REINFORCEMENT LEARNING MODEL

De Haagse Hogeschool Den Haag
24 januari 2023

Studenten:

Bonno van Nieuwenhout	19122381
Hidde Franke	19086504
Joeri Meijers	20123558
Manon Rongen	19075235
Michael Broer	20105533
Mohamed Amajoud	20198752

Begeleid door:

Jeroen Vuurens
Tony Andrioli
Karin de Smidt - Destombes
Edwin van Noort

Samenvatting

Dit onderzoek richt zich op het zo optimaal mogelijk indelen van containerterminals door middel van geautomatiseerde methoden. Het bedrijf Cofano, dat zich bezighoudt met het optimaliseren van terminalprocessen, streeft ernaar om de tijd dat schepen aan de kade liggen te minimaliseren en daarmee de kosten zo laag mogelijk te houden. Een van deze processen wordt gedaan door middel van stackers die containers kunnen verplaatsen. Het doel is om de stacker containers te laten ophalen voor verder vervoer in een minimaal aantal stappen. Er is een Reinforcement Learning model ontwikkeld om dit complexe probleem op te lossen en is schaalbaar zodat het makkelijk uit te breiden is naar grotere yards, opstapeling van containers of situaties met meer containers en schepen. Gefocust op deze factoren heeft het model een eindoplossing gecreëerd waarbij de stacker bij alle gewenste containers kan komen, zonder extra stappen. Na evaluatie blijkt dit model, met kleine aanpassingen, schaalbaar te zijn door een optimale oplossing te genereren voor yards met verschillende groottes.

Inhoudsopgave

1. Introductie

Introductie

In dit hoofdstuk zal een korte introductie gegeven worden over het containerbedrijf Cofano en het probleem wat behandeld wordt in dit onderzoek. Vervolgens wordt het onderzoeksdoel omschreven en ten slotte worden er andere werken, die gerelateerd zijn aan dit onderzoek, besproken in het literatuuronderzoek. Na de introductie kan er gelezen worden over de opzet van dit onderzoek, hoe de literatuur is gekoppeld aan dit probleem, de dataverzameling en de methodologie. Daarna volgt het hoofdstuk met de onderzoeksresultaten, waarin de belangrijkste resultaten getoond zullen worden. Ten slotte kunt u de conclusie en discussie lezen, waarin de resultaten teruggekoppeld worden aan de doelstelling van dit onderzoek en mogelijke verbeteringen en/of punten waar tegenaan is gelopen gedurende dit onderzoek worden benoemd.

Probleemstelling

Cofano is een bedrijf dat zich bezighoudt met het optimaliseren van terminalprocessen. Deze processen houden het volgende in. Er komen binnenvaartschepen aan, die containers af moeten lossen op de kade van een terminal. De containers staan daar vervolgens totdat een zeevaartschip de containers komt halen. In de volgorde waarop schepen de terminal in- en uitvaren zit een onzekerheid, met deze onzekerheid moet rekening gehouden worden. Cofano wil de tijd dat schepen aan de kade liggen minimaliseren om te kosten zo laag mogelijk te houden. Dit betekent dat de af- en inlaadprocessen geoptimaliseerd moeten worden.

Op de kade in een haven rijden stackers rond die containers kunnen verplaatsen. Hoe minder stappen een stacker nodig heeft om een container te bereiken, hoe sneller het proces verloopt. Het doel van Cofano is om deze verschillende processen te optimaliseren om zo de kosten zo laag mogelijk te houden.

Er wordt in de terminals gewerkt met reach stackers. Deze stackers kunnen alleen containers pakken vanaf de lange zijde en vanaf verschillende hoogtes. Als eerst moeten bovenste containers verplaatst worden, als een onderste container gepakt moet worden. Als een container tussen andere containers ligt, kan de stacker hier niet bij. Het is mogelijk maximaal vijf containers op elkaar te stapelen.

Visualisatie?

Het doel van dit onderzoek

In dit onderzoek is een methodologie ontwikkeld voor het automatiseren van het vinden van een optimale indeling van de yard in een terminal, zo dat een stacker containers ophaalt voor verder vervoer in een minimaal aantal stappen. Dit is een complex probleem. Er zijn diverse soorten containers, en die zijn van diverse grootte en gewicht. Daarnaast zijn er vele containerterminals in de wereld, en die verschillen qua grootte, aantal containers en soorten containers. Dit maakt de zoektocht naar een algemene oplossing lastig.

