Bepaling van een strategie voor het plaatsen van containers op een containeryard

Kevin VERELST

promotor :
Prof. dr. Gerrit JANSSENS

Eindverhandeling voorgedragen tot het bekomen van
de graad Handelsingenieur
major operationeel management en logistiek

universiteit hasselt

Inhoudsopgave

Samenvatting

Woord vooraf

Samenvatting

Binnen het kader van mijn eindverhandeling aan de Uhasselt, heb ik modellen voor de plaatstoewijzing van containers op de *container yard* bestudeerd. Speciale aandacht heb ik hierbij besteed aan genetische algoritmen en de vraag of deze kunnen bijdragen tot de toewijzing van een opslaglocatie aan een container. Deze problematiek is belangrijk omdat het aanmeren van een schip tijd en geld kost. Hoe vlotter een schip kan geladen (of gelost) worden, hoe korter de tijd dat een schip aan een kade moet doorbrengen. Hetzelfde geldt aan de landzijde voor vrachtwagens en goederentreinen. Met andere woorden, een snelle behandeling betekent tijdwinst en in transport betekent tijdwinst op zijn beurt geldwinst.

Eerst werd overgegaan tot een uitgebreide literatuurstudie. In deze literatuurstudie wordt eerst ingegaan op onderwerpen zoals havens, containers en containeryards in het algemeen. Deze kennis is nodig als achtergrond om de complexe toewijzingsmodellen te begrijpen. In deze literatuurstudie heb ik vervolgens de bestaande operationele onderzoeksmodellen overzichtelijk ingedeeld. Eveneens kwamen enkele belangrijke aandachtspunten bij de werking van een containerterminal en de concurrentie tussen containerterminals aan het licht.

Na het afronden van de literatuurstudie werd één artikel, “An approach to determine storage locations of containers at seaport terminals” (1999) van Preston en Kozan, uitgekozen en verder uitgewerkt aan de hand van een klein softwareprogramma. De keuze voor dit artikel is hoofdzakelijk te wijten aan het feit dat de publicatie een link legt tussen informatica, operationeel onderzoek en de logistiek. Het softwareprogramma dat in deze eindverhandeling gebruikt wordt, werd speciaal voor deze eindverhandeling geschreven door doctoraatsstudent Jose Maria Pangilinan en wordt beschreven in het onderzoeksdeel van deze eindverhandeling. Het is gebaseerd op het *container location model* (CLM) beschreven in het eerder genoemde artikel en maakt gebruik van genetische algoritmen.

In het onderzoeksdeel van deze eindverhandeling worden eveneens de conclusies uit experimenten met de software neergeschreven. Deze wijken soms af van de uitkomsten in het artikel van Preston en Kozan. Deze experimenten wijzen erop dat er een verschil is in de doelfunctiewaarde, met name de transporttijd, bij het gebruik van respectievelijk FCFS (*First Come First Serve)* en LCFS (*Last Come First Serve*). Eveneens wordt er nagekeken van welke parameters de besparing in tijd door het gebruik van LCFS in plaats van FCFS afhankelijk is. Daarnaast wordt ook de invloed van verschillende parameters op de doelfunctiewaarden bij FCFS en LCFS zelf nagegaan. Zo zal bijvoorbeeld de bezettingsgraad een exponentiële werking hebben op de doelfunctiewaarden indien er gebruik gemaakt wordt van FCFS en slechts een lineaire werking vertonen bij LCFS.

Woord vooraf

Deze eindverhandeling vormt het sluitstuk van mijn opleiding tot Handelsingenieur aan de UHasselt. Het gaf mij de mogelijkheid om inzicht te krijgen in een onderwerp dat perfect past in mijn major "Operationeel Management en Logistiek", namelijk de containerterminal. Dit was dan ook een belangrijke reden om het thesisvoorstel van Prof. dr. Janssens G. getiteld "Bepaling van een strategie voor het plaatsen van containers op een containeryard", te aanvaarden. Eveneens hoop ik mijn loopbaan in de logistieke sector te kunnen uitbouwen. Een gezonde interesse in de logistieke sector heeft mij dan ook geholpen om deze eindverhandeling tot een goed eind te brengen en was tevens mijn hoofdreden om dit onderwerp te kiezen.

Onmisbaar om deze opdracht tot een goed einde te brengen was Prof. dr. Janssens. Hij gaf mij de kans om een eindverhandeling te maken over een onderwerp waarin ik belangstelling heb. Tevens stond hij mij steeds met raad en daad bij. Daarnaast wil ik eveneens de doctoraatsstudent in *Computer Science*, Jose Maria Pangilinan, bedanken voor het uitstekende werk dat hij verricht heeft bij het programmeren van de software die in deze tekst gebruikt wordt. Tot slot dank ik nog mijn familie en vrienden voor hun steun en geduld tijdens het maken van deze eindverhandeling.

Hoofdstuk I: Inleiding

I.1 Probleemstelling

De centrale onderzoeksvraag, die als rode draad doorheen deze eindverhandeling loopt, is: **"Hoe gebeurt de plaatstoewijzing van containers binnen een containerterminal en kunnen genetische algoritmen bijdragen in de efficiëntie hiervan?"**

Deze centrale onderzoeksvraag zal uitgewerkt worden aan de hand van een aantal deelvragen. De eerste vier deelvragen richten zich op enkele algemene aspecten van de containerterminal, alsook de bestaande modellen binnen *Operations Research*. Ze luiden als volgt:

"Wat is een containerterminal en uit welke delen bestaat een terminal?"

"Welke factoren zijn belangrijk in de concurrentie tussen verschillende terminals?"

"Welke ideeën bestaan er binnen het vakdomein operationeel onderzoek in verband met de plaatsing van de containers op de container yard?"

"Kunnen de eventueel bestaande modellen ingedeeld worden in categoriën?"

Daarnaast zijn er nog enkele andere deelvragen waarop in deze eindverhandeling gepoogd wordt een antwoord te formuleren. Deze antwoorden zijn in het derde hoofdstuk terug te vinden. In dat hoofdstuk worden experimenten uitgevoerd met behulp van een softwareprogramma gebaseerd op een bestaand operationeel onderzoeksmodel.

"Wat zijn genetische algoritmen en hoe passen zij in de plaatstoewijzing van containers?"

"Kan een softwarepakket op basis van genetische algoritme goede oplossingen geven voor de plaatstoewijzing van containers op de container yard?"

"Welke aandachtspunten komen aan het licht tijdens de experimenten met deze software?"

I.2 Methodologie

Eerst en vooral werd een literatuurstudie gevoerd om vanuit een diversiteit aan bronnen een goed beeld te krijgen van onder andere de *layout* en werking van containerterminals. De informatie voor deze literatuurstudie kwam voornamelijk uit wetenschappelijke artikels. Daarnaast waren handboeken en cursussen ook een veel geraadpleegde bron van informatie. Als derde wil ik een informatiebron vermelden die in onze moderne samenleving niet meer weg te denken is, namelijk het internet. Deze informatiebron was niet alleen handig om diverse publicaties in het bezit te krijgen. Ook enkele websites droegen bij tot het creëren van deze literatuurstudie.

Naast een literatuurstudie werd er eveneens een onderzoeksdeel afgewerkt binnen deze eindverhandeling. In dit deel wordt vertrokken vanuit een bepaald artikel van Preston en Kozan, namelijk “An approach to determine storage locations of containers at seaport terminals” (1999). Op basis van het model in deze publicatie werd, met hulp van de doctoraatsstudent Jose Maria Pangilinan, een softwareprogramma gecreëerd dat, door gebruik te maken van genetische algoritmen, plaatsen op de terminal toewijst aan exportcontainers die arriveren aan de landzijde. Deze programmatuur wordt na een korte beschrijving onderworpen aan een reeks experimenten om tot besluiten te komen in verband met de plaatstoewijzing van containers op de *container yard*, al dan niet in lijn met de conclusies van Preston en Kozan.

