4	8098503.29	22667.78	2/3
¹ Aantal			³ Feeder 450/ Feeder 800		
² In seconde					

Uit deze tabel kan worden opgemaakt dat in twee van de acht gevallen de bovengrens niet goed was. Namelijk voor 5 havens en 8 vraagparen zijn twee routes voor de Feeder 800 schepen nodig, waar de bovengrens één was. En voor de 7 havens en 12 vraagparen waren drie routes voor de Feeder 800 schepen nodig, waar de bovengrens twee was. In bijlage C staat een verbeterde heuristiek, waarmee dit probleem niet meer voorkomt.

Met het model gepresenteerd in sectie 4 kunnen zogenaamde vlinderroutes worden gecreëerd. Echter gebeurt dit in de praktijk slechts één keer, in het geval van 5 havens en 8 vraagparen. In alle andere routes worden alle havens op de route slechts één keer bezocht. Opmerkelijk is dat overslagen nauwelijks gebruikt worden. Dit gebeurt bij 7 havens en 10 vraagparen nooit tot drie keer bij 9 havens en 16 vraagparen. In tabel 8 is het gemiddelde, mediaan, maximum en minimum van het aantal bezochte havens op een route weergegeven. Deze gegevens staan in deze tabel ook voor de reistijd. Uit de tabel volgt dat het kleinere scheepstype, de Feeder 450, kleinere routes met minder havens en een kleinere reisafstand maakt dan het grote scheepstype, de Feeder 800.

Tabel 8: Karakteristieken van de routes.

	Scheepstype	Gemiddelde	Mediaan	Maximale waarde	Minimale waarde	Standaard afwijking
Aantal havens	Feeder 450	2.54	2	4	2	0.78
	Feeder 800	3.43	3	5	3	0.79
Reisafstand	Feeder 450	1024.54	894	2356	226	588.04
	Feeder 800	2359.00	2378	3324	1189	662.11

5.2 Resultaten voor de clustertechniek

Voor de resultaten van de clustertechniek is de verzameling van 39 havens en 365 vraagparen van het Middellandse Zeegebied gebruikt. Verder is het minimum aantal havens voor een cluster gelijk gezet aan 4, het maximaal aantal havens in een cluster aan 6 en ρ gelijk aan 0.9. Als de clustertechniek hierop wordt toegepast, ontstaat het netwerk dat staat weergegeven in figuur 1. In bijlage A.3.1 wordt verwezen naar het document waarin de exacte tabellen staan hoe de vraag wordt getransporteerd. Iedere kleur representeert een cluster in de eerste clusterlaag. In ieder cluster heeft één haven met een donkere kleur. Deze haven is de hub van het cluster. De lichtgrijze en donkergrijze routes horen bij de eerste hub laag en bij de laatste laag hoort de zwarte route. Goed om op te merken is dat de clusters in de volgende volgorde zijn gevormd: oranje, rood, donkergroen, bruin, paars, lichtgroen, blauw, lichtgrijs, donkergrijs en als laatst zwart. Hieruit volgt dat, zoals verwacht, eerst clusters worden gemaakt met de grootste gemiddelde afstand naar alle andere havens.



Figuur 1: Netwerk voor het Middellandse Zeegebied.

Nadat de clusters gevormd zijn, is voor ieder cluster het bijbehorende model opgelost. Hiervoor wordt gebruik gemaakt van de Feeder heuristiek (zoals gepresenteerd in 4.1.2) om de bovengrens $|\mathcal{R}_k|$ vooraf te bepalen, waarna model 3 het probleem gaat oplossen. In tabel 9 staan

de kosten binnen het cluster, de overslagkosten tussen de clusters en de tijd weergegeven. De kosten binnen het cluster is de uitkomst van het oplossen van het model. Deze kosten zijn alle kosten die binnen het cluster plaatsvinden, zoals overslagen binnen het cluster, de kosten voor het laten varen van de schepen in het cluster en de aanlegkosten. De overslagkosten tussen de clusters zijn de overslagkosten van alle vraag die aankomt in de hub van het cluster en die hier overgeslagen zal worden naar een schip van een ander cluster. Als laatst is de tijd weergegeven die nodig was om deze oplossing te berekenen.

Tabel 9: Kosten clusters.

Cluster	Kosten binnen het cluster ¹	Overslagkosten ¹	Tijd ²
Oranje	5001317.86	577980.00	126.64
Rood	3287300.86	123266.00	8.41
donkergroen	2725593.29	39060.00	3.24
Bruin	2950044.86	27132.00	2.11
Paars	1357396.29	170064.00	73.55
lichtgroen	1967079.00	23904.00	0.05
Blauw	2717427.00	132276.00	1.34
lichtgrijs	7395074.00	805224.00	0.48
donkergrijs	6262949.29	55690.00	7.44
Zwart	12948729.71	375778.00	0.03
Totaal	4661291.22	2330374.00	223.29

¹ In USD

² In seconde

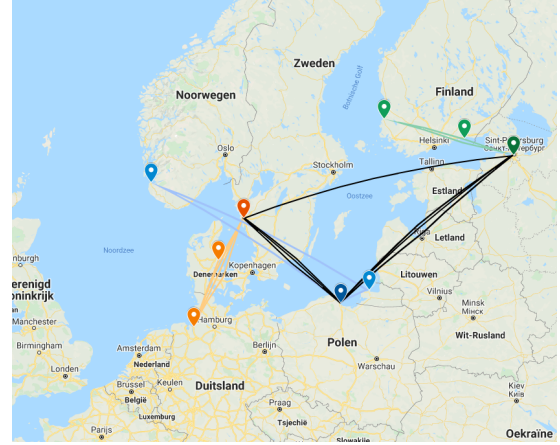
Uit tabel 9 volgt dat de totale kosten voor dit netwerk \$6991665.22 zijn en dat dit berekend wordt in 3 minuten en 43 seconde. Hiernaast waren er nog 16 seconden nodig om de clusters te maken. Overige exacte details over deze oplossing staan weergegeven in bijlage A.

5.3 Prestatie van de clustertechniek

Om de prestatie van de clustertechniek te beoordelen is de verzameling van 9 havens en 16 vraagparen gebruikt. Als deze exact wordt opgelost met het LSNDP model, dan ontstaat het netwerk weergegeven in figuur 2a. Voor de clustertechniek zijn de volgende parameters gebruikt: het minimum en maximum aantal havens gelijk gezet aan 3 en ρ is gelijk aan 0.9. Als de clustertechniek wordt opgelost, ontstaat het netwerk weergegeven in figuur 2b. In tabel 10 staan de kosten van het netwerk en de tijd die nodig is om dit te berekenen, deze tabel is het zelfde opgebouwd als tabel 9. De exacte tabellen staan weergegeven in bijlage A.3.2.



(a) Exacte oplossing



(b) Clustertechniek oplossing

Figuur 2: Netwerk voor het Baltische gebied met 9 havens en 16 vraagparen.

Tabel 10: Kosten clusters.

Cluster	Kosten binnen het cluster ¹	Overslagkosten ¹	Tijd ²
Oranje	4529963.29	264693.00	0.81
Groen	1346456.00	478.00	0.05
Blauw	1519246.00	1911.00	0.13
Zwart	5016589.29	132128.00	0.19
Totaal	12412254.58	399210.00	1.18

¹ In USD

² In seconde

Zoals weergeven in tabel 6 in sectie 5.1 zijn de kosten voor dit netwerk gelijk aan \$9566079.86 als deze wordt opgelost met het LSNDP. In tabel 10 staan alle kosten voor dit netwerk wanneer deze wordt opgelost met de clustertechniek. Hieruit volgt dat de totale kosten gelijk zijn aan \$12811464.58. Dit is een stijging van 33,9%.

Het is belangrijk om op te merken dat de clustertechniek beter werkt als het minimum en maximum aantal havens groter is. Hierdoor zullen er namelijk minder schepen rondvaren die niet vol geladen zijn. In bovenstaande netwerk varen namelijk in het blauwe en groene cluster 2 schepen rond die nooit meer dan 75% van de capaciteit bereiken en het grootste gedeelte van hun tocht zelfs minder dan 50%. Dit is een hoge kostenpost. Ook zijn de overslagkosten erg hoog, dit komt doordat de meeste vracht het netwerk binnenkomt in DEBRV. Hierdoor zijn de overslagkosten om dat cluster uit te gaan erg hoog.

Voor kleinere aantallen havens en vraagparen, moeten ook het minimum en maximum aantal havens in een cluster worden verlaagd om zo toch de clustertechniek toe te kunnen passen. Hierdoor zullen hier relatief nog meer schepen zijn die half leeg rondvaren, wat nog duurder is.

Voor een groter aantal havens en vraagparen is het niet mogelijk om het model op te lossen. Dit komt omdat het aantal variabelen te groot is en hierdoor een out of memory ontstaat. Hierdoor kan ook geen onder ondergrens berekend worden met een relaxatie.

6 Conclusie

In dit verslag zijn verschillende modellen voor het Liner Shipping Network Design Problem (LSNDP) met elkaar vergeleken. Ook is bekeken hoe voor een grotere verzameling aan havens ook een oplossing tot stand kan komen, door gebruik te maken van een clustertechniek. Het doel is om de totale kosten te minimaliseren. Om zo een antwoord op de onderzoeksvraag, *"Hoe kan een routeschema voor de lijnvaart worden gemaakt die de kosten minimaliseert?"*, te verkrijgen.

In totaal zijn er drie verschillende modellen met elkaar vergeleken. Hieruit volgt dat het model met de volgende onderdelen het sterkste model is: route constraints, stroom constraints, overslagconstraint, scheepsvloot constraints, eliminatie van subtours constraints, symmetrie brekende constraints en innerlijke representatie constraints. Om tot deze conclusie te komen zijn verschillende netwerken binnen het Baltische gebied opgelost met alle modellen. Het sterkste model is tot 95,7% sneller dan andere modellen die getest zijn. De zwakkere modellen waren zonder of met minder symmetrie brekende constraints en/of de innerlijke representatie constraints.

Voor de clustertechniek wordt het grote probleem in kleine problemen opgesplitst met behulp van een clustertechniek. Met behulp van deze techniek worden havens die dichtbij elkaar liggen met elkaar geclusterd. Hierdoor is de reisafstand tussen havens niet groot. Het voordeel van deze techniek is dat veel grotere problemen kunnen worden opgelost met het Liner Shipping Network Design Problem. Het nadeel van deze techniek is dat de er veel overslagkosten zijn als een vraag tussen clusters moet worden overgeslagen.

