

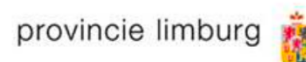
Rapport 'Ontwerp synchromodale planning en systemen –additionele functionaliteit'

30 september 2021
Hogeschool Vives

A. Louwyck, S. Haspeslagh



Met de financiële steun van



Situering en formulering doelstelling

In het SYN-ergie project werden in eerste instantie een clickable demo en een synchromodale cockpit gebouwd waarin de volledige functionaliteit voor synchromodaal boeken en de mogelijkheden en voordelen daarvan worden getoond. De clickable demo is een fictieve planningsomgeving waarin alle elementen van synchromodaal vervoer vanuit een fictieve situatie kan worden getoond en ervaren door de gebruiker. De synchromodale cockpit is een demonstratieversie van een planningstool die met reële, real-time informatie werkt. We noemen deze tool een synchromodale cockpit omdat van hieruit synchromodaal transport wordt aangestuurd.

Dit luik van het project bouwt verder op de clickable demo en op de synchromodale cockpit. Er werd onderzocht welke additionele functionaliteiten er moeten toegevoegd worden aan de synchromodale cockpit, om tot een betere en realistischere demonstratietool te komen, en die aan de noden en behoeften van logistieke spelers voldoet.

In zowel de clickable demo als in de synchromodale cockpit worden alle beslissingen over synchromodale planning nog altijd door een menselijke planner genomen. Met de doorbraak van Artificiële Intelligentie wordt meer en meer gestreefd om de systemen die ontwikkeld zijn om de digitalisering in allerlei sectoren te ondersteunen, te laten doorgroeien naar “decision supportsystemen”. Dit wil in de context van de clickable demo en de synchromodale cockpit zeggen dat de verantwoordelijkheid om beslissingen te nemen niet meer exclusief bij de menselijke planner ligt, maar dat het systeem de planner actief bijstaat bij het nemen van beslissingen. Vaak gebeurt dit door een aantal mogelijke oplossingen naar voor te schuiven waaruit de menselijke planner een definitieve oplossing kiest. De motor van een dergelijke planningstool voor beslissingsondersteuning is een slim algoritme dat in staat is een synchromodaal planningsprobleem op te lossen. De extra functionaliteit die uitgewerkt wordt is de implementatie van een proof-of-concept-algoritme, dat kan uitgebreid worden tot een volwaardige planningsmodule. In het vervolg van dit rapport beschrijven we het proces van ontwerp van het prototype algoritme tot de integratie van het algoritme in de clickable demo. De integratie in de clickable demo weerhoudt niet dat het algoritme niet aangesproken en geïntegreerd kan worden in andere tools. Tijdens het ontwerp werd specifiek aandacht geschonken aan die integratiemogelijkheden.

De initiële bedoeling was om een bevraging te doen bij de deelnemende marktpartijen uit werkpakket 3. Daarnaast is met verschillende partijen contact gelegd om data in de praktijk te valideren. Vanuit de Nederlandse kant is veelvuldig contact geweest met CTV en hebben we ook aan Belgische kant contacten gelegd met ECS2XL en zijn we ook daar op bezoek geweest om ons initiatief te presenteren. In de praktijk is gebleken dat het delen van bedrijfsspecifieke data toch erg gevoelig ligt of dat bedrijven dit intern ontwikkelen. Na verschillende hoopvolle gesprekken is het ons als projectgroep echter niet gelukt om de gewenste samenwerking te realiseren en de experimenten om het algoritme te verfijnen uit te voeren op data aangeleverd door die bedrijven. Verder hebben de ontwikkelingen in de logistieke sector omtrent het COVID-19 virus niet in ons voordeel gewerkt. Gezien het feit dat het niet gelukt is op dit vlak samen te werken met bedrijven, hebben we ervoor gekozen om op basis van de aanwezige opgedane kennis bij de clickable demo verschillende scenario's te schetsen die hieronder bij de beschrijving van de aanpak in Activiteit 1 verder toegelicht worden.