Hoofdstuk II: De bestaande modellen voor de plaatstoewijzing van containers op de container yard

II.1 Een algemene inleiding tot het transport en de havenwerking

II.1.1 De economische karakteristieken van transport

De vraag naar transport is een afgeleide vraag. Ze ontstaat uit de nood om goederen van de producent tot bij de klant te brengen. (Button, 1993) Deze nood is ontstaan door de ontwikkeling van de simpele dorpssystemen in geavanceerde productiesystemen. De bijkomende activiteiten en kosten die in deze systemen nodig zijn, wegen niet op tegen de economische schaalvoordelen van de productie in grotere hoeveelheden. (Benson et al., 1994)

De transportsector onderscheidt zich eveneens van de andere sectoren door het feit dat een relatief groot deel van de kapitaalinvesteringen per definitie mobiel van aard is. Denk hierbij aan de grote kapitaalinvestering in vrachtwagens bij transporteurs. De karakteristieken van het vaste deel van de infrastructuur kunnen ook sterk verschillend zijn met die van andere ondernemingen met betrekking tot de levensduur. De vaste componenten hebben meestal een extreem lange levensduur en zijn in vele gevallen duur om te vervangen. Zo worden momenteel nog havens gebruikt die aangelegd zijn in de tijd van het Romeinse Rijk, terwijl gebouwen van productieondernemingen zelden een leven van meer dan honderd jaar zijn beschoren. Een tweede voorbeeld zijn de kanalen die nog steeds gebruikt worden voor binnenvaart en nochtans reeds een zeer lange tijd aangelegd zijn. (Button, 1993) Verder kan transport zowel voor eigen rekening uitgevoerd worden, als uitbesteed worden aan vervoerders. (Coeck *et al*., 2006)

II.1.2 De haven in de vervoersketen

Zoals hierboven vermeld, bestaat er een behoefte aan vervoersketens. Een mogelijke vervoersmodulus is de scheepvaart. Indien hiervoor gekozen wordt, vormt de haven een schakel in de vervoersketen. (Kuiler, 1973) Een schakel die nog elk jaar aan omvang wint. Deze stijging is grotendeels te wijten aan de toename in containertransport. Zo was een stijging van 9% in containertrafiek de hoofdreden waardoor de Haven van Antwerpen in 2005 een recordhoeveelheid van bijna 160 miljoen ton vracht behandelde. (De Standaard, 2005)

De definitie van een haven is historisch uitgegroeid van "een beschutte baai waar schepen konden worden afgemeerd" naar een veel bredere definitie. Kuiler definieert een haven als: "een *terminal facility* in een goederenstroom, welke ontstaat door een zodanige combinatie van productiefactoren, dat daarin een aantal specifieke functies vervuld kunnen worden ten aanzien van zeevaart, *inland transport*, dienstverlening en industrie, en wel zodanig dat een optimaal resultaat voor de goederenstroom en de inkomensvorming wordt verkregen, zowel privaat-economisch als nationaal-economisch". (Kuiler, 1973) De Vlaamse Havencommissie of VHC definieert een zeehaven als volgt: "Een haven is een natuurlijke of kunstmatig aangelegde, veilige ligplaats voor schepen die goederen moeten laden en/of lossen, passagiers moeten in- en/of ontschepen, reparaties moeten ondergaan of waar de schepen kunnen schuilen in afwachting van beter weer. Een haven moet beschutting bieden tegen de elementen van de natuur (wind, golven en stromingen) en voorzien zijn van de nodige installaties voor het doel waarvoor de haven wordt gebruikt. Havens zijn de jongste decennia uitgegroeid tot meer dan louter plaatsen waar verschillende transportmodi samenkomen. Ze zijn ware logistieke knooppunten geworden van opslag, *value added logistics*, data-uitwisseling enz. Er zijn havens die uitsluitend bestemd zijn voor zeeschepen of voor binnenschepen, maar meestal zijn ze bestemd voor beide. Sommige havens dienen uitsluitend voor de behandeling van één soort lading. Voorbeelden daarvan zijn onder meer containerhavens, graan- en ertshavens, houthavens, petroleumhavens en stukgoedhavens." (Coeck *et al.*, 2006)

De haven heeft drie primaire functies in de goederenstoom. Ten eerste is er de oudste functie van een haven, namelijk het bieden van een ligplaats aan schepen. Dit om goederen te laden en/of te lossen of om betere weersomstandigheden af te wachten. Als er echter ook goederen worden aan- of afgevoerd van of naar het binnenland, dan heeft de haven eveneens een derde functie ten behoeve van het transport van en naar het binnenland en een overslagfunctie. (Kuiler, 1973)

Naast deze drie primaire functies vervult een haven tevens een dienstverlenende functie. Denk in deze context voornamelijk aan de handelsfunctie. Zo horen de werkzaamheden van een cargadoor of scheepsagent zeker tot het havengebeuren, maar er zijn zeker nog andere voorbeelden van dienstverlenende functies in de zeehavengebieden. Andere voorbeelden zijn de expediteur en de opslagfaciliteiten. (Kuiler, 1973) In "Haveneconomie en –logistiek" van Coeck *et al.* wordt deze functie de opslag, distributie en logistieke functie genoemd. Zij vermelden eveneens de *groupage* activiteiten van de vervoerders bij de voorbeelden. Bij *groupage* worden verschillende kleinere ladingen van verschillende verladers gebundeld en samen vervoerd. Het distributie onderdeel ontstond voornamelijk door de meestal gunstige geografische ligging ten opzichte van consumptie- en productiecentra, hun goede verbindingswegen die de prijs en de snelheid van transport gunstig beïnvloeden, hun mogelijkheden tot gespecialiseerde opslag en beschikbaarheid van competente en goed opgeleide arbeidskrachten. De opslagfunctie is de laatste jaren uitgebreid met *value added services* zoals de verpakking en etikettering. (Coeck *et al.*, 2006) Op deze manier worden er meer taken toevertrouwd aan de transportsector. Zo is het immers ook mogelijk dat het volledige voorraadbeheer uitbesteed wordt aan een logistieke dienstverlener. (Breedam, c. 2006, cursus)

Havengebieden zijn vaak ook de thuis van belangrijke industriële centra. Zowel bedrijven die verbonden zijn met de haven zoals scheepswerken, als bedrijven die afhankelijk zijn van de haven voor de aanvoer van grondstoffen vestigen zich bij voorkeur binnen de havengebieden. (Kuiler, 1973) Andere redenen zijn de aanwezigheid van een fijn vertakt

vervoersnetwerk en de mogelijkheid om diverse afzetgebieden, verspreid over de hele wereld, gemakkelijk te kunnen bereiken. (Coeck *et al.*, 2006)

Steeds meer wordt ook 'kennis en informatie' als een afzonderlijke functie van havens en het rondomgelegen logistieke centrum bekeken. Een goede informatie- en communicatietechnologie (ICT) voorziening is een steeds belangrijker wordende succesvoorwaarde om effectief in te spelen op de snel wijzigende kwaliteitseisen van het internationaal verladend en transporterend bedrijfsleven. *Third Party Logistics* (3PL) en *Fourth Party Logistics* (4PL) zijn voorbeelden van nieuwe economische activiteiten die uit deze havenfunctie gegroeid zijn. (Coeck *et al.*, 2006) Bij een 3PL partnerschap biedt de logistieke dienstverlener bijkomende waardetoevoegende diensten aan bovenop zijn operationele activiteiten. Hierbij is een informatiestroom vereist. De logistieke dienstverlener moet de informatiestroom van de verzender intensiever controleren bij een 4PL partnerschap. Hierbij wordt een deel van de *supply chain* van de verzender, bijvoorbeeld het voorraadbeheer, uitbesteed aan de logistieke dienstverlener. (Breedam, c. 2006, cursus)

II.1.3 Containers in de goederenstroom over water

Het gebruik van containers is enorm toegenomen sinds het begin van de container revolutie in april 1963. Op deze datum werd in Baltimore gestart met de bouw van de eerste containerterminal. Sindsdien is de container niet meer weg te denken uit het transport van goederen in het algemeen en via zee in het bijzonder. De container beïnvloedt niet alleen het ontwerp van schepen en de werking van havens, maar zorgde eveneens voor wijzigingen in het transport over weg of per spoor. Zelfs het opslaan van goederen ondervond invloed van de container revolutie. (Benson et al., 1994) Er ontstonden containerhavens die volgens de Vlaamse Havencommissie als volgt gedefinieerd worden: "Een containerhaven is een haven die speciaal is ontworpen en uitgerust voor de behandeling van containerschepen. Deze haven is uitgerust met containerkranen en wordt gekenmerkt door uitgestrekte terreinen achter de kaai, waar de

containers kunnen worden gestapeld en worden overgeslagen op vrachtwagens en/of treinwagons.(Coeck *et al.*, 2006)