Voor een klein aantal havens met vraagparen kan met het LSNDP het probleem worden opgelost. Voor een groter aantal havens met veel vraagparen kan het probleem worden opgesplitst in clusters met de clustertechniek, waarna voor deze clusters het LSNDP kan worden opgelost. Dit zal bijdragen aan het goedkoop inzetten van de schepen in de lijnvaart. Echter kan voor verder onderzoek het model nog verder uitgebreid worden door een variabele in plaats van vaste snelheid te nemen. Ook kan in verder onderzoek de nieuwe pseudocode (zoals gepresenteerd in bijlage C) voor de bovengrens voor het aantal routes die mee moet worden gegeven om het model te kunnen oplossen worden gebruikt, zodat deze niet meer te laag is. Als laatst kan de doelfunctie uitgebreid worden met de winst die gemaakt wordt bij het verschepen van de vracht.

Referenties

- [1] Richa Agarwal en Özlem Ergun. “Ship Scheduling and Network Design for Cargo Routing in Liner Shipping”. In: *Transportation Science* 42.2 (2008), p. 175–196. DOI: 10.1287/trsc.1070.0205.
- [2] Marie Ameln, Julie Sand Fuglum, Kristian Thun, Henrik Andersson en Magnus Stålhane. “A new formulation for the liner shipping network design problem”. In: *International Transactions in Operational Research* 28.2 (2019), p. 638–659. DOI: 10.1111/itor.12659.
- [3] Berit D. Brouer, J. Fernando Alvarez, Christian E. M. Plum, David Pisinger en Mikkel M. Sigurd. “A Base Integer Programming Model and Benchmark Suite for Liner-Shipping Network Design”. In: *Transportation Science* 48.2 (2014), p. 281–312. DOI: 10.1287/trsc.2013.0471.
- [4] Jeng-Fung Chen. “A hybrid heuristic for the uncapacitated single allocation hub location problem”. In: *Omega* 35.2 (2007), p. 211–220. DOI: 10.1016/j.omega.2005.05.004.
- [5] Drewry. “Seaborne trade annual report 2013”. In: *Technical Report, Drewry Maritime Research* (2014).
- [6] S.k. Jacobsen en O.b.g. Madsen. “A comparative study of heuristics for a two-level routing-location problem”. In: *European Journal of Operational Research* 5.6 (1980), p. 378–387. DOI: 10.1016/0377-2217(80)90124-1.
- [7] G. Laporte, M. Gendreau, J-Y. Potvin en F. Semet. “Classical and modern heuristics for the vehicle routing problem”. In: *International Transactions in Operational Research* 7.4-5 (2000), p. 285–300. DOI: 10.1111/j.1475-3995.2000.tb00200.x.
- [8] S.A. Lawrence. “International Sea Transport: The Years Ahead”. In: *Lexington Books, Lexington, MA* (1972).
- [9] Oli B.g. Madsen. “Methods for solving combined two level location-routing problems of realistic dimensions”. In: *European Journal of Operational Research* 12.3 (1983), p. 295–301. DOI: 10.1016/0377-2217(83)90199-6.
- [10] Christian E.m. Plum, David Pisinger en Mikkel M. Sigurd. “A service flow model for the liner shipping network design problem”. In: *European Journal of Operational Research* 235.2 (2014), p. 378–386. DOI: 10.1016/j.ejor.2013.10.057.
- [11] Line Blander Reinhardt en David Pisinger. “A branch and cut algorithm for the container shipping network design problem”. In: *Flexible Services and Manufacturing Journal* 24.3 (2011), p. 349–374. DOI: 10.1007/s10696-011-9105-4.

Bijlage A Kaarten en routes

A.1 Kaarten voor de verschillende gebieden

A.1.1 Alle twaalf havens in het Baltische gebied



Figuur A1: Kaart met alle twaalf havens in het Baltische gebied.

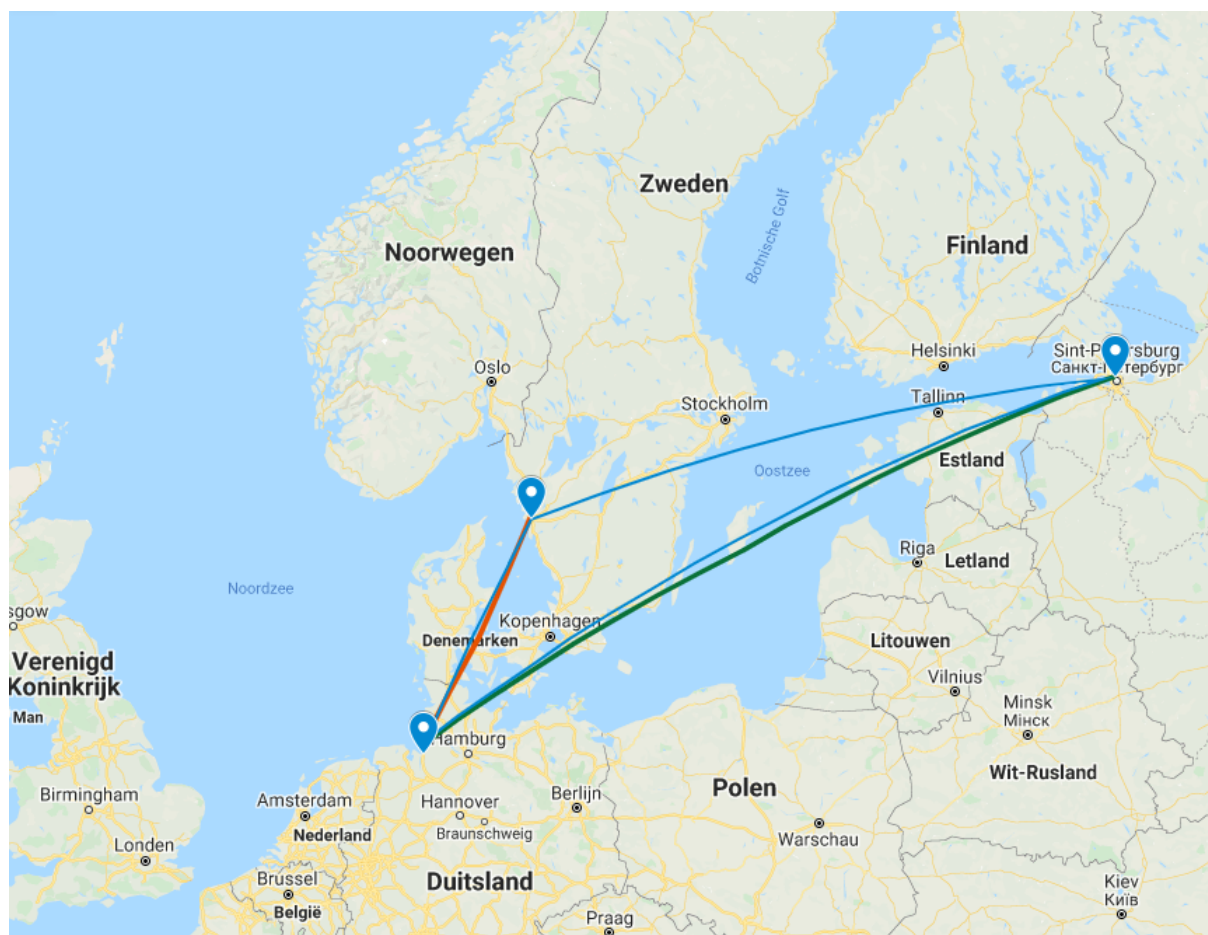
A.1.2 Alle 39 havens in het Middellandse Zeegebied



Figuur A2: Kaart met alle 39 havens in het Middellandse Zeegebied.

A.2 Exact opgeloste routes voor het Baltische gebied

A.2.1 Route met drie havens en vier vraagparen



Figuur A3: Kaart met de routes tussen drie havens.

In dit netwerk zijn vier Feeder 450 schepen beschikbaar en twee Feeder 800 schepen. De vraag die verscheept moet worden staat weergegeven in tabel A1. Uit de vloot van Feeder 450 worden drie schepen gebruikt. Het eerste schip vaart de eerste route (groen). De tweede en derde boot varen respectievelijk de tweede en derde route (beide oranje). Uit de vloot van de Feeder 800 wordt één boot gebruikt. Deze boot vaart de eerste route (blauw). In tabel A2 wordt weergegeven hoe de vracht wordt verscheept. Er vindt één overslag plaats, deze is weergegeven in tabel A3.

Tabel A1: Vraag in het netwerk.

Oorsprongshaven	Bestemmingshaven	FFE per week
SEGOT	DEBRV	660
DEBRV	SEGOT	597
DEBRV	RULED	1215
RULED	DEBRV	298

Tabel A2: Vracht verplaatsingen voor het netwerk.