Omdat het een dynamisch probleem is, is het van belang dat het model schaalbaar is. Dat wil zeggen dat het makkelijk uit te breiden is naar bijvoorbeeld een grotere yard of een situatie

met meer containers en schepen. Wel zijn er een aantal aannames gedaan waar het model mee werkt. Het model houdt geen rekening met verschillende soorten containers en gaat uit van één yard waar containers geplaatst kunnen worden, zonder rekening te houden met de verdere indeling van een terminal.

Literatuuronderzoek

Het Container Stacking Problem (CSP) is een bekend probleem, waar al meerdere onderzoeken naar zijn gedaan. In deze onderzoeken wordt er gekeken naar het optimaliseren van bepaalde terminalprocessen. Het probleem is op te delen in losse problemen, zoals bijvoorbeeld het uitladen en plaatsen van containers, het inzetten van stackers en het vervoeren van containers van de terminal naar de zeevaart schepen. Er zijn verschillende heuristieken gebruikt om zulke problemen op te lossen.

Voor het bepalen van een optimale opslag strategie is door Euchi et al. (2016) een Ant Colony Optimazation (ACO) algoritme gebruikt. In dit onderzoek is het doel om de afstand vanaf de container in de terminal naar de aanlegplaats van het schip, dat deze container komt ophalen, te minimaliseren. Het ACO geeft betere en efficiëntere oplossingen dan een eerder gebruikt Hybrid Genetic Simulated Annealing Algorithm (HGSAA) (Moussi et al., 2012). Dit probleem verschilt van het probleem omschreven in dit paper, aangezien het hier niet gaat om hoe goed een stacker bij een container kan.

Kefi et al. (2007) vergelijkt een Uninformed en Informed Search Algorithm voor het toewijzen van een slot aan een container. Het model minimaliseert het aantal bewegingen en verplaatsingen die nodig zijn om een container te pakken. Het Informed Search Algorithm geeft de meest optimale oplossing voor het probleem. Salido et al. (2009) heeft Artificial Intelligence (AI) toegepast op het CSP. Net als in het onderzoek wat in dit paper beschreven wordt, richt dit model zich op een optimale plaatsing van containers voor het ophaalproces. Het doel is het minimaliseren van het aantal verplaatsingen dat nodig is om een container te pakken. Ook in dit paper is het doel dat de stacker elke container zo makkelijk mogelijk kan bereiken.

In een ander onderzoek (Ries et al., 2014) wordt een Fuzzy Logic Model gebruikt om binnenkomende containers toe te wijzen aan een plek in de terminal. In dit onderzoek ligt de focus op het bouwen van een flexibel model, wat rekening houdt met onzekerheden van aankomsttijden van containers bij de terminals en wat bruikbaar is voor terminals met verschillende indelingen en infrastructuren. Het is voor dit onderzoek belangrijk te werken aan een flexibel model voor het vinden van een optimale indeling, omdat in dit onderzoek de onzekerheid van aankomst van containers een rol speelt.

Daarnaast is op dit gebied ook al gewerkt met Reinforcement Learning (RL) om optimale oplossingen te vinden voor terminalprocessen. In het onderzoek van Jiang et al. (2021) ligt de focus op het optimaal stapelen van containers op basis van prioriteit van een container. Het doel is om het aantal relocaties dat moet plaatsvinden voordat alle containers op een bepaalde volgorde gepakt kunnen worden, te minimaliseren. Dit probleem wordt aangepakt met RL en de resultaten zijn net zo goed als de meest optimale oplossingen die met andere methodes gevonden zijn. Dit onderzoek heeft veel overeenkomsten met het onderzoek beschreven in dit paper. In het onderzoek van Hu et al. (2021) wordt met twee verschillende

methodes, Integer Programming (IP) en RL, het vervoeren van containers naar het schip geoptimaliseerd. Het doel is het vinden van een optimale operatievolgorde en optimale routes voor de stackers, die rijden tussen de containers en het schip. Het RL-model scoort daarbij het beste. De resultaten van Jiang et al. (2021) en Hu et al. (2021) laten zien dat RL een geschikte methode is om soortgelijke problemen aan te pakken.