Nog steeds blijft de container aan belang winnen in de zeevaart. Denk maar aan de ingebruikname van het Deurganckdok van 5,3 km kaaimuur en 450 m breed in Antwerpen. Dit nieuwe dok, dat geopend is sinds juli 2005, zal de Antwerpse haven toelaten haar containeroverslag meer dan te verdubbelen. Het Deurganckdok kan jaarlijks 6,4 miljoen TEU (*twenty feet equivalent unit*; zie II.2) behandelen als het volledig operationeel is. (Port of Antwerp, 2006, online)

Op wereldvlak kan de toename in containerisatie aangetoond worden aan de hand van de containerisatiegraad (percentage van de vervoerde goederen dat vervoerd werd per container). Deze evolueerde van 21,8 % in 1980 naar 59,0 % in 2004. (Coeck *et al.*, 2006)

Het aantal kleinere *general cargo* schepen, die voor verschillende transporten kunnen ingezet worden, krimpt. Ze worden vooral vervangen door grotere containerschepen die, zoals de naam het al doet vermoeden, enkel kunnen gebruikt worden voor containervervoer. Dit versterkt de containerisatie. (Coeck *et al.,* 2006) De opmars van het containervervoer is natuurlijk eveneens te verklaren door het rijke gamma aan voordelen die de container biedt:

- Containers consolideren ladingen: ze maken namelijk een grotere eenheidslading van een aantal kleinere eenheden.
- Deze eenheidsladingen worden sneller en eenvoudiger behandeld waardoor de laad- en lostijden verkort worden.
- Vaak is er door het gebruik van containers minder verpakking nodig. Op deze manier kunnen regelmatig meer goederen geladen worden dan vroeger. Belangrijk is hier wel dat de containers secuur geladen worden. Anders zouden goederen beschadigd kunnen worden als de container bewogen wordt.
- De goederen kunnen eenvoudiger gedragen worden vermits ze beter vast gemaakt zijn dan losse cargo.

- Het verplaatsen zelf is voordeliger vermits een volgeladen container een beter gebruik van een bepaald volume opslagruimte in een schip voorstelt.
- Diefstal wordt tegengegaan. Het is immers moeilijker om met een volledige container aan de haal te gaan dan met een eenheid losse cargo.
- Een eenvoudigere documentatie is mogelijk. Eenzelfde hoeveelheid volume vraagt minder documentatie.
- De verzekeringskosten dalen omdat niet elke verpakking afzonderlijk behandeld moet worden.
- Door de mogelijkheid van het gebruik van containers zowel bij weg-, spoor- als zeevervoer is de container de ideale laadeenheid voor multimodaal en doorvoerverkeer.

Met andere woorden komt het succes van de container in het vervoer niet geheel onverwacht. (Benson *et al*., 1994)

Er zijn echter ook enkele nadelen verbonden aan het gebruik van containers. Sinds de start van het gebruik van containers doken er verscheidene technische en operationele problemen op. Deze zijn echter in grote mate opgelost en wogen niet op tegen de immense voordelen. (Benson *et al.,* 1994)

II.1.4 De containerterminal

II.1.4.1 Definiëring

Het gebruik van de container gaf binnen de haveninfrastructuur aanleiding tot het ontstaan van nieuwe faciliteiten. Eén van deze faciliteiten is de containerterminal.

Een containerterminal kan gedefinieerd worden als een ruimte toegewezen aan de opslag van containerladingen, meestal bereikbaar per vrachtwagen, spoor en maritiem transport. Er worden containers opgepikt, afgeleverd, gehandhaafd en gehuisvest. (The

transportation institute, 2006, online) Gezien deze activiteiten is een terminal dus meer dan een aanlegplaats. De terreinen van de terminal bevinden zich dan ook achter de kademuur. (Coeck *et al.*, 2006)

Essentieel voor een containerterminal zijn de aanwezigheid van voldoende ruime terreinen, de aanwezigheid van goede verbindingen met het hinterland, het feit dat de behandeling 24 uur op 24 het jaar rond kan gebeuren, het snel bereikbaar zijn van een ligplaats voor schepen en de aanwezigheid van voldoende beschutting tegen wind en golven. (Coeck *et al.*, 2006)

Een reden waarom deze terminals zijn ontstaan, is de nood aan een groter werkterrein bij containervervoer. De oorzaak hiervan ligt bij de hogere snelheid waartegen containers aan land kunnen gebracht worden. De transportmiddelen voor verder vervoer kunnen niet snel genoeg meer tot aan het schip gebracht worden om rechtstreeks overgeladen te worden. (Benson et al., 1994)

II.1.4.2 De onderdelen van een containerterminal

Als de lay-out van een containerterminal bekeken wordt, kunnen de volgende onderdelen afgeleid worden:

- Voorkaai: Dit is het gedeelte tussen de zone waar de containers gestapeld worden en het water. De breedte van deze zone kan verschillen naargelang de beschikbare ruimte. In deze zone bevinden zich sporen waarop de portainers (specifieke term voor de containerkranen op de voorkade) zich kunnen voortbewegen over de volledige lengte van de kaaimuur.
- Containerterrein: Hier worden de containers ontvangen, geleverd en gestapeld. Binnen dit terrein bevinden zich eveneens het rangeerterrein, de stockageplaats voor lege containers en het chassisterrein.
- Rangeerterrein: Deze zone, die veel oppervlakte vereist, wordt gebruikt om de containers in lijn te zetten. Dit onderdeel is opgedeeld in *slots* waarin de

containers, die moeten geladen worden, volgens het stuwageplan geplaatst worden.

- Controlecentrum: Deze staat in voor de controle en supervisie van de operaties en moet daarvoor geplaatst worden op een plaats met overzicht over de volledige terminal.
- Gate in/gate out: Hier worden de containers ontvangen van en geleverd aan vrachtwagens.
- Container freight station: In dit overdekte magazijn worden containeractiviteiten uitgevoerd zoals onder andere *stuffen* en *strippen*. Hier worden eveneens de *less than container loads* (LCL) ontvangen en gestockeerd. (Men spreekt van LCL wanneer de rederij of de vervoerder van containers ook partijen goederen, die in volume kleiner zijn dan de inhoud van een container, ter vervoer aanneemt. De vervoerder zal verschillende dergelijke kleine partijen bundelen tot hij er één container kan mee vullen.)

Naast deze hoofdcomponenten is er meestal ook een onderhoudsafdeling aanwezig. Het personeel van deze afdeling inspecteert en repareert containers en hun eventuele koelsystemen. Ook zijn deze werknemers verantwoordelijk voor het onderhoud van de behandelingsuitrusting. (Coeck *et al.*, 2006)

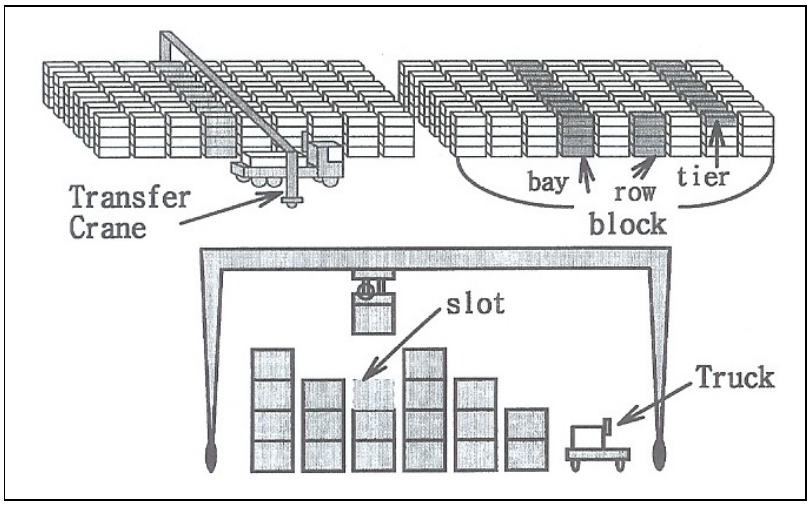
II.2 Algemene inleiding tot de terminologie rondom een containerterminal

In dit onderdeel zullen enkele belangrijke termen in verband met containers en de behandeling van containers op de *container yard* verklaard worden. Kennis van deze termen zal onmisbaar blijken doorheen het vervolg van deze tekst.