Scheepstype	Route	Transport		Vraag		Lading ¹
		Starthaven	Eindhaven	Oorsprongshaven	Bestemingshaven	
Feeder 450	1	DEBRV	RULED	DEBRV	RULED	450
	2	DEBRV	SEGOT	DEBRV	SEGOT	450
	3	DEBRV	SEGOT	DEBRV	SEGOT	147
		SEGOT	DEBRV	RULED	DEBRV	158
Feeder 800	1	SEGOT	DEBRV	SEGOT	DEBRV	660
		DEBRV	RULED	DEBRV	RULED	765
		RULED	SEGOT	RULED	DEBRV	298
		SEGOT	DEBRV	RULED	DEBRV	140

¹ In FFE per week

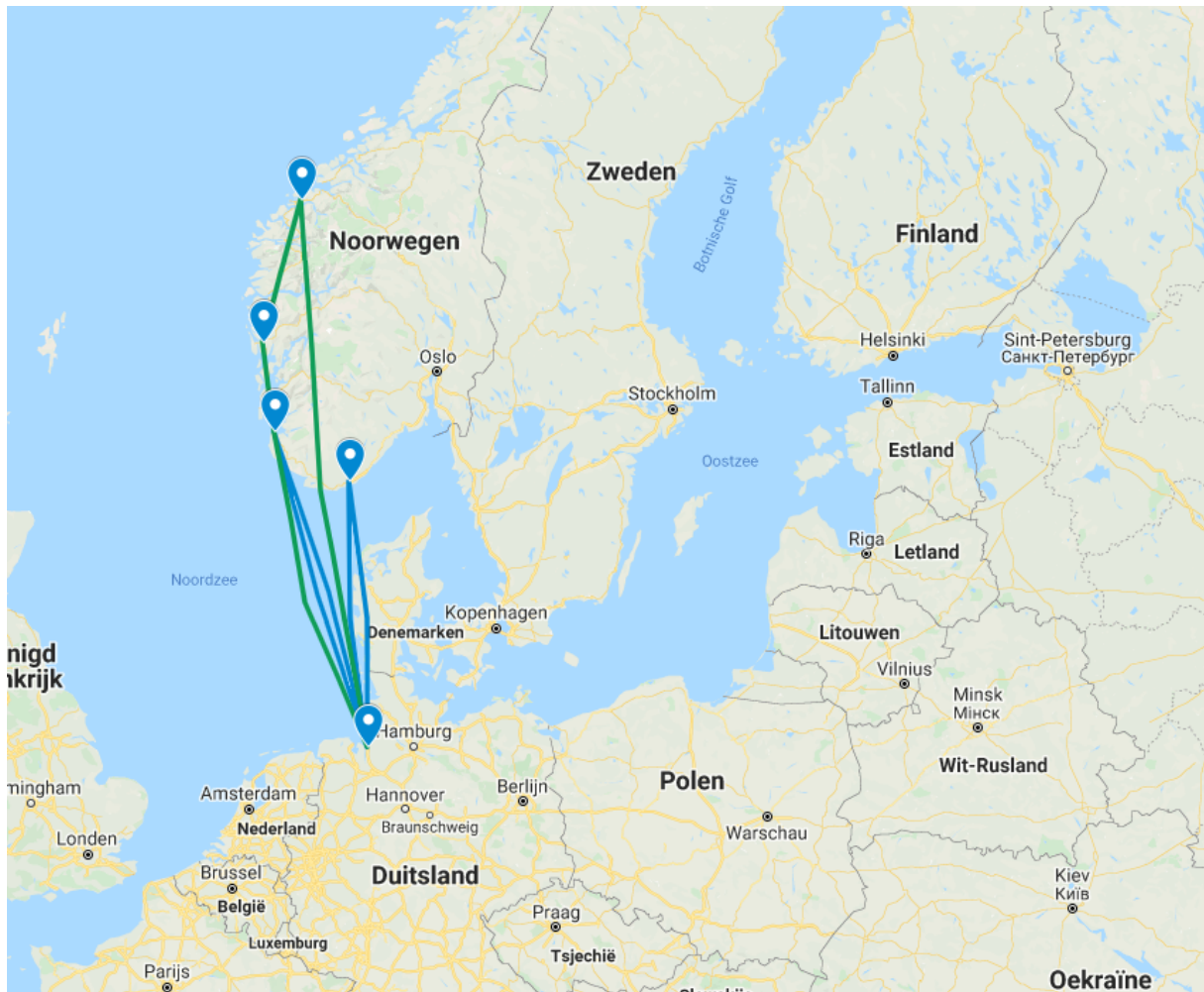
Tabel A3: Overslag in het netwerk.

Oorsprongshaven	Vraag		Overslaghaven	Lading ¹	Aankomend schip ²	Vertrekkend schip ²
	Bestemmingshaven					
RULED	DEBRV	SEGOT		158	Feeder 800 (1)	Feeder 450 (3)

¹ In FFE per week

² Tussen de haakjes staat het route nummer

A.2.2 Route met vijf havens en acht vraagparen



Figuur A4: Kaart met de routes tussen 5 havens.

In dit netwerk zijn vier Feeder 450 schepen beschikbaar en twee Feeder 800 schepen. De vraag die verscheept moet worden staat weergegeven in tabel A4. Uit de vloot van Feeder 450 wordt één schip gebruikt. Deze boot vaart de eerste route (blauw). Uit de vloot van de Feeder 800 wordt ook één boot gebruikt. Deze boot vaart de eerste route (groen). In tabel A5 wordt weergegeven hoe de vracht wordt verscheept. Er vindt één overslag plaats, deze is weergegeven in tabel A6.

Tabel A4: Vraag in het netwerk.

Oorsprongshaven	Bestemmingshaven	FFE per week
DEBRV	NOSVG	365
DEBRV	NOAES	310
DEBRV	NOBGO	317
NOKRS	DEBRV	316
NOBGO	DEBRV	337
NOAES	DEBRV	350
NOSVG	DEBRV	332
DEBRV	NOKRS	316

Tabel A5: Vracht verplaatsingen voor het netwerk.

Scheepstype	Route	Transport		Vraag		Lading ¹
		Starthaven	Eindhaven	Oorsprongshaven	Bestemmingshaven	
Feeder 450	1	DEBRV	NOSVG	DEBRV	NOSVG	365
		NOKRS	DEBRV	NOKRS	DEBRV	316
		NOSVG	DEBRV	NOBGO	DEBRV	219
Feeder 800	1	DEBRV	NOAES	DEBRV	NOAES	310
		DEBRV	NOAES	DEBRV	NOBGO	317
		NOAES	NOBGO	DEBRV	NOBGO	317
		NOBGO	NOSVG	NOBGO	DEBRV	337
		NOSVG	DEBRV	NOBGO	DEBRV	118
		NOAES	NOBGO	NOAES	DEBRV	350
		NOBGO	NOSVG	NOAES	DEBRV	350
		NOSVG	DEBRV	NOAES	DEBRV	350
		NOSVG	DEBRV	NOSVG	DEBRV	332

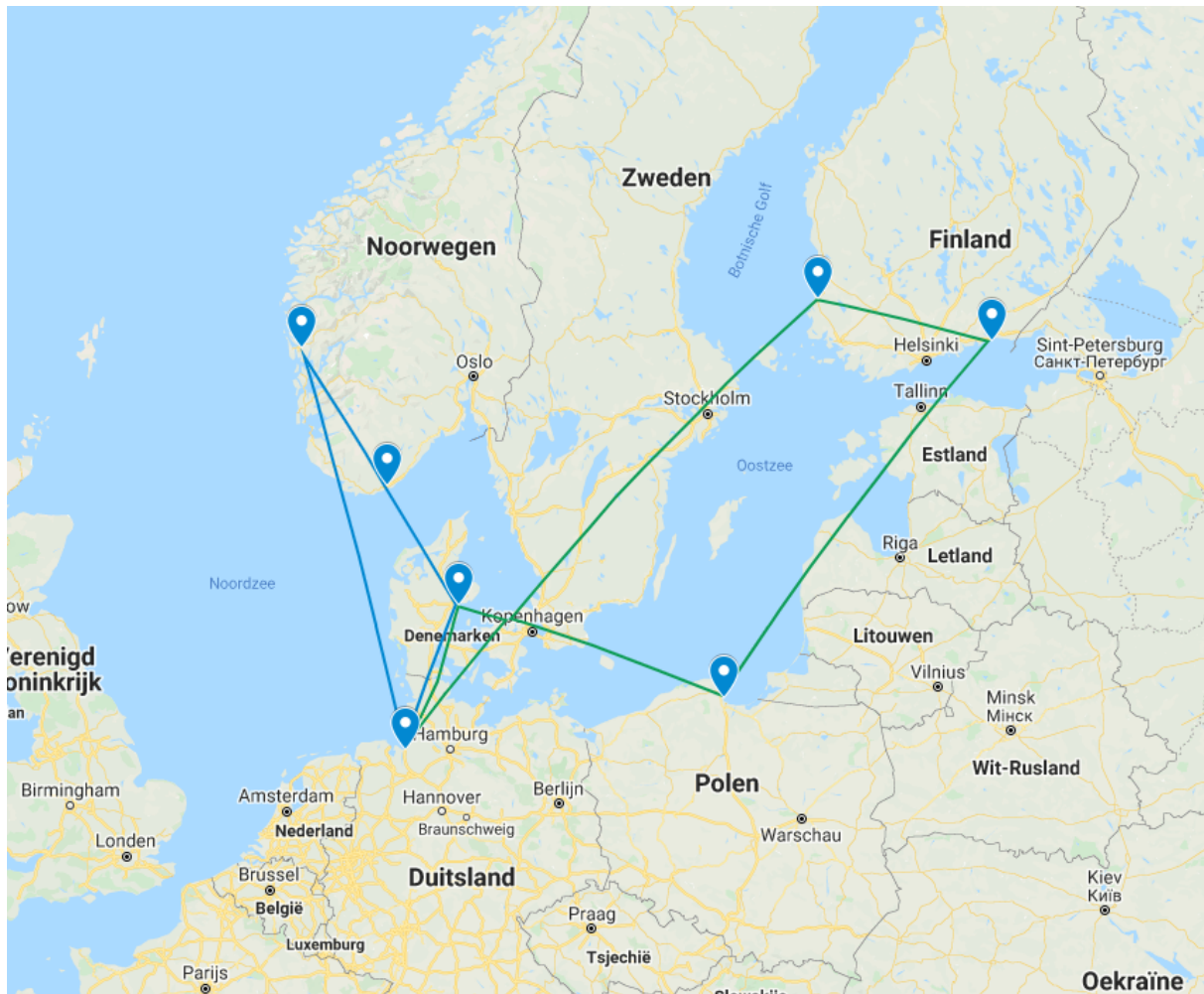
¹ In FFE per week

Tabel A6: Overslag in het netwerk.

Vraag		Overslaghaven	Lading ¹	Aankomend schip ²	Vertrekkend schip ²
Oorsprongshaven	Bestemmingshaven				
NOBGO	DEBRV	NOSVG	219	Feeder 800 (1)	Feeder 450 (1)

¹ In FFE per week² Tussen de haakjes staat het route nummer

A.2.3 Route met zeven havens en tien vraagparen



Figuur A5: Kaart met de routes tussen 7 havens.

In dit netwerk zijn vier Feeder 450 schepen beschikbaar en twee Feeder 800 schepen. De vraag die verscheept moet worden staat weergegeven in tabel A7. Uit de vloot van Feeder 450 wordt één schip gebruikt. Dit schip vaart de eerste route (blauw). Uit de vloot van de Feeder 800 worden twee boten gebruikt. Beide boten varen dezelfde route, de eerste route (groen). In tabel A8 wordt weergegeven hoe de vracht wordt verscheept. Er vindt geen overslag plaats.

Tabel A7: Vraag in het netwerk.