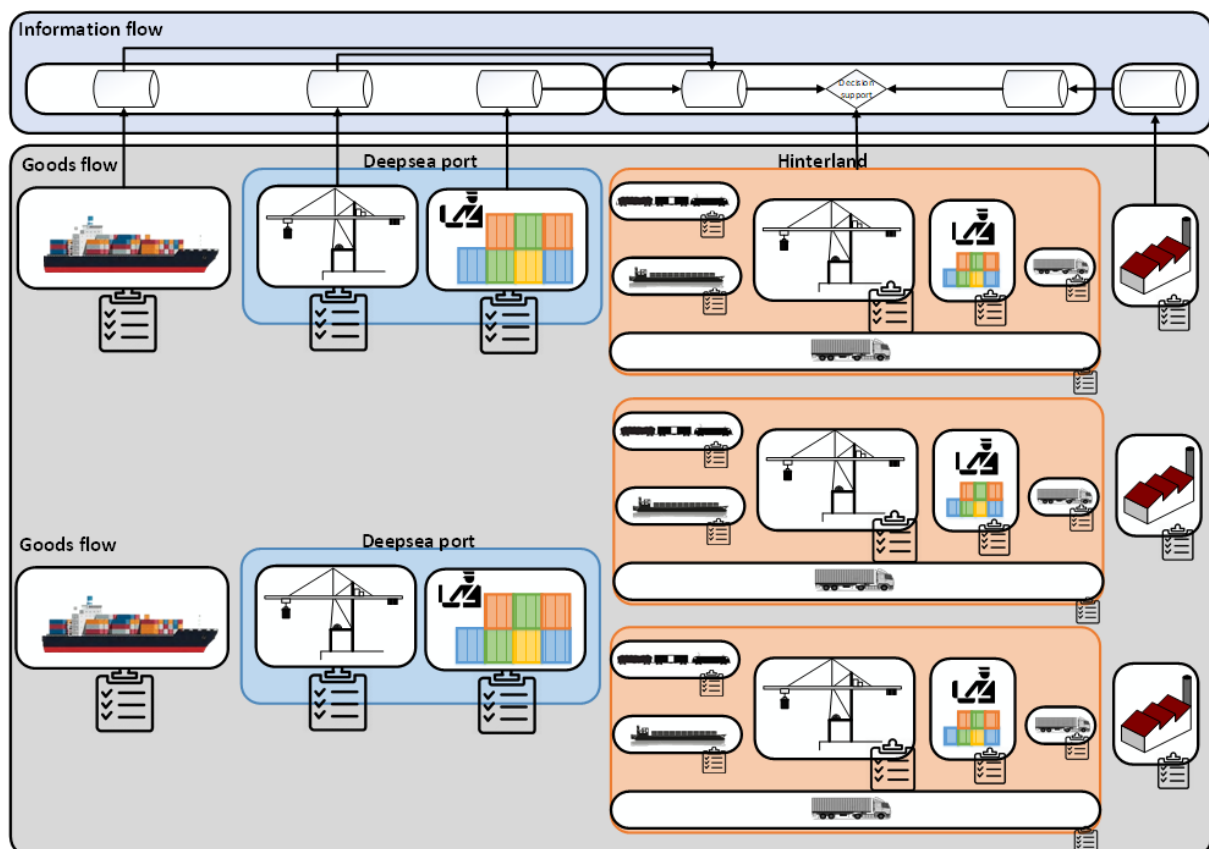
Beschrijving van aanpak en bekomen oplossing

In deze sectie beschrijven we hoe de uitbreiding van de clickable demo met een proof-of-concept optimalisatiealgoritme tot stand gekomen is.

