Container Loading – Rapportage Martin Rog 1792195 29-06-2022

**Probleem**

Het importeren van goederen uit Azië gaat via 20 ft en 40 ft containers. Die containers zijn lang onderweg en nemen veel ruimte in beslag. Het kost veel geld en daarom is het ten zeerste belangrijk dat ze optimaal gevuld worden met goederen. Momenteel is het namelijk zo dat de grootste dozen eerst de container in gaan en vervolgens alle kleinere dozen daarvoor worden gepropt. Nu is het probleem dus dat de beschikbare ruimte in een vrachtcontainer niet goed wordt benut. Dit maakt het ook extra moeilijk om een geschikt gedeelte invoerprijs bij de verkoopprijs van de producten op te tellen.

**Eisen**

De product owner had twee algemene eisen die luiden als volgt:

* Een uitkomst van hoe de container optimaal moet worden ingericht.
* Er moet een passende prijs worden bepaald voor de producten om de kosten te dekken.

Om deze twee eisen te realiseren kreeg ik de volgende informatie mee:

* De afmetingen van de verschillende soorten dozen.
* De verschillende gewichten van de dozen.
* Het type container die wordt gebruikt voor het verschepen van de goederen.
* Kosten van de laatste zending

**Algoritme**

Ik heb het 3D Best Fit algorithm gekozen uit de paper van Erick Dube en Leon Kanavathy (\*1). Deze keuze heb ik gemaakt omdat dit algoritme beweert de volgende punten waar kan maken: Een gegarandeerde oplossing, oplossing wordt in een gangbare tijd gemaakt, je kan makkelijk eigen data invoeren en het verstrekt continuïteit tussen de oplossing en het probleem. Mij trokken het tweede en het derde punt het meeste aan. Aangezien ik het belangrijk vindt dat de oplossing in een snelle tijd gemaakt kan worden (dit omdat je dan snel en makkelijk verschillende tests kan runnen). Ten tweede vond ik het belangrijk dat ik eigen data makkelijk kon invoeren. Hier heb ik uiteindelijk veel profijt van gehad omdat ik werkte met data die ik had verkregen bij mij opdrachtgever.

**Project samenvatting**

In de sprint 1 ben ik aan de slag gegaan met de paper die ik toen gekozen had. Ik ben hem grondig gaan lezen en bestuderen. Na enige tijd kwam ik erachter dat het doeleinde van deze wetenschappelijke paper niet helemaal aansloot bij mijn aanpak. Het ging namelijk erg diep in op de wiskundige achtergrond van het algoritme, hoe de container indeling tot stand kwam. Het was dus zeker interessant om te lezen, ik kreeg alleen niet het gevoel dat het mij heel erg ging helpen bij het maken van mijn oplossing. Vanaf dit punt (halverwege sprint 1) ging ik verder opzoek naar andere papers. Ik ben toen een hele interessante paper tegen gekomen. Deze ging vooral diep in op het programmeren van zo'n probleem. Hier kon ik dus goed mee aan de slag.

Voor mijn oplossing aan de hand van mijn gekozen paper een python project begonnen. Ik heb gebruik gemaakt van de aanpak “Object Orientated Programming”. Dit kan je zien aan de hand van alle classes. De classes worden aangeroepen in de main. Zo wordt eerst een object “Container” aangemaakt. Deze wordt vervolgens gevuld met “Box” objecten. Tot slot worden de dozen in de container verpakt m.b.v. de “Pack” class (In deze class zit het algoritme verwerkt).

Daarna komt er een output in de terminal. Deze geeft aan welke producten wel en welke dozen niet in de container passen. Hij baseert dit op de volumes van de dozen en container, maar ook de gewichten. De output geeft ook voor elke doos zijn positie aan. De uitput gebruikt de volgende formattering: (X,Y,Z). Aan de hand van elk gegeven coördinaat, weet je hoe je de doos moet plaatsen.

Voor het testen van de code heb ik enkele edge cases bedacht. Dit zijn gekke scenario’s die erachter moeten komen of mijn code ook dat soort gevallen goed behandelt.

Voor het gedeelte van de prijsbepaling, heb ik er uiteindelijk voor gekozen om dit achterwegen te laten. In mijn uiteindelijke oplossing, komt ook de volume van alle dozen naar voren. Met behulp hiervan kan je eigenlijk heel makkelijk een prijskaartje hangen, die de kosten zouden moeten dekken.

Informatie wat betreft containers heb ik verkregen van het bedrijf FULEX. Hun website verstrekt veel handige informatie over zee containers. (\*2)

**Conclusie/evaluatie**

Ik heb tijdens het project veel afwegingen moeten maken over de aanpak van het project. Ik zat natuurlijk eerst in de knoei met mijn oorspronkelijke paper, maar zodra ik de juiste had gevonden kon ik gelukkig snel verder. Ik heb er uiteindelijk voor gekozen om geen gui toe te voegen aan mijn project. Ik ben namelijk erg lang bezig geweest met de implementatie van het algoritme. Ik heb wel een begin gemaakt aan de gui. Het draaide er eigenlijk om dat de eindgebruiker de afmetingen van een doos in kon voeren + het gewicht en de hoeveelheid. Zo kreeg het programma de input binnen en kon hij de container inpakken. De rede dat ik dit uiteindelijk toch niet heb gedaan, is omdat ik er in mijn geval niet veel meerwaarde in zag en ik geen tijd meer over had om het op nette OOP-wijze aan mijn huidige programma te koppelen. Indien ik het niét OOP zou implementeren, zou dit de netheid van verschillende classes bellemeren.

Verder denk ik dat ik het probleem goed heb weten op te lossen, aangezien ik wel aan de verdere eisen voldoe.

Een nadeel van mijn oplossing is dat het programma niet altijd even snel werkt. Dit kan bijvoorbeeld voorkomen als ik een homogene data set meegeef. Het programma moet dan best wel even runnen. Bij een invoer van bijvoorbeeld 2000 verschillende dozen, moet hij voor elke doos een ouput printen.

**Bronvermelding**

\*1. Optimizing Three-Dimensional Bin Packing Through Simulation – Leon Reeves Kanavathy

\*2 <https://www.fulex.com/ocean-containers/#:~:text=The%20maximum%20gross%20mass%20for,)%2C%20it%20is%2034%2C000%20kg>.