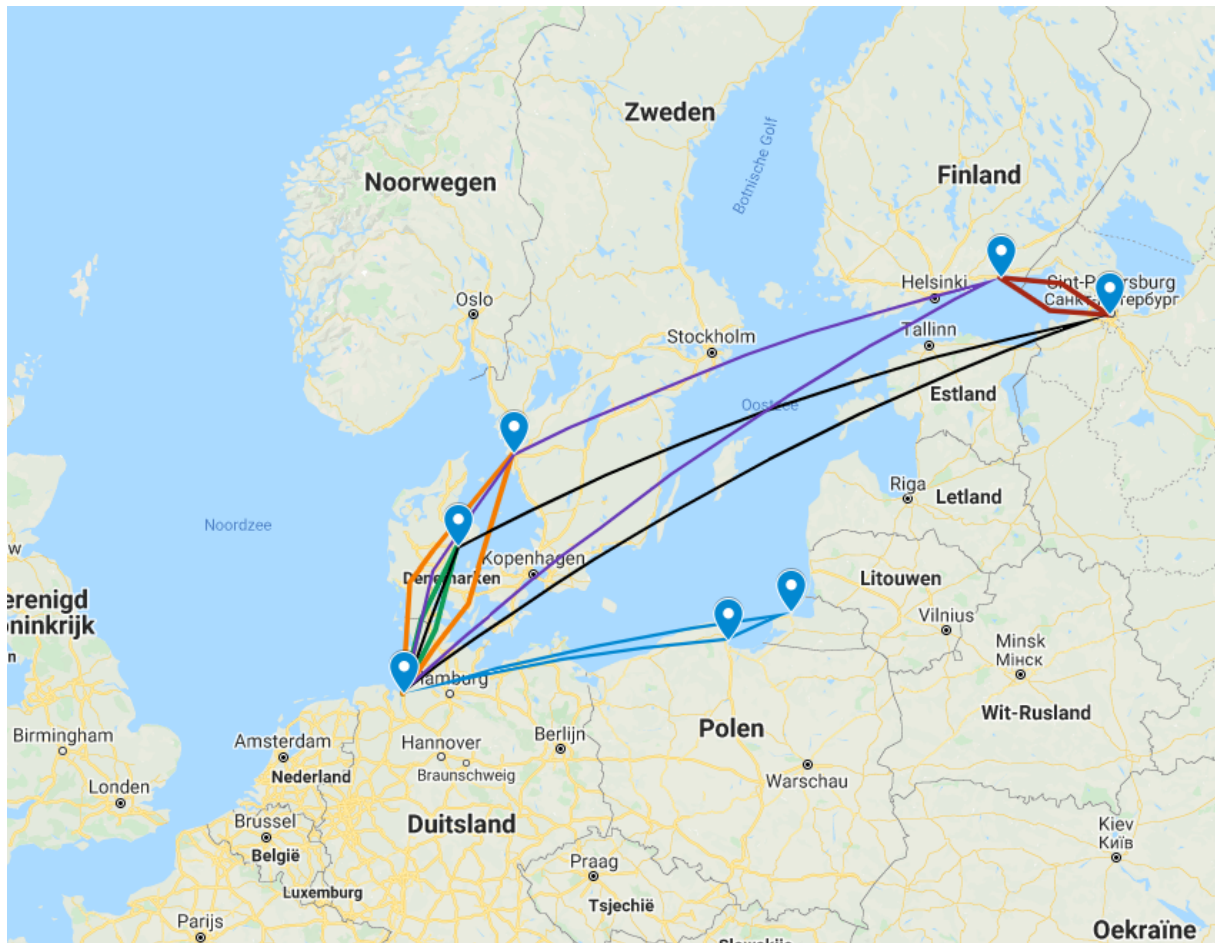
Oorsprongshaven	Bestemmingshaven	FFE per week
FIRAU	DEBRV	77
DEBRV	DKAAR	456
DEBRV	PLGDY	98
DEBRV	FIRAU	18
NOBGO	DEBRV	37
DEBRV	FIKTK	187
PLGDY	DEBRV	231
FIKTK	DEBRV	162
DKAAR	DEBRV	397
DEBRV	NOKRS	6

Tabel A8: Vracht verplaatsingen voor het netwerk.

Scheepstype	Route	Transport		Vraag		Lading ¹
		Starthaven	Eindhoven	Oorsprongshaven	Bestemmingshaven	
Feeder 450	1	DKAAR	DEBRV	NOBGO	DEBRV	37
		NOBGO	NOKRS	NOBGO	DEBRV	37
		NOKRS	DKAAR	NOBGO	DEBRV	37
		DKAAR	DEBRV	DKAAR	DEBRV	397
		DEBRV	NOBGO	DEBRV	NOKRS	6
		NOBGO	NOKRS	DEBRV	NOKRS	6
Feeder 800	1	FIRAU	DEBRV	FIRAU	DEBRV	77
		DEBRV	DKAAR	DEBRV	DKAAR	456
		DEBRV	DKAAR	DEBRV	PLGDY	98
		DKAAR	PLGDY	DEBRV	PLGDY	98
		DEBRV	DKAAR	DEBRV	FIRAU	18
		DKAAR	PLGDY	DEBRV	FIRAU	18
		FIKTK	FIRAU	DEBRV	FIRAU	18
		PLGDY	FIKTK	DEBRV	FIRAU	18
		DEBRV	DKAAR	DEBRV	FIKTK	187
		DKAAR	PLGDY	DEBRV	FIKTK	187
		PLGDY	FIKTK	DEBRV	FIKTK	187
		FIKTK	FIRAU	PLGDY	DEBRV	231
		FIRAU	DEBRV	PLGDY	DEBRV	231
		PLGDY	FIKTK	PLGDY	DEBRV	231
		FIKTK	FIRAU	FIKTK	DEBRV	162
		FIRAU	DEBRV	FIKTK	DEBRV	162

¹ In FFE per week

A.2.4 Route met zeven havens en twaalf vraagparen



Figuur A6: Kaart met de routes tussen 7 havens.

In dit netwerk zijn acht Feeder 450 schepen beschikbaar en vier Feeder 800 schepen. De vraag die verscheept moet worden staat weergegeven in tabel A9. Uit de vloot van Feeder 450 worden vier schepen gebruikt. Het eerste schip vaart de eerste route (blauw). De tweede boot vaart de tweede route (groen). De derde boot vaart de derde route (oranje). En de vierde boot vaart de vierde route (rood). Uit de vloot van de Feeder 800 worden drie boten gebruikt. De eerste en tweede boot varen de eerste route (zwart). En de derde boot vaart de tweede route (paars). In tabel A10 wordt weergegeven hoe de vracht wordt verscheept. Er vinden twee overslagen plaats, deze zijn weergegeven in tabel A11.

Tabel A9: Vraag in het netwerk.

Oorsprongshaven	Bestemmingshaven	FFE per week
DEBRV	DKAAR	456
RUKGD	DEBRV	7
DEBRV	PLGDY	98
SEGOT	DEBRV	660
DEBRV	RUKGD	268
DEBRV	FIKTK	187
PLGDY	DEBRV	231
DEBRV	SEGOT	597
FIKTK	DEBRV	162
DKAAR	DEBRV	397
DEBRV	RULED	1215
RULED	DEBRV	298

Tabel A10: Vracht verplaatsingen voor het netwerk.

Scheepstype	Route	Transport		Vraag		Lading ¹
		Starthaven	Eindhaven	Oorsprongshaven	Bestemmingshaven	
Feeder 450	1	PLGDY	DEBRV	RUKGD	DEBRV	7
		RUKGD	PLGDY	RUKGD	DEBRV	7
		DEBRV	RUKGD	DEBRV	PLGDY	98
		RUKGD	PLGDY	DEBRV	PLGDY	98
		DEBRV	RUKGD	DEBRV	RUKGD	268
		PLGDY	DEBRV	PLGDY	DEBRV	231
	2	DEBRV	DKAAR	DEBRV	DKAAR	450
	3	DEBRV	SEGOT	DEBRV	SEGOT	450
	4	FIKTK	RULED	FIKTK	DEBRV	22
		FIKTK	RULED	DEBRV	RULED	421
Feeder 800	1	DEBRV	RULED	DEBRV	DKAAR	6
		RULED	DKAAR	DEBRV	DKAAR	6
		DKAAR	DEBRV	FIKTK	DEBRV	22
		RULED	DKAAR	FIKTK	DEBRV	22
		DKAAR	DEBRV	DKAAR	DEBRV	397
		DEBRV	RULED	DEBRV	RULED	794
		DKAAR	DEBRV	RULED	DEBRV	298
		RULED	DKAAR	RULED	DEBRV	298
	2	SEGOT	DEBRV	SEGOT	DEBRV	660
		DEBRV	FIKTK	DEBRV	FIKTK	187
		DEBRV	FIKTK	DEBRV	SEGOT	147
		FIKTK	SEGOT	DEBRV	SEGOT	147
		FIKTK	SEGOT	FIKTK	DEBRV	140
		SEGOT	DEBRV	FIKTK	DEBRV	140
		DEBRV	FIKTK	DEBRV	RULED	421

¹ In FFE per week

Tabel A11: Overslag in het netwerk.

Vraag		Overslaghaven	Lading ¹	Aankomend schip ²	Vertrekkend schip ²
Oorsprongshaven	Bestemmingshaven				
DEBRV	RULED	FIKTK	421	Feeder 800 (2)	Feeder 450 (4)
FIKTK	DEBRV	RULED	22	Feeder 450 (4)	Feeder 800 (1)

¹ In FFE per week² Tussen de haakjes staat het route nummer

A.2.5 Route met negen havens en zestien vraagparen



Figuur A7: Kaart met de routes tussen 9 havens.

In dit netwerk zijn acht Feeder 450 schepen beschikbaar en vier Feeder 800 schepen. De vraag die verscheept moet worden staat weergegeven in tabel A12. Uit de vloot van Feeder 450 worden vier schepen gebruikt. Het eerste schip vaart de eerste route (blauw). De tweede boot vaart de tweede route (groen). De derde boot vaart de derde route (oranje). En de vierde boot vaart de vierde route (rood). Uit de vloot van de Feeder 800 worden drie boten gebruikt. De eerste en tweede boot varen de eerste route (zwart). En de derde boot vaart de tweede route (paars). In tabel A13 wordt weergegeven hoe de vracht wordt verscheept. Er vinden drie overslagen plaats, deze zijn weergegeven in tabel A14.