Hu et al. (2023) bekijkt het optimaliseren van het planningsproces in containerterminals. Hier wordt een systeem environment gebouwd en door middel van RL gezocht naar een optimale planning van operaties in de terminal. Hier is ondervonden dat RL efficiënt werkt en flexibeler is dan andere heuristieken. Dit is erg belangrijk voor het CSP, aangezien daar veel onzekerheden bij komen kijken en het een dynamisch probleem is.

In het onderzoek van Krishna & Sudhir (2020) zien we meerdere experimenten waarbij RL modellen worden vergeleken. Dit onderzoek kan toegepast worden voor het zo efficiënt mogelijk indelen van de kade. Uit onderzoek naar de modellen bleek dat het A2C-model voor een deel gebaseerd is op het PPO-model. Echter zijn de resultaten van het PPO-model voor de experimenten, die besproken zijn in dit onderzoek, beter.

In een video (Renotte, 2021) over Reinforcement Learning wordt aandacht besteed aan het zelf bouwen van een environment en RL-model. Stap voor stap wordt een douche environment opgebouwd en een model die de temperatuur gedurende een uur optimaliseert. Met behulp van deze video kan er een vergelijkbaar RL-model gebouwd worden die, bijvoorbeeld voor dit onderzoek, containers op een zo optimaal mogelijke manier plaatst.

2. Onderzoeksopzet

Toepassing van het theoretisch kader

Voordat er een model ontwikkeld wordt, is er eerst onderzoek gedaan naar de methoden die al zijn ontwikkeld om containerterminals efficiënter in te richten. Hierdoor is een goed beeld geschetst van de huidige stand van zaken en kan er gericht worden gekeken naar een verbetering van deze methoden. Uit het literatuuronderzoek bleek dat Reinforcement Learning een geschikte techniek is om dit probleem te tackelen.
Deze onderzoeksopzet richt zich dan ook op het gebruik van Reinforcement Learning. Er is voor deze benadering gekozen omdat Reinforcement Learning zich leent voor problemen waarbij een agent leert van zijn acties en de gevolgen daarvan, in een dynamische omgeving. Ook is het een stabiele en robuuste methode voor het oplossen van problemen met veel discrete acties.

Uit het literatuuronderzoek blijkt dat er een aantal modellen geschikt zijn om toe te passen voor het zo efficiënt mogelijk indelen van de kade. Uit onderzoek naar de modellen bleek dat het PPO-model voor een deel gebaseerd is op het A2C-model. Het verschil is dat het A2C-model agressiever zoekt naar een verbetering. Dit zien we dan ook terug in de value loss van het A2C-model vergeleken met het PPO-model, de value loss is een grafiek die laat zien hoe het model leert over de tijd. Bij het A2C-model is dit een heel erg fluctuerende lijn, bij het PPO-model loopt deze lijn een stuk vlakker. Uiteindelijk zijn beide modellen uitgeprobeerd en is er voor dit onderzoek gekozen voor het PPO-model omdat hier de beste resultaten uit voort kwamen.

Dataverzameling

Om dit Reinforcement Learning model te trainen en te evalueren, is er gebruik gemaakt van gesimuleerde data. Dit is gedaan, omdat de beschikbare data van Cofano niet bruikbaar was voor dit onderzoek. De gesimuleerde data bestaat uit een lijst met containers die elk een nummer krijgen, dit nummer staat voor het zeevaartschip dat de container weer komt ophalen vanaf de kade. De lijst wordt dan gevuld met voor elk schip evenveel containers. Aangezien de volgorde van binnenkomst van containers kan verschillen, wordt de lijst met containers op een willekeurige volgorde ingedeeld.