Vooreerst is er de container. “Dit zijn de grote stapelbare laadkisten, vervaardigd uit aluminium of (roestvrij) staal, waarin de goederen geladen worden zoals in de laadruimte van een vrachtwagen of een spoorwegwagon.” (Witlox, c. 2004, cursus) Containers vormen de basis van een concept van eenheidslading. Zo moeten de goederen die erin opgeslagen zijn niet bij elke transfer uitgepakt worden. De containers kunnen eenvoudig en snel verplaatst worden. (Steenken *et al.*, 2004)

TEU of *twenty feet equivalent unit* is een algemeen aanvaarde standaardmaat voor containers. Deze werd vastgelegd door de “International Standard Organisation”. De breedte van een TEU-container bedraagt 8 voet en de lengte bedraagt 20 voet. (Witlox, c. 2004, cursus) Deze afmetingen komen overeen met 2,4384 m breedte en 6,096 m lengte.

Eveneens is het belangrijk om kennis te hebben van de structuur binnen een containeropslagplaats. Een *container yard* bestaat uit verschillende *blocks*. Elke *block* bestaat uit twintig tot dertig *bays,* die meestal elk een zestal rijen tellen. (Kim *et al., 2000*) Deze worden ook wel *stacks* of stapels genoemd. Zo een stapel ontstaat door het op elkaar stapelen van containers. Gemiddeld heeft een stapel een hoogte van drie tot vier containers. De hoogte tot waarop een container gestapeld is, noemt men de *tier*. (Kim en Kim, 1999) Het maximale aantal *tiers* wordt bepaald door stapeluitrusting. (Steenken *et al.*, 2004) Figuur 1 vat de structuur samen in een duidelijke schematische voorstelling.



Figuur 1: De structuur binnen een containeropslagplaats (Kim et al., 2000)

De positie van een container op de opslagplaats of *slot* wordt beschreven aan de hand van de *block*, de *bay*, de *row* en de *tier*.

Doordat containers in een opslagplaats gestapeld worden, hebben de stapel- en transportmachines geen rechtstreekse toegang tot elk van de laadkisten. Met de term *rehandle* (of *reshuffle*) duidt men de handeling aan die nodig is wanneer andere containers zich bovenop de container bevinden die men nodig heeft op een bepaald moment. Deze bovenliggende laadkisten moeten in dat geval eerst verplaatst worden. (Steenken *et al.*, 2004)

II.3 Redenen voor de optimalisatie van de werking van de containeropslagplaats.

Een eerste reden voor de optimalisatie is het toenemende plaatsgebrek binnen de havens. Er moeten steeds meer containers opgeslagen worden in de havens omdat het containerverkeer continu blijft groeien. De ruimte waarover men beschikt moet dus zo efficiënt mogelijk gebruikt worden. (Steenken *et al.*, 2004)

Tevens is de graad van klantenservice een belangrijk aandachtspunt bij het beheren van een container terminal. Dit komt door een sterke concurrentie tussen de verschillende container terminals. Een gewenste importcontainer moet zo snel mogelijk kunnen afgeleverd worden aan de vrachtwagen of trein die op de container wacht. Indien er veel herplaatsingen (*rehandles*) nodig zijn voordat de transferkranen de juiste container kunnen grijpen, zal de truck of de trein langer moeten wachten. (Kim en Kim, 1999) Een betere methode voor het toewijzen van containers aan bepaalde plaatsen op de container opslagplaats verhoogt ook de snelheid waarmee een export container van de opslagplaats naar de *marschalling area* of laadplaats kan worden gebracht. (Preston en Kozan, 1999) Dit verlaagt de totale laadtijd en kan op zijn beurt de benodigde tijd van een schip aan een bepaalde ligplaats beperken.

II.4 Ondersteunend model

Een evaluatie van het aantal verplaatsingen (*rehandles*) die nodig zullen zijn, is onmisbaar in de zoektocht naar een ideale techniek voor het toewijzen van de beschikbare opslagplaatsen aan containers. Hiervoor zal teruggegrepen worden naar het werk van Kap Hwan Kim in het artikel "Evaluation of the number of rehandles in container yards" (1997) en de in de appendix A van "Segregating space allocation models for container inventories in port container terminals" (1999).

Bij deze evaluatie wordt uitgegaan van twee veronderstellingen (Kim, 1997):

- De analyse van de *rehandles* wordt beperkt tot één baai. Hieraan gekoppeld is de veronderstelling dat elke verplaatste container verplaatst wordt naar een nieuwe plaats binnen dezelfde baai.
- Er wordt verondersteld dat er geen containers toegevoegd worden aan de containerstapel tijdens het willekeurig oppikken van de te verplaatsen containers.

Als gevolg van de tweede veronderstelling kan gesteld worden dat containers van twee verschillende containerschepen niet gemengd zullen worden binnen dezelfde containerbaai.

Het aantal te verwachten verplaatsingen voor een willekeurige doelcontainer is afhankelijk van verschillende factoren die rechtstreeks zijn verbonden met de *layout* van de baai:

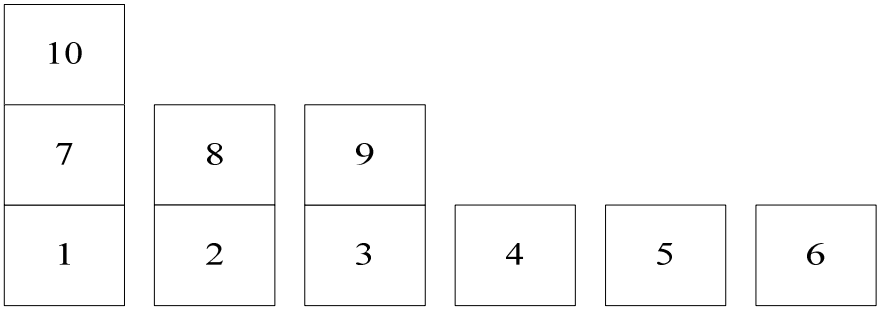
- het totaal aantal containers
- het aantal rijen
- de verdeling van de stapelhoogten van de containers in de baai

Deze laatste verdeling is beperkt door de maximale hoogte van de stapel die toegelaten wordt door de capaciteit van de transferkraan. Verder wordt eveneens een vereenvoudigende veronderstelling aan de methode toegevoegd. Er wordt namelijk verondersteld dat het effect van de plaatspositie niet significant is. Met andere woorden is

er geen verschil in het aantal verwachte verplaatsingen tussen de *layout* (3, 2, 1, 0, 0, 0) en het geval (0, 3, 0, 2, 0, 1). In deze notatie voor de *layout* van een baai stelt elk cijfer het aantal opeengestapelde containers in een bepaalde rij voor. (Kim, 1997)

Bij deze methode koos Kap Hwan Kim voor de volgende notaties (Kim, 1997):

- a het aantal rijen binnen een *block*
- b het aantal baaien binnen een *block*
- c het aantal *tiers* binnen een *block*
- $(n_1,n_2,n_3,\ldots,n_a)$ de configuratie van een baai



Figuur 2: Voorbeeld van een (3,2,2,1,1,1) baaiconfiguratie (Kim, 1997)

- $v(n_1,n_2,n_3,\ldots,n_a)$ het verwacht aantal verplaatsingen indien de volgende doelcontainer willekeurig gekozen wordt en $(n_1,n_2,n_3,\ldots,n_a)$ de huidige configuratie is.
- c_{kj} de j^{de} configuratie van de baai met k containers
- $p_k(i,j)$ de kans dat de configuratie van een baai veranderd wordt van c_{ki} naar $c_{(k-1)j}$ wanneer een willekeurige container in de baai opgepikt wordt
- s_{kj} de kans dat een baai met k containers de j^{de} configuratie heeft
- $v_k(i,j)$ het aantal verplaatsingen nodig om over te gaan van configuratie c_{ki} naar $c_{(k-1)j}$
- v_k het verwachte aantal verplaatsingen voor het nemen van een volgende, willekeurige container als de baai k containers telt