Tabel A12: Vraag in het netwerk.

Oorsprongshaven	Bestemmingshaven	FFE per week
FIRAU	DEBRV	77
DEBRV	DKAAR	456
DEBRV	NOSVG	65
RUKGD	DEBRV	7
DEBRV	PLGDY	98
SEGOT	DEBRV	660
DEBRV	RUKGD	268
DEBRV	FIRAU	18
DEBRV	FIKTK	187
PLGDY	DEBRV	231
DEBRV	SEGOT	597
NOSVG	DEBRV	32
FIKTK	DEBRV	162
DKAAR	DEBRV	397
DEBRV	RULED	1215
RULED	DEBRV	298

Tabel A13: Vracht verplaatsingen voor het netwerk.

Scheepstype	Route	Transport		Vraag		Lading ¹
		Starthaven	Eindhaven	Oorsprongshaven	Bestemmingshaven	
Feeder 450	1	DEBRV	DKAAR	DEBRV	DKAAR	71
		PLGDY	DEBRV	RUKGD	DEBRV	7
		RUKGD	PLGDY	RUKGD	DEBRV	7
		DEBRV	DKAAR	DEBRV	PLGDY	98
		DKAAR	RUKGD	DEBRV	PLGDY	98
		RUKGD	PLGDY	DEBRV	PLGDY	98
		DEBRV	DKAAR	DEBRV	RUKGD	268
		DKAAR	RUKGD	DEBRV	RUKGD	268
		PLGDY	DEBRV	PLGDY	DEBRV	231
	2	DEBRV	NOSVG	DEBRV	DKAAR	385
		NOSVG	DKAAR	DEBRV	DKAAR	385
		DEBRV	NOSVG	DEBRV	NOSVG	65
		DKAAR	DEBRV	NOSVG	DEBRV	32
		NOSVG	DKAAR	NOSVG	DEBRV	32
		DKAAR	DEBRV	DKAAR	DEBRV	397
	3	DEBRV	SEGOT	DEBRV	SEGOT	450
		SEGOT	DEBRV	FIKTK	DEBRV	5
	4	FIKTK	RULED	FIKTK	DEBRV	17
		FIKTK	RULED	DEBRV	RULED	433
Feeder 800	1	FIRAU	DEBRV	FIRAU	DEBRV	77
		DEBRV	RULED	DEBRV	FIRAU	18
		RULED	FIRAU	DEBRV	FIRAU	18
		FIRAU	DEBRV	FIKTK	DEBRV	17
		RULED	FIRAU	FIKTK	DEBRV	17
		DEBRV	RULED	DEBRV	RULED	782
		FIRAU	DEBRV	RULED	DEBRV	298
		RULED	FIRAU	RULED	DEBRV	298
	2	SEGOT	DEBRV	SEGOT	DEBRV	660
		DEBRV	FIKTK	DEBRV	FIKTK	187
		DEBRV	FIKTK	DEBRV	SEGOT	147
		FIKTK	SEGOT	DEBRV	SEGOT	147
		FIKTK	SEGOT	FIKTK	DEBRV	145
		SEGOT	DEBRV	FIKTK	DEBRV	140
		DEBRV	FIKTK	DEBRV	RULED	433

¹ In FFE per week

Tabel A14: Overslag in het netwerk.

Vraag		Overslaghaven	Lading ¹	Aankomend schip ²	Vertrekkend schip ²
Oorsprongshaven	Bestemmingshaven				
FIKTK	DEBRV	SEGOT	5	Feeder 800 (2)	Feeder 450 (3)
DEBRV	RULED	FIKTK	433	Feeder 800 (2)	Feeder 450 (4)
FIKTK	DEBRV	RULED	17	Feeder 450 (4)	Feeder 800 (1)

¹ In FFE per week² Tussen de haakjes staat het route nummer

A.3 Cluster oplossing

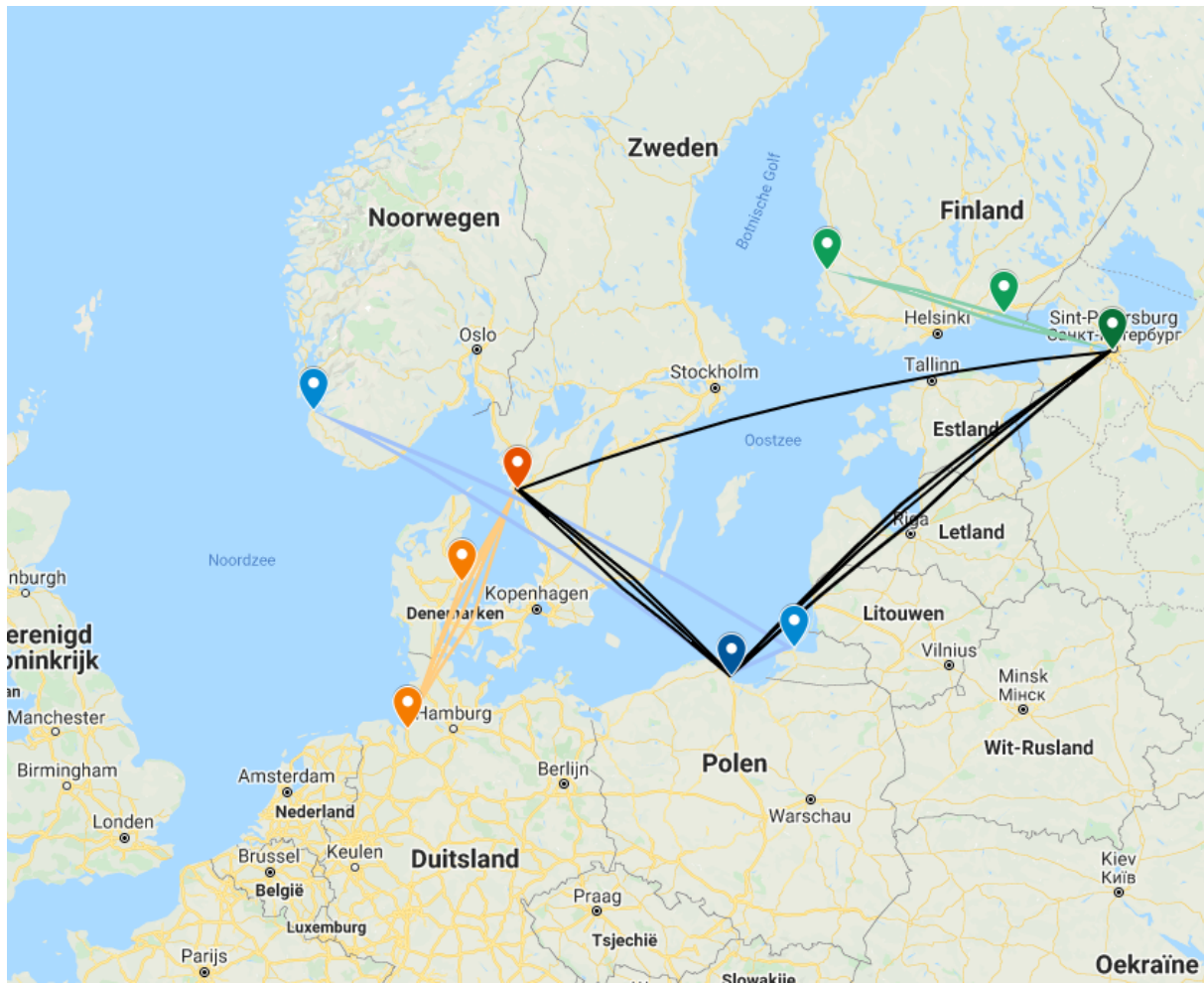
A.3.1 Route met 39 havens en 365 vraagparen in het Middellandse Zeegebied



Figuur A8: Kaart met de routes tussen 39 havens.

De exacte tabellen van dit netwerk staan weergegeven in het document "Routes voor het Middellandse Zeegebied." Deze tabellen staan in dit document in plaats van in deze bijlage, omdat dit document 39 pagina's lang is.

A.3.2 Route met 9 havens en 16 vraagparen in het Baltische gebied



Figuur A9: Kaart met de routes tussen 9 havens.

In dit netwerk zijn onbeperkt veel schepen aanwezig van de volgende typen: Feeder 450 en Feeder 800. De vraag die verscheept moet worden staat weergegeven in tabel A12. De clusters die gevormd worden door de clustertechniek staan weergegeven in figuur A9. Voor ieder cluster staan drie tabellen weergegeven. In de eerste tabel staat weergegeven hoe de vraagparen binnen het cluster zijn opgebouwd. In de eerste en tweede kolom staat de vraagpaar voor het cluster, in de derde en vierde kolom staat het vraagpaar zoals deze origineel was. In de tweede tabel wordt weergegeven hoe de vracht wordt verscheept. Als er een of meerdere overslag(en) plaats vinden binnen het netwerk staan deze weergegeven in de derde tabel. Als er geen overslagen zijn, is er ook geen derde tabel.

Het eerste cluster: oranje

In dit cluster worden 3 Feeder 450 schepen gebruikt. Het eerste schip vaart de eerste route, het tweede schip vaart de tweede route en derde schip vaart de derde route. Uit de vloot van Feeder 800 schepen worden 2 schepen gebruikt. Het eerste schip vaart de eerste route en het tweede schip vaart de tweede route.

Tabel A15: De nieuwe en oorspronkelijke vraag.

Nieuwe oorsprongshaven	Nieuwe bestemmingshaven	Oude oorsprongshaven	Oude bestemmingshaven	FFE
DEBRV	DKAAR	DEBRV	DKAAR	456
DEBRV	SEGOT	DEBRV	NOSVG	65
		DEBRV	PLGDY	98
		DEBRV	RUKGD	268
		DEBRV	FIRAU	18
		DEBRV	FIKTK	187
		DEBRV	SEGOT	597
		DEBRV	RULED	1215
DKAAR	DEBRV	DKAAR	DEBRV	397
SEGOT	DEBRV	FIRAU	DEBRV	77
		RUKGD	DEBRV	7
		SEGOT	DEBRV	660
		PLGDY	DEBRV	231
		NOSVG	DEBRV	32
		FIKTK	DEBRV	162
		RULED	DEBRV	298

Tabel A16: Vracht verplaatsingen voor het netwerk.