De containers worden dan een voor een vanuit de lijst in een matrix, wat de container yard weergeeft, geplaatst en weer uit de lijst met containers verwijderd. Voor het identificeren van de omgeving is er in het begin voor een drie bij drie matrix gekozen. Vanuit deze matrix kunnen makkelijk uitbreidingen plaatsvinden, zowel in de breedte als in de hoogte. Voor een drie bij drie matrix kan de lijst met containers verschillende vormen aannemen. Er moeten negen containers geplaatst worden omdat er negen plekken zijn en de matrix van leeg naar vol wordt gevuld. Maar over hoeveel binnenvaartschepen deze zijn verdeeld is een variabele die veranderd kan worden. Als er bijvoorbeeld wordt gekozen voor een situatie met drie binnenvaartschepen, dan worden er drie containers voor schip 1 toegevoegd aan de lijst en zo geldt dat ook voor schip 2 en schip 3. De lijst met containers is zo opgebouwd dat er per rij in de matrix genoeg containers zijn om een gehele rij van hetzelfde nummer te kunnen voorzien.

Methodologie

Opbouw van het model

Voor het toepassen van het PPO-model zijn er een aantal belangrijke factoren die benoemd moeten worden. Zo is het onder andere van belang dat het duidelijk is voor welke action space is gekozen. De action space die aan het model wordt meegegeven is een discrete actie die een x en y coördinaat meegeeft. Deze x en y coördinaat staan voor de plek waar de container wordt geplaatst in de matrix. Aan het model wordt ook een observation space meegegeven. Deze observation space bestaat uit de environment waarin wordt gewerkt, die bestaat weer uit drie verschillende onderdelen. Voor een drie bij drie matrix gaat het om een box van drie bij drie, het nummer van de container die geplaatst gaat worden in die zet en de lijst met containers die nog geplaatst moeten gaan worden zodat het model weet welke containers nog gaan komen. Om ervoor te kunnen zorgen dat het model leert van zijn acties, moet het model fouten of slechte zetten gaan herkennen. Door een reward functie op te stellen kan er een score per zet worden berekend. Wanneer het model bijvoorbeeld een container plaatst op een plek waar deze niet optimaal staat kan er een negatieve score worden gegeven. Hierdoor zal het model leren om in het vervolg die container niet meer op die plek te plaatsen. Met de som van alle zetten kan er een eindscore worden berekend. Het model zal gaan proberen om elke nieuwe iteratie zijn eindscore te gaan verbeteren. Op die manier traint het model om een betere opstelling van containers te creëren. Om het model te trainen wordt er aangegeven hoeveel iteraties deze moet doorlopen.

De gevonden factoren die van invloed zijn op de efficiëntie van containerterminals zijn in dit geval de plaatsing van de containers en de daarbij gepaarde tijd die de stackers nodig hebben om bij een container te komen. Wanneer de stacker eerst container A moet verplaatsen om bij container B te kunnen is dat niet optimaal, deze handeling kost meer tijd dan wanneer de stacker meteen container B kan pakken. Om de stacker in eerste instantie bij alle gerichte containers te laten komen, zijn er een twee factoren die hierin een rol spelen. Aangezien de stackers de containers alleen vanaf de lange zijde kunnen verplaatsen, is het niet handig om containers in te boxen. Inboxen vindt plaats als er aan beide lange zijdes van een container een container staat van een ander nummer. De stacker kan niet meteen bij de geïnitieerde container, maar heeft daar een extra stap voor nodig. Daarnaast is het belangrijk dat er geen gaten ontstaan tussen containers bij het plaatsen van de containers. Met de stackers is het namelijk zo dat deze geen container kan pakken of plaatsen wanneer er aan beide lage zijdes van de geïnitieerde container nog containers staan. Het is dus handig om de containers bij plaatsing dicht bij de containers te plaatsen die voor hetzelfde schip bestemd zijn. Nu de knelpunten zijn geïdentificeerd kan daar rekening mee worden gehouden bij het creëren van het model. Dat wordt gedaan door in de reward functie rewards en penalty’s toe te kennen waardoor het model gaandeweg leert wat het meest optimale is.

Evaluatie

Wanneer het model de training heeft afgerond is het noodzakelijk dat er wordt geëvalueerd hoe het model de kade heeft ingedeeld en of dat op een zo efficiënt mogelijke manier is gedaan. Dit wordt handmatig gedaan. Het model print zijn uitkomsten en door te kijken hoe het model de kade in heeft gedeeld, kan er worden gezien hoe het model dat heeft gedaan en hoe het mogelijk beter kan. Naast het evalueren van de indeling van de kade zal ook het model zelf geëvalueerd worden. Dit wordt gedaan aan de hand van de grafieken die laten

zien hoe het model leert over de tijd. Door deze grafieken te bestuderen kan er gezien worden of het model optimaal presteert.