Activiteit 1: Vooronderzoek en analyse

In de eerste fase vond een diepgaand vooronderzoek plaats waarbij de huidige situatie grondig wordt geanalyseerd. De verschillende parameters en het kenmerkende probleem van het synchro-modale-planningsprobleem werden geïnventariseerd, dit op basis van de interface van de clickable demo en gesprekken met de partners in WP 4 die een beter zicht hebben op de problematiek.

Op basis van een aantal scenario's in de clickable demo is het algoritme ontwikkeld en getest. Zie onderstaand plaatje voor een visualisatie van het boekingsproces en de benodigde informatiestromen. In deze afbeelding wordt de goederenstroom van en naar een zeehaven gevisualiseerd. De containers komen aan in de zeehaven en kunnen via verschillende modaliteiten worden getransporteerd naar het achterland (hinterland). Dit proces vindt vice versa ook plaats als er containers vanuit het achterland naar de zeehaven worden getransporteerd.



Figuur 1: Visualisatie achterland transport

Belangrijk uitgangspunt hierbij is dat de beslissing wordt genomen voor het achterland transport. In dat proces wordt bepaald welke modaliteit het beste geschikt is, maar daarvoor

heb je wel informatie nodig vanuit verschillende actoren in de keten. Op basis van deze logistieke ketens is vervolgens een gedetailleerde uitwerking gemaakt voor alle benodigde input variabelen voor het algoritme.

Er werd geanalyseerd welke data nog ontbreekt in de clickable demo om verder genoemde experimenten te kunnen faciliteren.

Concreet werd de volgende use case uitgewerkt: een aantal transportopdrachten (bv. Voor een week) moeten worden ingepland. In elke opdracht moeten enkele containers van een bepaald type van een startlocatie naar een eindlocatie worden getransporteerd. De containers moeten binnen een bepaald tijdsinterval opgehaald worden op de startlocatie en moeten op tijd geleverd worden op de eindbestemming. Anders dan bij intermodaal transport, ligt de modaliteit om een bepaald order of een bepaalde container te vervoeren op voorhand niet vast.

Er zijn een vast aantal containerplekken beschikbaar bij verschillende uitvoerende dienstverleners die volgens een vast tijdschema containers vervoeren van een startlocatie naar een eindlocatie tegen een vast tarief, verder *legs* genoemd. De transporteurs gebruiken voor elke leg een bepaalde transportmodus of modaliteit - truck, schip of trein - die wordt gekenmerkt door een gekende transporttijd en een gegeven CO2-uitstoot.

Activiteit 2: Optimalisatiemodel opstellen

Vanuit de probleembeschrijving die verzameld werd in Activiteit 1 werd een mathematisch model opgesteld (zie bijlage op het eind van dit rapport) dat een snelle evaluatie toelaat en daardoor dus bruikbaar is in de context van optimalisatiealgoritmen. Belangrijk was om tot een generiek model te komen dat een diversiteit van de problemenstellingen kan capteren. Het model werd in verdere activiteiten getest op de probleemstelling uit Activiteit 1 en bijhorende data.

Concreet werd een doelfunctie opgesteld die in staat is de kwaliteit van een oplossing te meten in termen van operationele kost en (klant)tevredenheid. Er wordt bij deze keuze van functionaliteiten een onderscheid gemaakt harde en zachte beperkingen. Harde beperkingen mogen niet geschonden worden, b.v. elke aanvraag voor transport moet ingeroosterd en uitgevoerd worden. Zachte beperkingen zijn beperkingen die eventueel afgezwakt mogen worden, b.v. van bepaalde klanten mag de uiterlijke levertermijn overschreden worden, weliswaar ten koste van de tevredenheid van die klant of de kwaliteit van de oplossing.

Activiteit 3: Ontwikkelen optimalisatiealgoritme

Op basis van het bekomen wiskundig optimalisatiemodel werd in de literatuur gezocht naar bestaande algoritmen die in staat zijn een oplossing te berekenen voor het wiskundig probleem. Er werden twee pistes bewandeld.