Scheepstype	Route	Transport		Vraag		Lading ¹
		Starthaven	Eindhaven	Oorsprongshaven	Bestemmingshaven	
Feeder 450	1	DEBRV	SEGOT	DEBRV	SEGOT	450
	2	DEBRV	SEGOT	DEBRV	SEGOT	450
	3	DEBRV	SEGOT	DEBRV	SEGOT	450
		SEGOT	DEBRV	DKAAR	DEBRV	264
Feeder 800	1	SEGOT	DEBRV	SEGOT	DEBRV	667
		DEBRV	DKAAR	DEBRV	DKAAR	456
		DEBRV	DKAAR	DEBRV	SEGOT	298
		DKAAR	SEGOT	DEBRV	SEGOT	298
		DKAAR	SEGOT	DKAAR	DEBRV	397
		SEGOT	DEBRV	DKAAR	DEBRV	133
	2	DKAAR	DEBRV	SEGOT	DEBRV	800
		SEGOT	DKAAR	SEGOT	DEBRV	800
		DEBRV	SEGOT	DEBRV	SEGOT	800

¹ In FFE per week

Tabel A17: Overslag in het netwerk.

Vraag		Overslaghaven	Lading ¹	Aankomend schip ²	Vertrekkend schip ²
Oorsprongshaven	Bestemmingshaven				
DKAAR	DEBRV	SEGOT	264	Feeder 800 (2)	Feeder 450 (3)

¹ In FFE per week

² Tussen de haakjes staat het route nummer

Het tweede cluster: groen

In dit cluster wordt 1 Feeder 450 schip gebruikt. Dit schip vaart de eerste route. Uit de vloot van Feeder 800 schepen wordt geen schip gebruikt.

Tabel A18: De nieuwe en oorspronkelijke vraag.

Nieuwe oorsprongshaven	Nieuwe bestemmingshaven	Oude oorsprongshaven	Oude bestemmingshaven	FFE
FIKTK	RULED	FIKTK	DEBRV	162
FIRAU	RULED	FIRAU	DEBRV	77
RULED	FIKTK	DEBRV	FIKTK	187
RULED	FIRAU	DEBRV	FIRAU	18

Tabel A19: Vracht verplaatsingen voor het netwerk.

Scheepstype	Route	Transport		Vraag		Lading ¹
		Starthaven	Eindhaven	Oorsprongshaven	Bestemmingshaven	
Feeder 450	1	FIRAU	RULED	FIRAU	RULED	77
		FIKTK	FIRAU	RULED	FIRAU	18
		RULED	FIKTK	RULED	FIRAU	18
		RULED	FIKTK	RULED	FIKTK	187
		FIKTK	FIRAU	FIKTK	RULED	162
		FIRAU	RULED	FIKTK	RULED	162

¹ In FFE per week

Het derde cluster: blauw

In dit cluster worden 1 Feeder 450 schip gebruikt. Dit schip vaart de eerste route. Uit de vloot van Feeder 800 schepen worden geen schip gebruikt.

Tabel A20: De nieuwe en oorspronkelijke vraag.

Nieuwe oorsprongshaven	Nieuwe bestemmingshaven	Oude oorsprongshaven	Oude bestemmingshaven	FFE
NOSVG	PLGDY	NOSVG	DEBRV	32
PLGDY	NOSVG	DEBRV	NOSVG	65
PLGDY	RUKGD	DEBRV	RUKGD	268
RUKGD	PLGDY	RUKGD	DEBRV	7

Tabel A21: Vracht verplaatsingen voor het netwerk.

Scheepstype	Route	Transport		Vraag		Lading ¹
		Starthaven	Eindhaven	Oorsprongshaven	Bestemmingshaven	
Feeder 450	1	PLGDY	NOSVG	PLGDY	NOSVG	65
		RUKGD	PLGDY	RUKGD	PLGDY	7
		NOSVG	RUKGD	PLGDY	RUKGD	268
		PLGDY	NOSVG	PLGDY	RUKGD	268
		NOSVG	RUKGD	NOSVG	PLGDY	32
		RUKGD	PLGDY	NOSVG	PLGDY	32

¹ In FFE per week

Het vierde cluster: zwart

In dit cluster wordt 1 Feeder 450 schip gebruikt. Dit schip vaart de eerste route. Uit de vloot van Feeder 800 schepen worden 2 schepen gebruikt. Het eerste schip vaart de eerste route en het tweede schip vaart de tweede route.

Tabel A22: De nieuwe en oorspronkelijke vraag.

Nieuwe oorsprongshaven	Nieuwe bestemmingshaven	Oude oorsprongshaven	Oude bestemmingshaven	FFE
PLGDY	SEGOT	RUKGD	DEBRV	7
		PLGDY	DEBRV	231
		NOSVG	DEBRV	32
RULED	SEGOT	FIRAU	DEBRV	77
		FIKTK	DEBRV	162
		RULED	DEBRV	298
SEGOT	PLGDY	DEBRV	NOSVG	65
		DEBRV	PLGDY	98
		DEBRV	RUKGD	268
SEGOT	RULED	DEBRV	FIRAU	18
		DEBRV	FIKTK	187
		DEBRV	RULED	1215

Tabel A23: Vracht verplaatsingen voor het netwerk.

Scheepstype	Route	Transport		Vraag		Lading ¹
		Starthaven	Eindhaven	Oorsprongshaven	Bestemmingshaven	
Feeder 450	1	PLGDY	SEGOT	RULED	SEGOT	7
		SEGOT	PLGDY	SEGOT	PLGDY	431
Feeder 800	1	PLGDY	SEGOT	RULED	SEGOT	530
		RULED	PLGDY	RULED	SEGOT	537
		PLGDY	SEGOT	PLGDY	SEGOT	270
		SEGOT	RULED	SEGOT	RULED	620
		SEGOT	RULED	SEGOT	RULED	800

¹ In FFE per week

Tabel A24: Overslag in het netwerk.

Vraag		Overslaghaven	Lading ¹	Aankomend schip ²	Vertrekkend schip ²
Oorsprongshaven	Bestemmingshaven				
RULED	SEGOT	PLGDY	7	Feeder 800 (1)	Feeder 450 (1)

¹ In FFE per week

² Tussen de haakjes staat het route nummer

Bijlage B Verbeteringen Model

In deze bijlage worden alle veranderingen aan het model vergeleken met het Liner Shipping Network Design Problem van Ameln e.a. (2019). Ook wordt gemotiveerd waarom deze verandering zijn doorgevoerd in dit verslag.

De notatie van het model van Ameln e.a. (2019) is niet overal even duidelijk. Deze is daarom in dit verslag aangepast. Een voorbeeld hiervan staat in constraint (B1). Deze constraint impliceert dat dit moet gelden voor alle combinaties (i, j) van havens, echter geldt deze constraint alleen voor de combinaties (i, j) waartussen zich ook een lijn bevindt. Om dit duidelijker te maken zijn alle constraints waarin dit gebeurt verbeterd. Als voorbeeld is constraint (B2) de verbeterde versie van constraint (B1).

$$\sum_{i \in \mathcal{N}} x_{ijk r} - y_{kr} \leq 0 \quad j \in \mathcal{N}, k \in \mathcal{K}, r \in \mathcal{R}_k \quad (\text{B1})$$

$$\sum_{i \in \mathcal{N} | (i, j) \in \mathcal{A}} x_{ijk r} - y_{kr} \leq 0 \quad j \in \mathcal{N}, k \in \mathcal{K}, r \in \mathcal{R}_k \quad (\text{B2})$$

In het model van Ameln e.a. (2019) stond constraint (B3), deze is in het model van dit verslag vervangen door constraint (B4). In de nieuwe versie van de constraint is er rekening gehouden met de tijd schalingsfactor van T_{ijk} , in dit verslag wordt T_{ijk} gemeten in uren. Door de reistijd in uren te delen door $24 * 7 = 168$ wordt met constraint (B4) het wekelijkse bezoek aan iedere haven gegarandeerd.

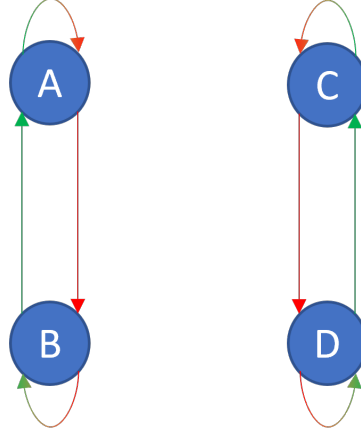
$$w_{kr} \geq \sum_{i \in \mathcal{N}^U} \sum_{j \in \mathcal{N}^U} T_{ijk} (x_{ijk r} + X_{P+i, P+j, kr}) \quad k \in \mathcal{K}, r \in \mathcal{R}_k \quad (\text{B3})$$

$$w_{kr} \geq \sum_{i \in \mathcal{N}^U} \sum_{j \in \mathcal{N}^U} \frac{T_{ijk}}{168} (x_{ijk r} + X_{P+i, P+j, kr}) \quad k \in \mathcal{K}, r \in \mathcal{R}_k \quad (\text{B4})$$

Als laatst kunnen in het model van Ameln e.a. (2019) stromen van goederen ontstaan die niet toegestaan zijn. Eerst zal er een voorbeeld worden uitgewerkt van een route en de bijbehorende stromen die kunnen ontstaan in het model van Ameln e.a. (2019), daarna zal de oplossing die gebruikt is in dit model worden gepresenteerd.

Dit voorbeeld bestaat uit een netwerk van vier havens: haven A, haven B, haven C en haven D. Er is één scheepstype: scheepstype 1, met capaciteit 600 FFE. Van dit type varen twee schepen rond en de bovengrens voor het aantal routes voor één schip is één. Als laatst is er één vraag: deze bedraagt 500 FFE en de oorsprongshaven is haven A en de bestemmingshaven is

haven B. In figuur B1 is een mogelijke route weergegeven die voldoet aan de constraints van het model van Ameln e.a. (2019). De groene lijnen zijn in de bovenste laag (UL) en de rode lijnen (LL) zijn in de onderste laag.



Figuur B1: Voorbeeld van een foute route.