Van 90% naar 80% naar 70%…. Van de ingevulde grid.

Complexiteit van het model

Het doel van dit onderzoek is het ontwikkelen van een model dat makkelijk schaalbaar is. Om te kunnen laten zien dat dit met dit model kan, worden er verschillende situaties uitgewerkt. Om te beginnen wordt er gekeken naar een kade in een vorm van een matrix van drie bij drie. Hier kunnen negen containers geplaatst worden. Wanneer dit werkt kan er worden gekeken naar een uitbreiding in de vorm van een matrix van vier bij vier. Er komt dan wel een extra kolom waarbij er containers ingeboxt kunnen worden. Daar moet rekening mee gehouden worden in de reward functie. De reward functie zal dus een kleine aanpassing moeten krijgen. Nadat een vier bij vier matrix optimaal werkt kan er worden gekeken naar een vijf bij vijf matrix. Hier komt er nog een extra kolom bij en zal de reward functie dus weer een kleine aanpassing krijgen.

Nadat het blijkt dat er redelijk makkelijk kan worden uitgebreid in de lengte en breedte, is het idee om dat ook te gaan onderzoeken voor de hoogte. Bij een containerterminal worden de containers immers ook de hoogte in gestapeld. Hier komen wel een aantal andere componenten bij kijken. Zo moet er in de observation space een aantal dingen worden veranderd. Tot hiervoor werd er gewerkt in een twee dimensionale omgeving. De observation space wordt nu een drie dimensionale omgeving. Daarvoor moest ook de reward functie weer worden bijgeschaafd, er moest ook rekening worden gehouden met het feit dat het model ook containers op elkaar mocht gaan stapelen.

3. Onderzoeksresultaten

Een RL-algoritme leert door rewards en penalty's te krijgen bij elke actie die het model onderneemt. Door deze scores leert het model om betere acties uit te voeren. Nadat het model alle iteraties heeft doorgelopen is het model op een bepaald niveau geleert in het oplossen van het probleemdomein. Vervolgens moet er worden gekeken naar wat het model precies geleerd heeft en waar het toe in staat is. De resultaten worden verwerkt en het model wordt daarop verbeterd. Hieronder zal worden beschreven wat de resultaten zijn van het uiteindelijke model en wat deze resultaten betekenen.

Verzamelde data

Om te kijken naar de resultaten van het model kan er naar 2 aspecten worden gekeken. Het eerste aspect is de voorspellingen die het model geeft en de visualisaties hiervan. Met deze visualisatie kan worden gekeken naar het resultaat van het model. In dit onderzoek houdt dat in hoe de containers op de yard zijn geplaatst.
Het tweede aspect is de ontwikkeling van het model. Tijdens het leerproces wordt data bijgehouden van het model, waarbij deze data geplot kan worden in een grafiek. Uit deze gegevens kan vervolgens worden achterhaald of het model naar verwachting geleerd heeft.
Deze twee aspecten zullen daarom beide beschreven worden. Zoals eerder is beschreven zijn er uitwerkingen parallel aan elkaar gemaakt. Het beste resultaat hiervan zal worden beschreven.