Onder de assumptie dat alle containers een gelijke kans hebben om de volgende doelcontainer te worden, kan het verwachte aantal verplaatsingen voor het opnemen van deze container eenvoudig berekend worden. Toegepast op de configuratie in figuur 2 krijgen we bijvoorbeeld het volgende:

v(3,2,2,1,1,1) = 0,1 * (het aantal containers met twee containers boven zich) * (het aantal verplaatsingen noodzakelijk wanneer twee containers zich boven de doelcontainer bevinden)
+ 0,1 * (het aantal containers met één container boven zich) * (het aantal verplaatsingen noodzakelijk wanneer één container zich boven de doelcontainer bevindt)
= (0,1 * 1 * 2) + (0,1 * 2 * 1)
= 0,4

Als we dit voorbeeld nu doortrekken, krijgen we in figuur 3 de vier mogelijk nieuwe configuraties als 1 container werd verwijderd uit de oorspronkelijke configuratie van dit voorbeeld. De eerste regel in deze figuur geeft weer dat (3,2,2,1,1,0) de eindconfiguratie is van (3,2,2,1,1,1) wanneer container 2, 3, 4, 5 of 6 verwijderd wordt. Dus in 5 van de tien gevallen zal deze eindconfiguratie bereikt worden. Wat leidt tot de kans van 0,5. In het geval container 2 of 3 verwijderd wordt, zal telkens één verplaatsing uitgevoerd moeten worden om de eindconfiguratie te bereiken. Dit in tegenstelling tot nul bij de drie overige gevallen waar de gewenste containers onmiddellijk opgepakt kunnen worden. (Kim, 1997)

Resulterende configuratie	Kans	Containernummer van de opgenomen container om tot deze situatie te komen (aantal verplaatsingen)
(3,2,2,1,1,0)	0,5	2(1), 3(1), 4(0), 5(0), 6(0)
(3,2,1,1,1,1)	0,2	8(0), 9(0)
(2,2,2,2,1,0)	0,1	1(2)
(2,2,2,1,1,1)	0,2	7(1), 10(0)

Figuur 3: De resulterende configuratie na één opname in de baai (3,2,2,1,1,1) (Kim, 1997)

In de voorgaande figuur stelt c_{93} de stapelconfiguratie (2,2,2,2,1,0) voor. Als we nu de beginconfiguratie (3,2,2,1,1,1) noteren als $c_{(10)1}$, dan is $p_{10}(1,1) = 0{,}5$, $p_{10}(1,2) = 0{,}2$, $p_{10}(1,3) = 0{,}1$ en $p_{10}(1,4) = 0{,}2$. Laat P_k vervolgens de matrix zijn met als element (i,j) de waarde $p_k(i,j)$. De kans dat een baai met k containers de j^{de} configuratie heeft (*state probability*), s_{kj}, kan daardoor recursief geëvalueerd worden door:

$$S_k = S_{k+1} + P_{k+1} \qquad \text{(A)}$$

met S_k de rijvector waarvan het j^{de} element s_{kj} is. (Kim, 1997)

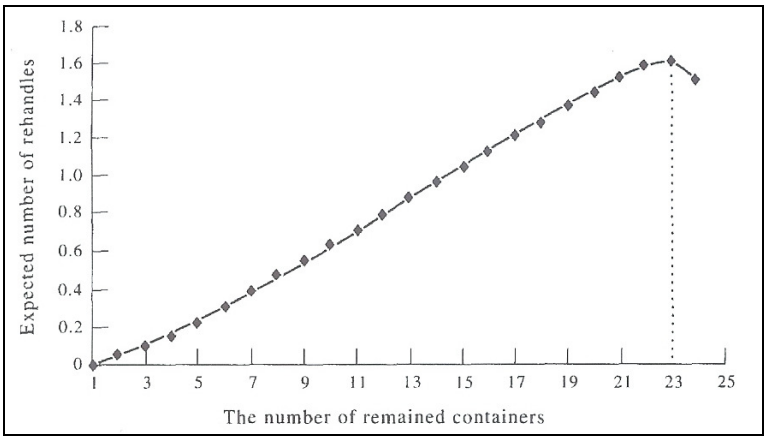
Veronderstel dat er k containers in de baai aanwezig zijn en de configuratie van die baai c_{ki} is. Dan kan het verwachte aantal verplaatsingen voor de volgende *pick-up* uitgedrukt worden als $\sum_j p_k(i,j)v_k(i,j)$ waarin $v_k(i,j)$ het aantal verplaatsingen is noodzakelijk voor de overgang van c_{ki} naar $c_{(k-1)j}$. Dus wordt voor het aantal containers k de formule voor het aantal verwachte verplaatsingen:

$$v_k = \sum_i s_{ki} \sum_j P_k(i,j) v_k(i,j) \qquad \text{(B)}$$

Vervolgens wordt er verondersteld dat het initiële aantal containers n is en dat de initiële stapelconfiguratie, S_n, gegeven is. De probabiliteit van elke staat voor k = n-1, n-2, …, 1 wordt dan geëvalueerd aan de hand van formule (A) en het verwachte aantal verplaatsingen voor de volgende *pick-up* wordt geëvalueerd voor alle k containers door middel van vergelijking (B). Gebruikmakend van deze resultaten kan eveneens het totaal aantal verplaatsingen berekend worden om alle containers uit de baai op te nemen en te verwijderen. Dit gebeurt door het optellen van alle verwachte verplaatsingen bekomen via formule (B) voor k = n tot k = 1.

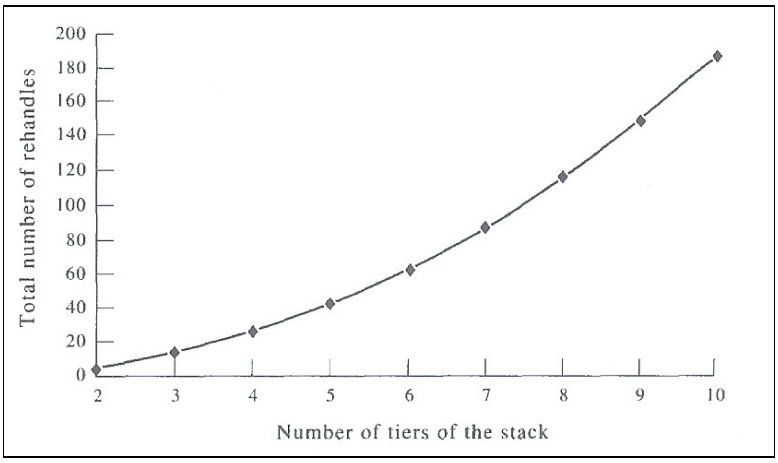
Uit de uitwerking van een voorbeeld waarbij een baai 6 rijen telt en een initiële hoogte van 4 aanwezig is, trekt Kim nog de volgende conclusies. De *state probability* kan significant verschillen naargelang éénzelfde aantal containers op verschillende manieren gestapeld zijn. Eveneens neemt het aantal verwachte verplaatsingen toe naargelang het aantal aanwezige containers stijgt. Wanneer de resultaten van het voorbeeld uitgezet worden in een grafiek kan hier zelfs duidelijk een lineair verband onderscheiden worden.

Als we deze tabel in een grafiek uitzetten zoals in figuur 4 wordt duidelijk dat de deze relatie lineair is tot vlak voor het einde van de curve, waar de lijn afbuigt naar beneden.