Voor een beperkt planningsprobleem, d.i. voor een beperkt aantal transportorders en containers, en een beperkt aantal *beschikbare legs*¹, werd een exact algoritme uitgewerkt

¹ Hiermee bedoelen we dat er een beperkte keuze is in beschikbare modaliteiten en capaciteit bij de dienstverleners, wat resulteert in een klein aantal mogelijke legs in het wiskundig model.

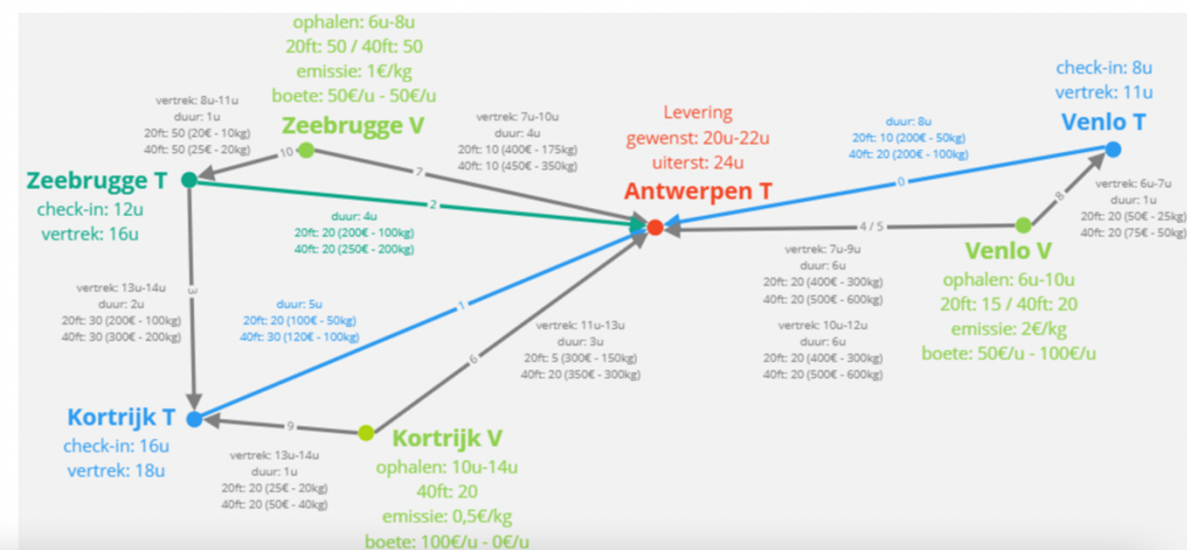
gebaseerd op lineaire programmering. Om het lineaire programma op te lossen werd gebruikt gemaakt van een *open source solver*, PULP². Het voordeel van deze methodiek is dat steeds de optimale oplossing gevonden wordt (voor zover er een oplossing is). Als het planningsprobleem te groot wordt, is deze methode echter qua rekentijd ontoereikend. Afhankelijk van de probleemgrootte kan het ettelijke uren of dagen duren voordat het exacte algoritme een oplossing vindt.

Om ook voor grotere planningsproblemen een goede oplossing te vinden, hebben we een bestaand algoritme, gebaseerd op Adaptive Large Neighbourhood (ALNS), uit de literatuur³ aangepast aan het wiskundig model uit Activiteit 2. Dit algoritme is in staat om snel een goede oplossing te vinden, een gelijkwaardige of in sommige gevallen zelfs betere oplossing dan die gevonden door een menselijke planner.

Om de werking van de algoritmen te illustreren, werd een interactieve tutorial⁴, in de vorm van een Jupyter notebook, opgesteld (zie figuren 2 en 3 hieronder, meer informatie is te vinden in de notebooks). Het ALNS-algoritme is als een proof-of-concept versie uitgewerkt. Het algoritme is aanpasbaar geïmplementeerd om integratie in andere software dan de clickable demo mogelijk te maken. Hiervoor is dan wel een verder finetunen van het algoritme aan de functionaliteiten van de nieuwe software en aan de wensen van de nieuwe gebruikers noodzakelijk.