Reward functie

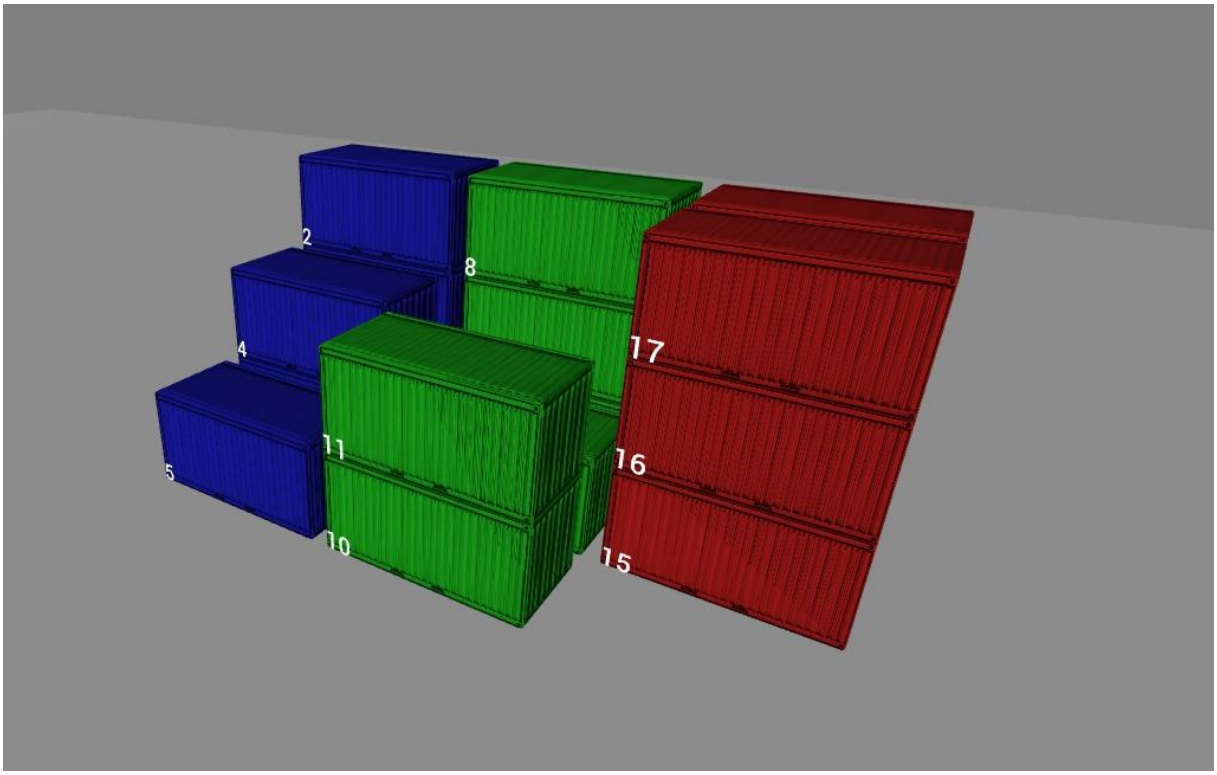
De reward functie is een fundamenteel onderdeel van een reinforcement learning algoritme. Door deze functie kan de agent begrijpen wat er goed of slecht is aan de actie die het heeft uitgevoerd. Bij het leren van wat goed of slecht is aan de actie zal de agent de acties aanpassen om een grotere beloning te verwerkelijken. De reward moet duidelijk zijn en het doel van de agent weergeven. Het moet een richtlijn zijn voor de agent om het probleemdomein zo snel en effectief mogelijk op te lossen. Als de reward functie niet goed in elkaar zit, zal de agent nooit tot een snel en effectieve oplossing komen.
Bij het maken van het reward functie is er gekeken naar de volgende punten

- De container mag niet boven de maximale hoogte komen
- De container mag niet op een container worden gestappeld die voor een ander schip bedoeld is
- De container mag niet worden ingeboxed[12]

Bij het kijken naar de maximale hoogte wordt alleen een penalty gegeven als de agent een container op een container wil plaatsen die al op de maximale hoogte staat. Hierbij wordt de bovenste container overschreven als dit het geval is. In het geval dat de maximale hoogte nog niet is bereikt wordt er gekeken naar voor welk schip de container eronder bestemd is. Als het voor hetzelfde schip bestemd is, wordt er een beloning toegekend. Als het voor een ander schip bestemd is zal er een penalty worden gegeven. Daarnaast wordt er een beloning gegeven als een container op een lege plek wordt neergezet. Vervolgens is er een check voor het inboxen waarbij alleen een penalty gegeven wordt als er volgens de check wordt geconstateerd dat er een container is ingeboxed. Het neerzetten op een lege plek en checken voor inboxen wordt berekend als 1 beloning of penalty. De score wordt bij elkaar opgeteld en die som krijgt de agent terug.