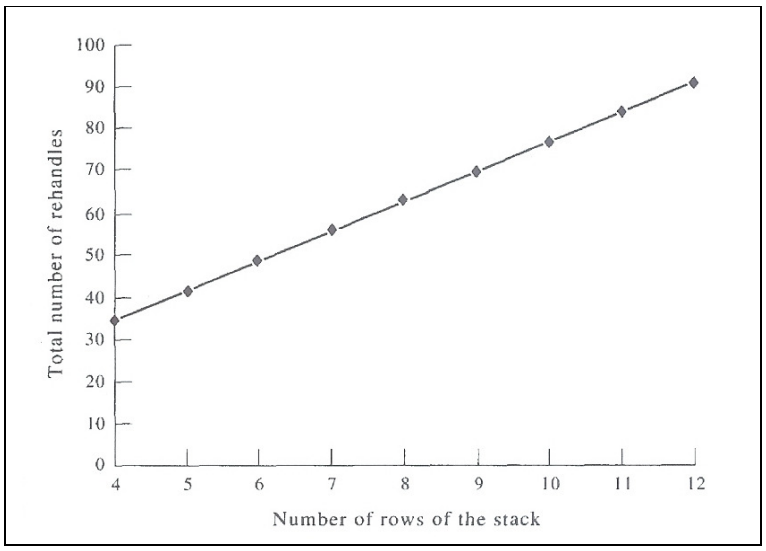


Figuur 4: Het verwachte aantal verplaatsingen bij de volgende afhaling in functie van het aantal overblijvende containers in de baai (a = 6, c = 4) (Kim, 1997)

Maar Kim toont niet enkel aan dat het aantal *rehandles* afhankelijk is van het aantal aanwezige containers, maar eveneens van de breedte en hoogte binnen een baai. Figuur 5 toont het aantal noodzakelijke verplaatsingen in functie van het initiële aantal *tiers*. Met andere woorden toont deze grafiek de evolutie van het aantal geschatte verplaatsingen naargelang de maximale hoogte van de stapels op de yard. Dit gebeurt in de grafiek bij een vastgelegd aantal rijen. De invloed van het aantal rijen binnen een baai op het geschatte aantal benodigde verplaatsingen bij een maximale hoogte van 6 wordt weergegeven in figuur 6.



Figuur 5 : Het totale aantal verplaatsingen in functie van het initiële aantal *tiers* binnen de stapeling (Kim, 1997)



Figuur 6: Het totale aantal verplaatsingen in functie van het aantal rijen binnen de stapeling (Kim, 1997)

Uit de voorgaande grafieken kan afgeleid worden dat het geschatte aantal verplaatsingen een exponentiële invloed ondervindt van de maximale hoogte en slechts een lineaire invloed van het aantal rijen binnen de baai. Er kan dus geconcludeerd worden dat het geschatte aantal benodigde *rehandles* sterker afhankelijk is van de maximale hoogte binnen de baai dan het aantal rijen binnen diezelfde baai. (Kim, 1997)

Vervolgens werd een benaderende formule afgeleid voor de berekening van het verwachte aantal verplaatsingen. Het verwachte aantal verplaatsingen wordt benaderd door een lineaire lijn die door de oorsprong (0,0) gaat. Dit wordt getoond in figuur 7. Deze lineaire lijn benadert de originele waarden zeer goed. Er moet echter opgemerkt worden uit figuur 7 dat het aantal verplaatsingen niet altijd monotoon daalt als het aantal resterende containers (k) daalt. Bij het oppikken van de eerste container moeten de hiervoor verplaatste containers bovenop de hoogste *tier* van een grenzende rij geplaatst worden. Dit leidt tot een sprong op het punt k = ac – 1. Pas na dit punt, als het aantal resterende containers verder afneemt, zal het aantal verplaatsingen monotoon dalen. Om de ideale lijn te vinden die door (0,0) gaat en de berekende data het dichtst benadert, moet er bij het vastleggen van het ander eindpunt rekening gehouden met de sprong op k = ac – 1. (Kim, 1997)

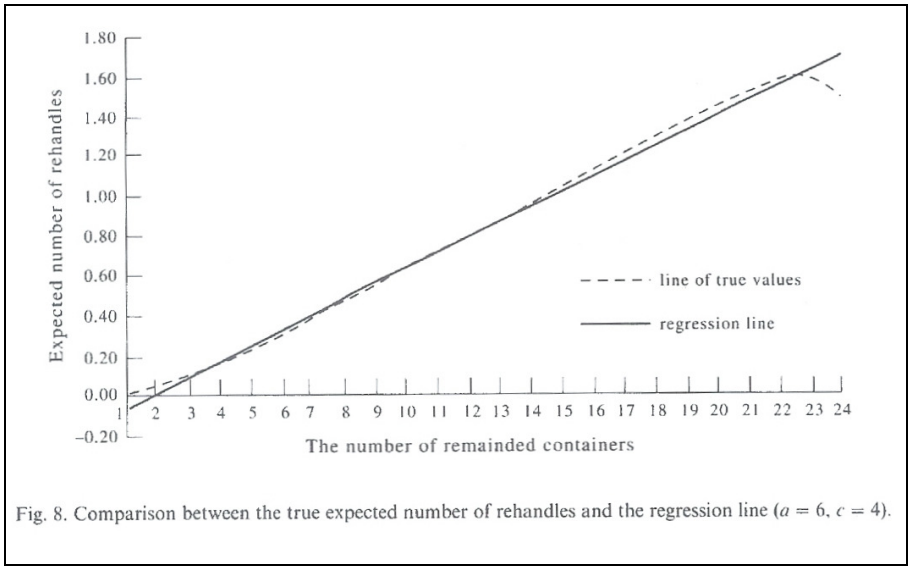
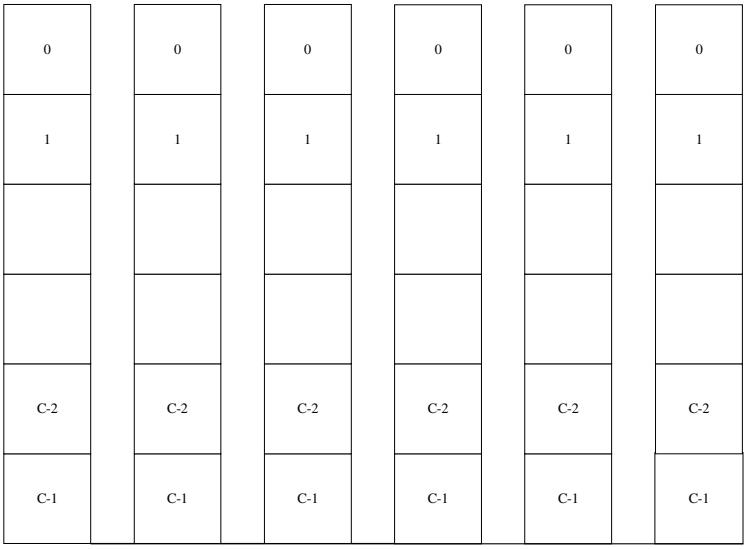


Fig. 8. Comparison between the true expected number of rehandles and the regression line ($a = 6$, $c = 4$).

Figuur 7: vergelijking tussen het werkelijke aantal verwachte verplaatsingen en de regressielijn (a=6, c=4)

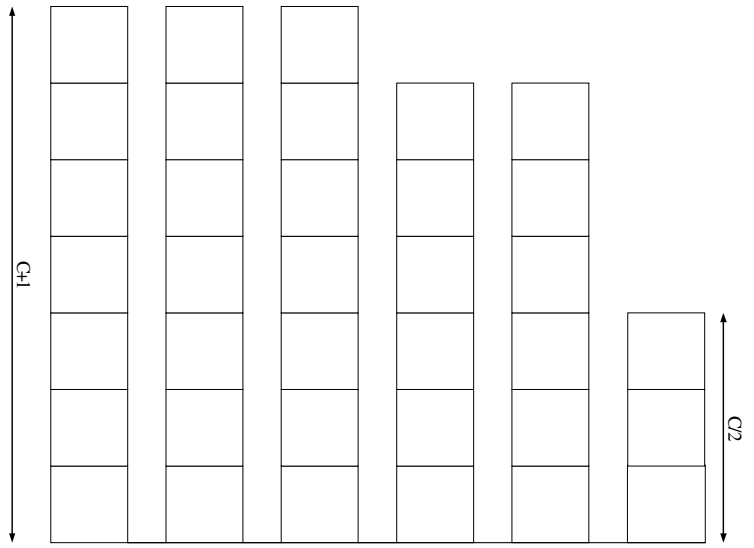


Figuur 8: De initiële configuratie van de baai (Kim, 1997)