Voorbeeld 2

In het tweede voorbeeld gaan we een iets complexer planningsprobleem oplossen. Transportorders en capaciteiten worden op de figuur weergegeven.



Figuur 2: voorbeeldprobleem

² <https://coin-or.github.io/pulp/>

³ Pickup and Delivery Problem with Transshipment for Inland Waterway Transport, Zhang et al, Computational Logistics, LNCS, volume 12433, 2020, doi 10.1007/978-3-030-59747-4_2

⁴ <https://www.project-synergie.eu>

Linear Programming

De optimale oplossing vinden we door *Linear Programming* toe te passen.

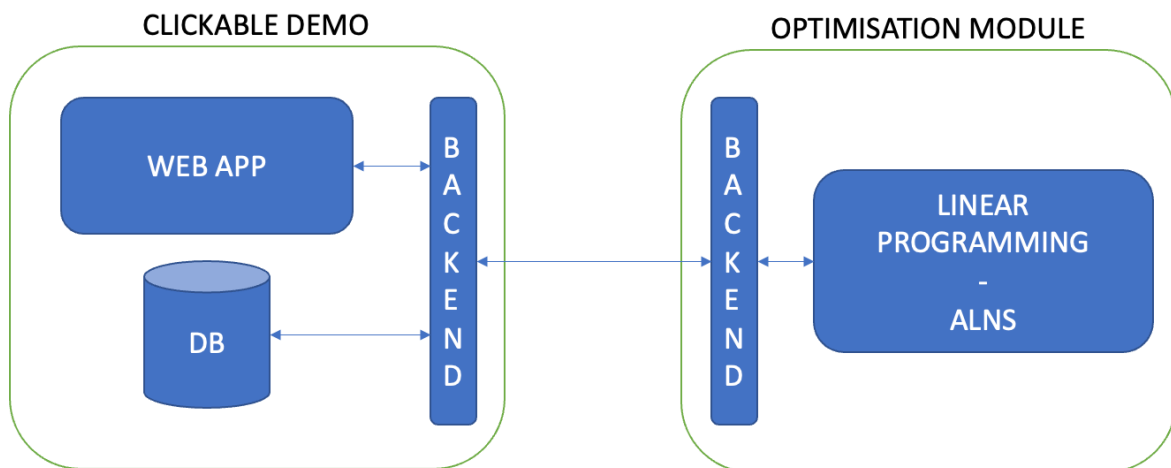
```
In [40]: lp = LinearProgramming(planning)
         lp.solve()
```

```
Elapsed time: 0.26 sec
Solution status: Optimal
Minimal cost: 97050.0
```

Figuur 3: kost van een exacte oplossing voor het probleem in figuur 2

Activiteit 4: Integratie optimalisatiealgoritme en demo-applicatie

Een van de vereisten bij het implementeren en het uitwerken van het optimalisatiealgoritme was integreerbaarheid in andere planningssystemen. Hiervoor werd volgens de state-of-the-art een REST-API ontwikkeld rond het planningsalgoritme heen (zie figuur hieronder). Zo ontstaat een module voor decision support bruikbaar in andere softwaresystemen. Er werd gekozen voor een implementatie in Python, een backend in Flask en een integratie in de demonstrator. De REST-API is gedocumenteerd volgens de OPENAPI-specificatie.



Figuur 4: softwarearchitectuur

In de bovenstaande figuur kan de clickable demo vervangen worden door een andere planningsapplicatie. Zoals eerder aangehaald is voor elke nieuwe integratie een finetuning van het ALNS-algoritme nodig opdat de optimalisatiemodule planningen volgens de voorkeuren van de nieuwe gebruiker(s) zou kunnen genereren.