Hierin is de stroming van de goederen als volgt:

$$f_{ALLBLLAD11} = 500$$

$$f_{BULAUHAD11} = 500$$

$$f_{CLLDLLAD11} = 500$$

$$f_{DULCULAD11} = 500$$

De volgende constraints zorgen voor de stroming van de goederen:

$$\sum_{i \in \mathcal{N}^U} \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}_k} (f_{ijodkr} + f_{P+i, P+j, odkr} - f_{jiodkr} - f_{P+j, P+i, odkr}) = 0 \quad j \in \mathcal{N}^U, (o, d) \in \mathcal{D}, j \neq o, j \neq d \quad (\text{B5})$$

$$\sum_{j \in \mathcal{N}^U} \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}_k} (f_{ojodkr} + f_{P+o, P+j, odkr}) = D_{od} \quad (o, d) \in \mathcal{D} \quad (\text{B6})$$

$$\sum_{i \in \mathcal{N}^U} \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}_k} (f_{idodkr} + f_{P+i, P+d, odkr}) = D_{od} \quad (o, d) \in \mathcal{D} \quad (\text{B7})$$

Aan alle constraints wordt voldaan, want: voor alle havens, behalve de oorsprongs- en bestemmingshaven geldt dat de instroom van de lading gelijk is aan de uitstroom van de lading. Hieruit volgt dat constraint (B5) houdt. Constraint (B6) houdt ook, omdat de uitstroom van de bestemmingshaven gelijk is aan de vraag (A, D). Evenzo houdt constraint (B7) ook, omdat de vraag (A, D) aankomt in de bestemmingshaven.

Bovenstaande fout kan ontstaan, omdat de in- en uitstroom van lading in alle punten, behalve de oorsprongshaven en bestemmingshaven, geregeld wordt door constraint (B5). De uitstroom van de oorsprongshaven wordt goed geregeld door (B6) en instroom van de bestemmingshaven wordt goed geregeld door (B7). Echter wordt de instroom van de oorsprongshaven en de uit-

stroom van de bestemmingshaven niet geregeld. Om deze fout in het model op te lossen zijn er twee mogelijke oplossingen. De eerste oplossing is het toevoegen van twee constraints. Deze constraints zijn de volgende:

$$f_{ojodkr} = f_{o,P+j,odkr} = 0 \quad j \in \mathcal{N}^U, (o, d) \in D, j \neq o, k \in \mathcal{K}, r \in \mathcal{R}_k \quad (\text{B8})$$

$$f_{idodkr} = f_{P+i,dodkr} = 0 \quad i \in \mathcal{N}^U, (o, d) \in D, i \neq d, k \in \mathcal{K}, r \in \mathcal{R}_k \quad (\text{B9})$$

Met behulp van deze constraints wordt nu ook de instroom van de oorsprongshaven en de uitstroom van de bestemmingshaven geregeld. Deze worden gelijk gesteld aan 0. Een andere oplossing is om deze stromen bij voorbaat vast te leggen en gelijk te stellen aan 0. Deze tweede oplossing wordt toegepast in het model van dit verslag.

Bijlage C Pseudocode voor het berekenen van het maximum aantal routes

In deze bijlage wordt dezelfde notatie gebruikt als in de methodologie (sectie 4). In algoritme C1 wordt de techniek voor het berekenen van de bovengrens voor het aantal routes weergegeven. Daarna wordt in figuur C1 grafisch weergegeven hoe de havens op hun hoek worden gesorteerd met behulp van een voorbeeld. Daarna wordt in algoritme B2 een verbeterde algoritme gegeven. Hierin zijn de bovengrenzen niet meer te scherp.

Algorithm *BovengrensRoutes*(G)

Data: Een graaf $G = (\mathcal{N}^U, \mathcal{A})$

Result: Een bovengrens voor het aantal routes ($|\mathcal{R}_k|$)

Initialiseer: Zoek de haven waar de meeste vracht naar toe moet en vandaan komt en maak deze de hub haven.

$routesTeller = 0$

$nietBezocht \leftarrow \mathcal{N}^U \setminus Hub$

En sorteer alle overige havens op de hoek tussen de verticale lijn vanuit de haven en de lijn van de hub naar deze haven (zie figuur C1).

while $nietBezocht$ niet leeg is **do**

$routesTeller \leftarrow routesTeller + 1$

$totaleVrachtOproutesaarHavens \leftarrow 0$

$totaleVrachtOproutesaarHub \leftarrow 0$

while $totaleVrachtOproutesaarHavens < \text{Capaciteit schepstype}$ **en** $totaleVrachtOproutesaarHub < \text{Capaciteit schepstype}$ **do**

for Haven $i \in nietBezocht$ **do**

$totaleVrachtNaarHaveni \leftarrow$ alle vraag met bestemmingshaven i

$totaleVrachtVanafHaveni \leftarrow$ alle vraag met oorsprongshaven i

if $totaleVrachtOproutesaarHavens + totaleVrachtNaarHaveni \leq \text{Capaciteit schepstype}$ **en**

$totaleVrachtOproutesaarHub + totaleVrachtVanafHaveni \leq \text{Capaciteit schepstype}$ **then**

$nietBezocht \leftarrow nietBezocht \setminus i$

$totaleVrachtOproutesaarHavens = totaleVrachtOproutesaarHavens + totaleVrachtNaarHaveni$

$totaleVrachtOproutesaarHub = totaleVrachtOproutesaarHub + totaleVrachtVanafHaveni$

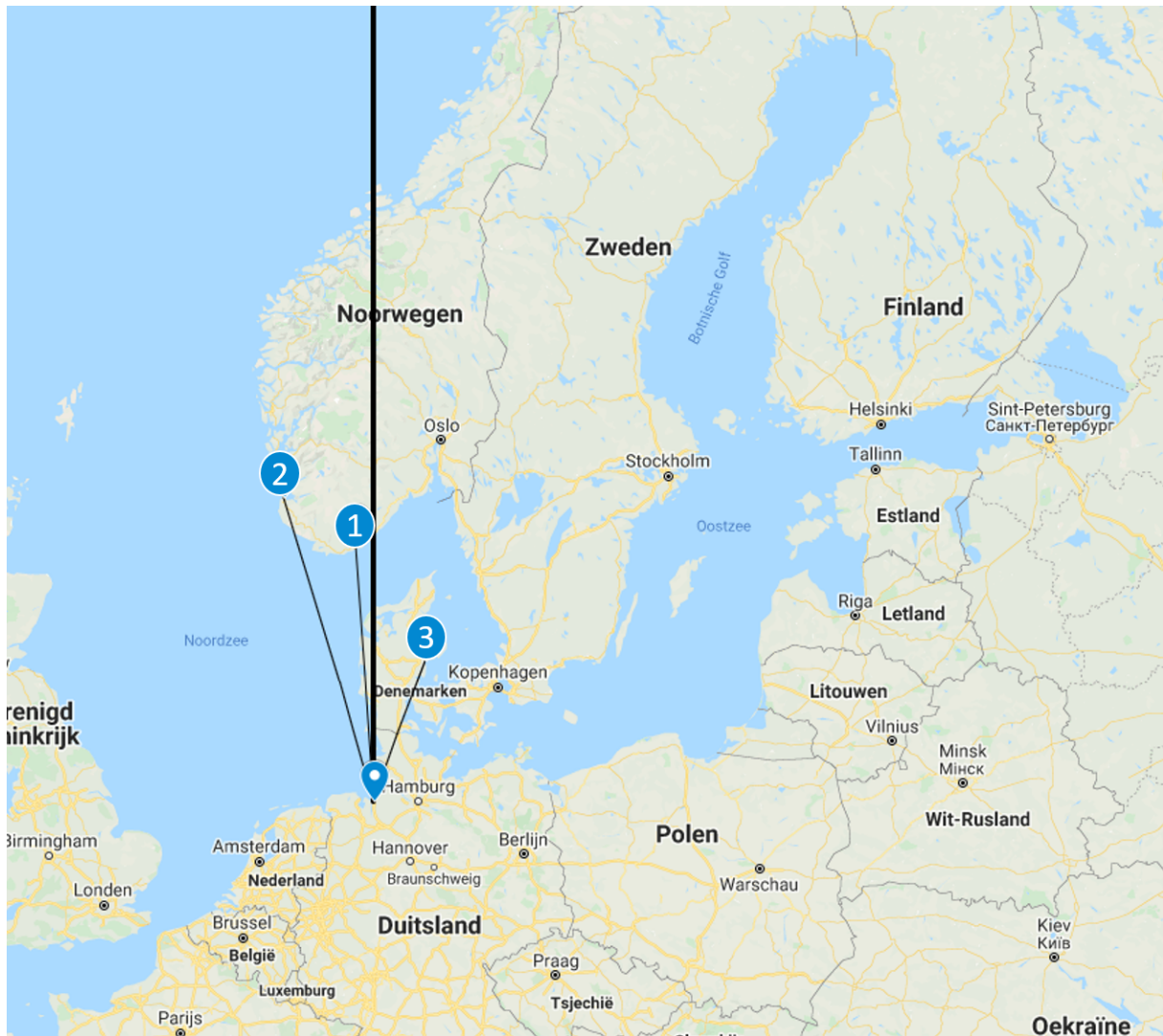
end

end

 bovengrens van het aantal routes ($|\mathcal{R}_k|$) = $\frac{routesTeller}{2}$

end

Algorithm C1: Het bepalen van de bovengrens voor het aantal routes ($|\mathcal{R}_k|$).



Figuur C1: Voorbeeld gesorteerde havens op hoeken met de hub.

Algorithm *BovengrensRoutes*(G)

Data: Een graaf $G = (\mathcal{N}^U, \mathcal{A})$

Result: Een bovengrens voor het aantal routes ($|\mathcal{R}_k|$)

Initialiseer: Zoek de haven waar de meeste vracht naar toe moet en vandaan komt en maak deze de hub haven.

$routesTeller = 0$

$nietBezocht \leftarrow \mathcal{N}^U \setminus Hub$

En sorteer alle overige havens op de hoek tussen de verticale lijn vanuit de haven en de lijn van de hub naar deze haven (zie figuur C1).

while $nietBezocht$ niet leeg is **do**

$routesTeller \leftarrow routesTeller + 1$

$totaleVrachtOproutesaarHavens \leftarrow 0$

$totaleVrachtOproutesaarHub \leftarrow 0$

while $totaleVrachtOproutesaarHavens < \text{Capaciteit schepstype}$ **en** $totaleVrachtOproutesaarHub < \text{Capaciteit schepstype}$ **do**

for Haven $i \in nietBezocht$ **do**

$totaleVrachtNaarHaveni \leftarrow$ alle vraag met bestemmingshaven i

$totaleVrachtVanafHaveni \leftarrow$ alle vraag met oorsprongshaven i

if $totaleVrachtOproutesaarHavens + totaleVrachtNaarHaveni \leq \text{Capaciteit schepstype}$ **en**

$totaleVrachtOproutesaarHub + totaleVrachtVanafHaveni \leq \text{Capaciteit schepstype}$ **then**

$nietBezocht \leftarrow nietBezocht \setminus i$

$totaleVrachtOproutesaarHavens = totaleVrachtOproutesaarHavens + totaleVrachtNaarHaveni$

$totaleVrachtOproutesaarHub = totaleVrachtOproutesaarHub + totaleVrachtVanafHaveni$

end

end

 bovengrens van het aantal routes ($|\mathcal{R}_k|$) = $routeTeller$

end

Algorithm B2: Het bepalen van de bovengrens voor het aantal routes ($|\mathcal{R}_k|$).