Visualisatie resultaten

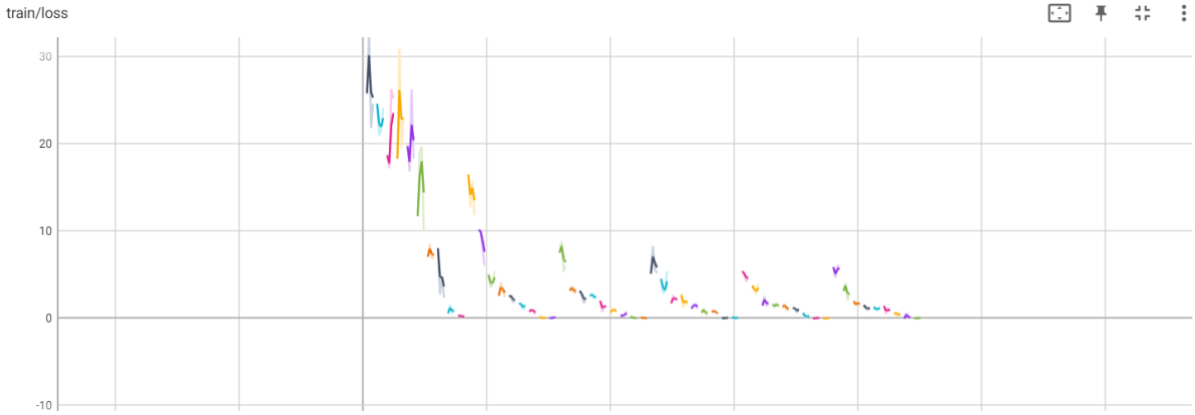
Nadat de agent geleerd heeft, geeft het model zijn uitkomst van het neerzetten van de containers. Deze uiteindelijke uitkomst is het beste wat het model kan neerzetten voor het probleemdomein. De lijst van containers die het model krijgt om neer te zetten is random geshuffled en hieronder wordt visueel getoond hoe die containers uiteindelijk geplaatst zijn.



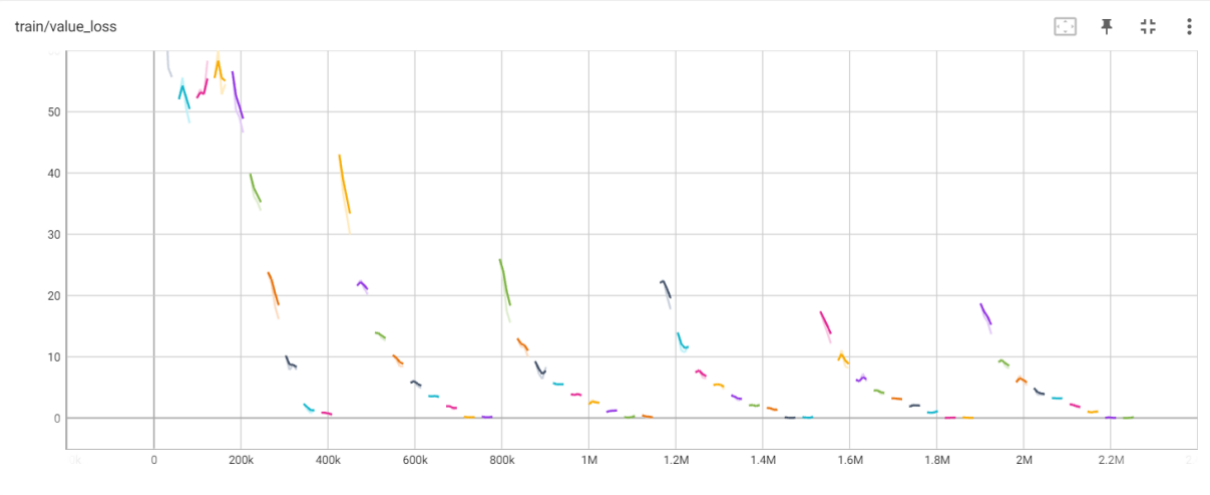
De kleurcodes die in het figuur te zien zijn, zijn indicaties voor welk schip de container bedoeld is. In het figuur is te zien dat de containers per kleurcode achter elkaar via de lange zijde staan.

Grafieken leerproces modellen

Om te begrijpen of een model geleerd heeft of progressie maakt tijdens het leren kan gebruik worden gemaakt van het plotten van data dat het model heeft tijdens het leerproces van het model. Hieronder zijn een paar plots te zien over het model.