Als we veronderstellen dat Figuur 8 de initiële configuratie toont van de baai, zullen (c-k) containers verplaatst moeten worden om een container in de k^{de} *tier* te bereiken. Het aantal benodigde verplaatsingen voor elke container wordt ook getoond in figuur 8. Dus wordt op het punt k = ac het verwacht aantal verplaatsingen voor de volgende *pick-up* geschat door:

$$\begin{aligned} v(a,c,k) &= (a/ac)\{1+2+3+...+(c-1)\} \\ &= (1/c)\{c(c-1)/2\} \\ &= (c-1)/2 \end{aligned}$$

met *a* het aantal rijen en *c* de maximale hoogte. (Kim, 1997) Deze formule maakt echter een onderschatting van het verwachte aantal benodigde verplaatsingen. De veronderstelling van een gelijke hoogte bij elke stapel binnen de baai is echter te optimistisch. Er kan wel vanuit gegaan worden dat de hoogten van alle stapels, behalve van degene waarvan de gewenste container wordt weggenomen, niet meer van elkaar verschillen dan één niveau. De oorzaak hiervan ligt in het feit dat men de verplaatste containers steeds zal plaatsen op zo laag mogelijk gelegen beschikbare plaatsen. (Kim en Kim, 1999)



Figuur 9: De aangepaste configuratie van de stapeling (Kim, 1997)

De gemiddelde hoogte van de stapel waar het laatst een container van uit de baai genomen is, is c/2. Het gevolg hiervan is het gebruik van een configuratie als in figuur 9 in plaats van de schatting te baseren op een situatie als in figuur 8. Als deze twee figuren vergeleken worden, kan opgemerkt worden dat een bijkomende container geplaatst wordt op (c/2) rijen. Om deze reden moet bij voorgaande formule c(c/2)/(ac) opgeteld worden. De bijdrage van de laatste rij, waarin de gewenste container zich bevond, zal niet langer {1 + 2 + … + (c-1)}/(ac) bedragen, maar [1 + 2 + … + {(c/2) - 1}]/(ac). Hiertussen is een verschil van $(3c^2 - 2c)/(8ac)$. De totale aanpassing in de eerder gegeven vergelijking wordt: $c(c/2)/(ac) - (3c^2 - 2c)/(8ac) = (c + 2)/(8a)$. Het eindpunt van de benaderde lijn wordt dan $(ac, (c - 1)/2 + (c + 2)/(8a))$ en vervolgens wordt de helling van deze lijn $(c - 1)/(2ac) + (c + 2)/(8a^2c)$. Het verwachte aantal verplaatsingen noodzakelijk bij de volgende afhaling in een baai met k containers wordt dan:

$$\hat{v}(a,c,k) = \{(c-1)/(2ac) + (c+2)/(8a^2c)\}k$$

Het verwachte aantal verplaatsingen kan dus geschat worden als volgt:

$$\hat{v}(a,c) = \int_0^{ac} \left(\frac{c-1}{2ac} + \frac{c+2}{8a^2c} \right) x dx$$
$$= ac(c-1)/4 + c(c+2)/16$$

(Kim, 1997)

II.5 Bestaande modellen voor de optimalisatie van de containeropslagplaats

Binnen de bestaande modellen met het doel om optimaal gebruik te maken van de beschikbare ruimte in een containerterminal kunnen twee grote groepen onderscheiden worden. Meestal behandelen de modellen enkel het geval van import- of exportcontainers. Enkel in "Storage space allocation in container terminals" (2003) wordt het opslagprobleem in een complexe containerterminal met zowel import-, export- als transitcontainers bestudeerd door Zhang et al..

De reden voor deze opsplitsing is dat de inkomende en uitgaande containers verschillende eigenschappen bezitten en daardoor verschillende aandachtspunten belangrijk zijn. Exportcontainers worden gewoonlijk naar de terminal gebracht door vrachtwagens en hun aankomsttijden zijn daarom vaak oncontroleerbaar. Eveneens kunnen de opslaglocaties op een schip van exportcontainers met dezelfde eindbestemming en gewicht omgewisseld worden. De stapelregelingen voor deze containers zijn dus ook gelijk vermits ze dezelfde inspanningen vereisen voor hun ophaling. Bij exportcontainers wordt de meeste aandacht besteed aan de toewijzing van opslaglocaties aan groepen containers. In tegenstelling hiermee zijn er de importcontainers die op vastgelegde tijdstippen in groepen aankomen. Dit type containers vertrekt in het merendeel van de gevallen per vrachtwagen uit de terminal. Het verwachte aantal bewegingen is het belangrijke aandachtspunt bij deze categorie van laadkisten. (Zhang *et al.*, 2003)

Ook zal hier uitgegaan worden van geladen containers. De distributie van lege containers naar havens wordt immers als een afzonderlijk probleem met gespecialiseerde aanpakken beschouwd. Nochtans verschillen de methoden van opslaan en stapelen ervan niet met die van volle containers. Vaak worden de lege containers wel afzonderlijk opgeslagen omdat ze door het gebruik van verschillend materiaal hoger kunnen gestapeld worden. (Steenken *et al.*, 2004)

II.5.1 Modellen vertrekkende vanuit het gegeven van exportcontainers

Vooreerst zullen enkele modellen beschreven worden waarbij de containers, die het onderwerp vormen van het model, geëxporteerd zullen worden.

II.5.1.1 Vermindering van laad- en ligtijd van de schepen als doelstelling

In "An approach to determine storage locations of containers at seaport terminals" (1999) beschrijven P. Preston en E. Kozan hun model met het objectief een optimale opslagstrategie te bepalen, die de noodzakelijke tijd minimaliseert om de containers van de opslagplaats naar de laadplaats binnen de perimeter van het schip te verplaatsen. Ze gaan ervan uit dat als deze tijd geminimaliseerd wordt, ook de *turn-around time* van het containerschip zal verkleinen. Dit wil zeggen dat het schip niet zo lang aangemeerd moet liggen. Preston en Kozan willen dit doel bereiken door de ontwikkeling van een *container location model* (CLM) en dit model op te lossen door middel van genetische algoritmen. (Preston en Kozan, 1999) Omdat het verdere onderzoek in hoofdstuk 3 gebaseerd is op deze paper zijn de verdere details opgenomen in dat latere hoofdstuk.

In 1998 publiceerden Kap Hwan Kim en Jong Wook Bae in "Computers & Industrial Engineering" het artikel: "Re-marshaling export containers in port container terminals". Dit artikel heeft als doel het laden van schepen efficiënter te doen verlopen en de beschikbare opslagruimte zo efficiënt mogelijk te gebruiken. Enigszins zijn de doelstellingen gelijklopend met die van het model van Preston en Kozan, maar anderzijds pakken Kim en Bae het probleem op een verschillende manier aan.

Zij stellen een methodologie voor om de huidige schikking van exportcontainers op een containeropslagplaats om te zetten in beste *yard layout* met het oog op het latere laden van het schip. Het doel is om een zo klein mogelijk aantal containers en/of een zo klein

mogelijke afstand te vinden en zodoende de totale *turn-around time* van een schip in de haven te minimaliseren. Kim en Bae splitsen het probleem op en mathematische modellen werden opgesteld voor de drie subproblemen: *bay matching*, *move planning* en *task sequencing*. Heuristische algoritmen werden gebruikt om deze subproblemen op te lossen wegens de anders te tijdrovende berekeningen. (Steenken *et al.*, 2004) Deze herschikking zal meestal plaats vinden net voordat een bepaalde hoeveelheid ruimte wordt toegewezen aan een overeenkomstig containerschip en is nodig omdat bij het aankomen van de containers in de haven de gedetailleerde informatie in verband met de ladingsvolgorde vaak nog niet gekend is. De oorspronkelijke *layout* van de terminal zal dus niet ideaal zijn. (Kim en Bae, 1998)

In “Re-marshaling export containers in port container terminals” tonen Kim en Bae de door hun voorgestelde techniek aan de hand van een voorbeeld, dat hier verder niet letterlijk zal overgenomen worden. In dit voorbeeld worden de eerste twee subproblemen samen genomen. Eerst zal elke bestaande baai gelinkt worden met een specifieke configuratie in de doelschikking (*bay matching*). Het tweede subprobleem, dat samen zal uitgevoerd worden met het eerste, is het bepalen van het aantal containers dat moet verplaatst worden van een *bay* met een surplus aan containers van een bepaalde groep naar een *bay* die meer containers van die groep nodig heeft om aan de ideale schikking te voldoen (*move planning problem*). (Kim en Bae, 1998)