Bijlage D Pseudocode van de clusteringtechniek

In deze bijlage wordt dezelfde notatie gebruikt als in de methodologie (sectie 4). In algoritme D1 wordt de heuristiek voor het clusteren van de havens weergegeven. Nadat de havens zijn geclusterd, moet de vraag naar deze clusters worden aangepast. Een voorbeeld van het splitsen van de vraag staat weergegeven in figuur D1, waarna het algoritme hiervoor staat weergegeven in algoritme D2.

Algorithm *ClusterTechniek*(G, n, m)

Data: Een graaf $G = (\mathcal{N}^U, \mathcal{A})$

Minimaal aantal havens in een cluster n

Maximaal aantal havens in een cluster m

Result: Cluster van havens in een hub and spoke netwerk.

Initialiseer: $havens \leftarrow$ Alle havens in de graaf

$clusters \leftarrow$ Lege verzameling, waarin de clusters worden toegevoegd

while Aantal havens in $havens > 1$ **do**

/* Maak de clusters */

 Bereken voor iedere haven in $havens$ de gemiddelde afstand tot alle andere havens in $havens$

 Sorteer alle havens in $havens$ op de gemiddelde afstand tot alle andere havens van groot naar klein

while $havens$ niet leeg is **do**

$haven \leftarrow$ de haven in $havens$ met de grootste gemiddelde afstand

 Initialiseer: $cluster \leftarrow$ Nieuw cluster

$cluster \leftarrow$ Voeg $haven$ hieraan toe

for $i \leftarrow 0$ **to** n **do**

$havenDichtstbij \leftarrow$ De haven die het dichtstbij één van de havens in $cluster$ ligt.

$cluster \leftarrow$ Voeg $havenDichtstbij$ hieraan toe

end

for $i \leftarrow n$ **to** m **do**

$havenDichtstbij \leftarrow$ De haven die het dichtstbij één van de havens in $cluster$ ligt.

if De afstand naar $havenDichtstbij < 2 * \frac{\text{Totale afstand in het cluster}}{i}$ **then**

$cluster \leftarrow$ Voeg $havenDichtstbij$ hieraan toe

end

$clusters \leftarrow$ Voeg $cluster$ hieraan toe

end

$havens \leftarrow$ Verwijder alle havens

/* Bepaal voor elk cluster welke haven de hub is */

for $cluster \in clusters$ **do**

for $haven \in cluster$ **do**

$hubWaarde = \frac{\text{vraag naar haven-vraag vanaf haven}}{2 * \text{totale vraag}} * \frac{\text{totale afstand van en naar alle havens}}{\text{totale afstand vanaf haven}}$

end

$hub \leftarrow$ De $haven$ met de grootste $hubWaarde$

$havens \leftarrow$ Voeg hub hieraan toe

end

end

Algorithm D1: Het construeren van clusters in een graaf.



Figuur D1: Voorbeeld splitsen vraag.

In figuur D1 zijn negen havens, verdeeld over drie clusters. Iedere cluster heeft zijn eigen kleur en de hub uit het cluster is de donkere kleur. In dit cluster is er een vraag met oorsprongshaven FIRAU en bestemmingshaven DEBRV met 77 FFE per week. Om deze vraag door dit netwerk van clusters te laten stromen moet deze worden opgesplitst in drie verschillende reizen. Als eerst moet de vraag vanaf FIRAU naar RULED (de hub van het oranje cluster). Waarna de vraag vanaf RULED via het de hub cluster naar SEGOT gaat. Als laatst moet de vraag van SEGOT naar DEBRV. Door de vraag op deze manier te splitsen, kan het model voor ieder cluster goed gerund worden.

Om het model voor ieder cluster efficiënter te maken, worden de transporten die hetzelfde zijn binnen een cluster samengevoegd. Als voorbeeld is er nog een extra vraag in dit netwerk: de oorsprongshaven is FIKTK en de bestemmingshaven is DEBRV, met 162 FFE per week. Ook deze vraag zal worden gesplitst, waarbij onder andere de volgende splitsing ontstaat: vanaf RULED naar SEGOT. Omdat deze splitsing ook bestaat voor de vraag van FIRAU naar DEBRV, worden deze gecombineerd. Hierdoor moet $77 + 162 = 239$ FFE per week vanaf RULED naar SEGOT vervoerd worden.

Algorithm *CreateDemand*(G, P)

Data: Een graaf $G = (\mathcal{N}^U, \mathcal{A})$

Een lijst van alle havens in het cluster P

Result: Vraag in het cluster.

Initialiseer: $vraagGR \leftarrow$ Alle vraag in de graaf

$vraagCL \leftarrow$ Lege verzameling, waarin de vraag voor het cluster wordt toegevoegd

for $vraag \in vraagGR$ **do**

/* Zoek alle vraagparen behorende bij het cluster */

if de oorsprongshaven en de bestemmingshaven van $vraag$ in het cluster zit **then**

 | $vraagCL \leftarrow$ Voeg $vraag$ hieraan toe

else if de oorsprongshaven van $vraag$ in het cluster zit **en** de oorsprongshaven van $vraag$ niet gelijk is aan hub **then**

if de bestemmingshaven niet in een lagere cluster laag zit **then**

if de bestemmingshaven van $vraag$ in een lagere laag van een andere haven in het cluster zit **then**

 | $vraagCL \leftarrow$ Voeg $vraag$ hieraan toe, met als bestemmingshaven de haven waaronder de originele bestemmingshaven ligt

else

 | $vraagCL \leftarrow$ Voeg $vraag$ hieraan toe, met als bestemmingshaven de hub

else if de bestemmingshaven van $vraag$ in het cluster zit **en** de bestemmingshaven van $vraag$ niet gelijk is aan hub **then**

if de oorsprongshaven niet in een lagere cluster zit **then**

if de oorsprongshaven van $vraag$ in een lager clusterlaag van een haven in de cluster ligt **then**

 | $vraagCL \leftarrow$ Voeg $vraag$ hieraan toe, met als oorsprongshaven de haven waaronder de originele oorsprongshaven ligt

else

 | $vraagCL \leftarrow$ Voeg $vraag$ hieraan toe, met als oorsprongshaven de hub

else

for $haven1 \in cluster$ **do**

if de oorsprongshaven van $vraag$ in een lagere laag van $haven1$ zit **then**

if de oorsprongshaven en bestemmingshaven van $vraag$ niet allebei in de lagere laag van $haven1$ zitten **then**

for $haven2 \in cluster$ **do**

if $haven1 \neq haven2$ **then**

if de bestemmingshaven van $vraag$ in een lagere laag van $haven2$ zit **then**

 | $vraagCL \leftarrow$ Voeg $vraag$ hieraan toe, met als oorsprongshaven $haven1$ en als bestemmingshaven $haven2$

else

if $haven1 \neq hub$ **then**

 | $vraagCL \leftarrow$ Voeg $vraag$ hieraan toe, met als oorsprongshaven $haven1$ en als bestemmingshaven de hub

end

else if de bestemmingshaven van $vraag$ in een lagere laag van $haven1$ zit **then**

if de oorsprongshaven niet in een lagere laag van één van de havens uit het cluster zit **then**

 | $vraagCL \leftarrow$

 Voeg $vraag$ hieraan toe, met als oorsprongshaven hub en als bestemmingshaven de haven waaronder de originele bestemmingshaven ligt

end

end

```

/* Koppel vraagparen die bij elkaar horen */
for  $vraag1 \in vraagCL$  do
  for  $vraag2 \in vraagCL$  do
    if  $vraag1 \neq vraag2$  en de oorsprongs- en bestemmingshaven van  $vraag1$  en  $vraag2$  gelijk zijn then
      Combineer  $vraag1$  en  $vraag2$ 
    end
  end
end

```

Algorithm D2: Het construeren van de vraag in een cluster.

Bijlage E Bestanden uit de codebijlage

De volgende Java classes zitten in de codebijlage:

- `CreateData.java`: Hierin worden alle objecten van de `Object` classes geconstrueerd
- `Data.java`: Hierin worden alle objecten van de `Object` classes opgeslagen
- `Main.java`: Hoofd class om het Model of de Heuristiek uit te voeren
- `Model.java`: Implementeert het model (1a) - (1t)
- `Object classes`:
 - `Arc.java`: Creëert een lijn in het netwerk
 - `Cluster.java`: Creëert een cluster in het netwerk
 - `Demand.java`: Creëert een vraag, bestaande uit een oorsprong en bestemming, in het netwerk
 - `Port.java`: Creëert een haven in het netwerk
 - `PortArcs.java`: Combineert een haven met alle ingaande en uitgaande lijnen
 - `PortIntegers.java`: Combineert een haven met twee integers
 - `Route.java`: Creëert een route
 - `RouteVesselClass.java`: Combinaties van alle routes en scheepstypen
 - `RouteVesselClassArc.java`: Combinaties van alle routes, scheepstypen en lijnen
 - `RouteVesselClassDemandArc.java`: Combinaties van alle routes, scheepstypen, vraag en lijnen
 - `RouteVesselClassDemandPort.java`: Combinaties van alle routes, scheepstypen, vraag en havens
 - `RouteVesselClassPort.java`: Combinaties van alle routes, scheepstypen en havens
 - `UpperLayerLowerLayer.java`: Combinaties van de bovenste laag en de onderste laag representatie van een haven
 - `Vessel_Class.java`: Creëert een schip in het netwerk
 - `Vessel_Class.java`: Creëert een scheepstype in het netwerk
- `Heuristic class`:
 - `HeuristicClustering.java`: Implementatie van algoritme D1