De loss functie is een weergave van het verlies van de acties die de agent maakt dat wordt beoordeeld door een critic. Als de loss minimaal is, betekend dat de agent de beste stappen heeft gemaakt om tot het eindresultaat te komen.



De value loss functie laat zien hoe goed het model is in het voorspellen van de waarde van elke state. Deze grafiek zou moeten afnemen naarmate de beloningen stabiliseert.

Analyse

Om de prestaties van het model te analyseren hebben we gekeken naar de visualisatie van wat het model predict en naar de leergrafiek. Uit deze data blijkt dat het model, voor een yard van drie bij drie met een maximale hoogte van drie containers, een uitstekende voorspelling maakt. Het model zet de containers voor elk schip bij hetzelfde blok en er is geen verlies met te hoog stapelen. Er is dus geen sprake van inboxen of een container op een andere container zetten die voor een ander schip bedoeld is. Door te kijken naar de data van het model kan dit beaamd worden. De leergrafiek laat zien dat de uiteindelijke loss minimaal is. Dit houdt in dat er geen penalties wordt gegeven aan de agent door de reward functie. Het model geeft dus volgens de grenzen van de reward functie een perfecte oplossing voor het probleemdomein.

4. Conclusie & discussie

Conclusie

Hier komt de conclusie

- Hoe optimaal
- Hoe schaalbaar

Discussie

Hier komt de discussie

- Discussiepunten:
  - …
- Aanbevelingen:
  - Aantal zetten in reward
  - Niet handmatig evalueren
  - …

Literatuurlijst

- Euchi, J., Moussi. R., Ndiaye, F., Yassine, A (2016). Ant Colony Optimization for Solving the Container Stacking Problem. *International Journal of Applied Logistics, 6*(2), 81-101. doi: 10.4018/IJAL.2016070104

- Hu, H., Yang, X., Xiao, S., & Wang, F. (2023). Anti-conflict AGV Path Planning in Automated Container Terminals Based on Multi-agent Reinforcement Learning. *International Journal of Production Research, 61*(1), 65-80. doi: 10.1080/00207543.2021.1998695

- Hu, X., Yang, Z., Zeng, Q. (2012) A Method Integrating Simulation and Reinforcement Learning for Operation Scheduling in Container Terminals. *Transport, 26*(4), 383-393. doi: 10.3846/16484142.2011.638022

- Jiang, T., Zeng, B., Wang, Y., & Yan, W. (2021) A New Heuristic Reinforcement Learning for Container Relocation Problem. *Journal of Physics: Conference Series, 1873*(1). 012050. doi: 10.1088/1742-6596/1873/1/012050

- Kefi, M., Korbaa, O., Ghedira, K., & Yim, P. (2007). Heuristic-based model for container stacking problem. In *19th International Conference on Production Research-ICPR* (Vol. 7).

- Krishna, V., Sudhir, Y. (2020). *Comparison of Reinforcement Learning Algorithms* [Powerpoint-slides]. Departure of Computere Science and Engeneering, University at Buffalo. Geraadpleegd op 28 november 2022, van https://cse.buffalo.edu/~avereshc/rl_fall20/

- Moussi. R., Ndiaye, F., Yassine, A. (2012). Hybrid Genetic Simulated Annealing Algorithm (HGSAA) to Solve Storage Container Problem in Port. *Intelligent Information and Data base Systems*, 301-310. doi: 10.1007/978-3-642-28490-8_32

- Renotte, N. (2021, 6 juni). *Reinforcement Learning in 3 hours | Full Course Using Python* [Video]. YouTube. Geraadpleegd op 21 november 2022, van https://www.youtube.com/watch?v=Mut_u40Sqz4&t

- Ries, J., González-Ramírez, R. G., Miranda, P.(2014). A Fuzzy Logic Model fort he Container Stacking Problem at Container Terminals. *International Conference on Computational Logistics,* 93-111. doi: 10.1007/978-3-319-11421-7_7

- Salido, M. A., Sapena, O., & Barber, F. (2009). An artificial intelligence planning tool for the container stacking problem. *2009 IEEE Conference on Emerging Technologies & Factory Automation* , 1-4. doi: 10.1109/ETFA.2009.5347007.