Voor de *bay matching* wordt dynamische programmering gebruikt. (Kim en Bae, 1998) Dynamisch programmeren is een techniek voor het oplossen van problemen waarbij een oplossing gezocht wordt voor een klein onderdeel van het originele probleem. Waarna vervolgens het probleem stelselmatig uitgebreid wordt en nieuwe optimale oplossingen gevonden worden op basis van de voorgaande optimale oplossing totdat het originele probleem in zijn geheel opgelost is. (Hillier en Lieberman, 2001) Op basis van de resultaten hiervan wordt een bewegingsplanning opgesteld door middel van het transportprobleem techniek. Het kan echter voorvallen dat het *transportation problem* een oplossing oplevert die interferentie tussen de transferkranen veroorzaakt. Deze kranen kunnen immers niet gelijktijdig functioneren binnen een bepaalde afstand van elkaar. In

dit geval wordt de combinatie van *bays* die problemen veroorzaakt, opgenomen in de lijst van verboden combinaties (*matches*). Vervolgens wordt de bovenvermelde procedure herhaald tot een uitvoerbare oplossing gevonden is. (Kim en Bae, 1998)

Eens de *bay matching* en *move planning* afgelopen zijn, kan men de bewegingstaken voor alle groepen samenvatten in één tabel. Een voorbeeld van zo een tabel vindt u hieronder in figuur 10. Deze figuur zal nadien gebruikt worden als basis voor het *task sequencing problem*. (Kim en Bae, 1998)

Taaknummer	Bronbaai	Bestemmings-baai	# te verplaatsen containers
1	02	10	4
2	02	12	1
3	04	12	6
4	07	12	2
5	01	10	6
6	01	11	3
7	04	11	2
8	06	02	5
9	11	02	2
10	05	02	5
11	16	02	5
12	04	07	7
13	12	07	3
14	12	06	5
15	15	06	7
16	16	06	3
17	06	11	8

Figuur 10: De lijst met bewegingstaken (Kim en Bae, 1998)

Het *task sequencing problem* heeft als doel de bewegingstaken te schikken zodat de totale *travel time* van de behandelingsmachines wordt geminimaliseerd. Er moet echter wel opgemerkt worden dat sommige bewegingen niet kunnen uitgevoerd worden voordat er voldoende plaats vrijkomt in de bestemmingsbaai. Het probleem kan beschreven worden als een *travelling salesman problem* met prioriteitsbeperkingen waarbij de prioriteitsrelaties uitgedrukt worden in de vorm van een conditie, namelijk de conditie dat

de plaats in de bestemmingsbaai vrij moet zijn voordat de bewegingstaak begint. Of de plaats al dan niet vrij is, is afhankelijk van de aan de taak voorafgaande verplaatsingen van containers. Na de probleemoplossing wordt niet alleen de ideale opeenvolging van taken bekomen, maar eveneens de minimale tijd nodig voor de *re-marshaling*. Om dit tweede resultaat te kunnen berekenen moet echter de omschakeltijd van de transferkranen om taak j uit te voeren na taak i gekend zijn. (Kim en Bae, 1998)

Een derde model dat in deze categorie thuis hoort is ontwikkeld door Kap Hwan Kim en Kang Tae Park en gepubliceerd in het artikel "A note on a dynamic space-allocation method for outbound containers". Dit artikel uit 2003 werd gepubliceerd in "European journal of operational research". Het optimale gebruik van ruimte is een tweede doelstelling, naast de doelstelling in verband met het efficiënter laden van een schip, waaraan men in het artikel wil voldoen. (Kim en Park, 2003) Met deze doelen in het achterhoofd tonen Kim en Park een dynamische plaats-toewijzingsmethode (*dynamic space allocation method*). Ze formuleren een basis MIP-model en zullen 2 heuristieken toepassen. Deze beslissingsregels zullen nog steeds goede resultaten opleveren, maar sneller zijn in berekening. (Steenken *et al.*, 2004) De auteurs nemen ook de veronderstelling aan dat een derde soort container, naast de inkomende en uitgaande, in de context van behandeling en opslag gelijk is aan de uitgaande container. Dit derde type noemt men de *transschipment* container. Zij worden van een boot geladen, voor een bepaalde tijd opgeslagen op de containeropslagplaats en vervolgens terug op een ander schip geladen. (Kim en Park, 2003)

De selectie van een opslagplaats voor een uitgaande container kan meestal opgesplitst worden in twee stadia: het stadium van de toewijzing aan een ruimte en het lokaliseringstadium van een individuele container. (Kim en Park, 2003)

In de eerste fase wordt de hoeveelheid ruimte van elke *block* bepaald die toegewezen zal worden aan aankomende containers voor een bepaald schip. Deze beslissing kan maandelijks, wekelijks, dagelijks, enz. genomen worden. Toekomstige schepen en de

invoer- en uitvoerstroom moeten in rekening gebracht worden bij de ruimtetoewijzing. (Kim en Park, 2003)

Tijdens het tweede stadium moet een beslissing genomen worden over de exacte opslagplaats van elke uitgaande container die in de terminal toekomt. Op deze tweede fase concentreert dit artikel zich niet, maar verwijst hiervoor naar "Deriving decision rules to locate export containers in container yards" (2000) van Kap Hwan Kim, Young Man Park en Kwang-Ryul Ryu. Dit artikel is later in deze tekst opgenomen.

Het model dat voorhanden is, concentreert zich dus, in tegenstelling tot de voorgaande, niet op het selecteren van een specifieke opslagplaats voor één bepaalde container, maar op het selecteren van een grotere ruimte, die toegewezen wordt aan alle containers bedoeld voor een bepaald schip. Desalniettemin moeten er enkele veronderstellingen vastgesteld worden voor het opstellen van het mathematische model:

- De ruimte die vrij wordt gemaakt tijdens een periode, wordt pas beschikbaar op het einde van de periode.
- Het aantal aankomende containers voor elk schip over de planningshorizon is deterministisch en gekend.
- Het behandelingsmateriaal van de opslagplaats behandelt maar één container tegelijkertijd.
- De reiskosten van een *straddle carrier* of andere behandelingsmaterialen binnen éénzelfde *block* worden buiten beschouwing gelaten. (Kim en Park, 2003)

Bij de opstelling van het mathematische model heeft men als doel de afstand tussen de opslagplaats en de laadkranen van het schip te bepalen en wordt gebruik gemaakt van volgende notaties (Kim en Park, 2003):

- L de verzameling van schepen waaraan ruimte moet toegewezen worden gedurende de planningshorizon.
- n het aantal stages. De tijdsperiode tussen twee opeenvolgende aftochten van schepen wordt een "stage" genoemd.

- T_j de verzameling van schepen waaraan ruimte is ingenomen of toegewezen in stage *j*.
- S_j de verzameling van stages waarin plaats moet toegewezen worden aan schip *i*.
- M de verzameling van *blocks*.
- b_i het maximale aantal *blocks* waarin laadkisten van schip *i* mogen opgeslagen worden.
- d_{ij} het aantal uitgaande containers voor schip *i*, die zullen aankomen in de terminal tijdens stage *j*.
- C_k de maximale stapelcapaciteit van *block k*.
- x_{ijk} de ruimte in *block k* toegewezen aan de toekomende containers van schip *i* tijdens stage *j*.
- x_{i0k} het initieel aantal containers opgeslagen in *block k* die in schip *i* geladen zullen worden.
- c_1 de reiskost van het behandelingsmateriaal per container per eenheid reisafstand tussen de perimeter van het schip en de opslagplaats.
- c_2 de reiskost van het behandelingsmateriaal per eenheid afstand binnen de opslagplaats.
- t_{ik} de reisafstand van het behandelingsmateriaal tussen laadplaats van schip *i* en *block k*.
- $S_{k_1 k_2}$ de reisafstand van het behandelingsmateriaal tussen *block* k_1 en k_2.
- δ_{ik} $\begin{cases} 0, \ x_{ijk} = 0 \ \ voor\, alle\ j \in S_i \\ 1, \ anders \end{cases}$

Bij het samenstellen van de doelfunctie vertrekt men vanuit volgend dwingend doelelement: de totale leverkost. De leverkost per container van *block k* tot de laadplaats is $2c_1t_{ik}$ en dus is de totale kost als volgt:

$$2c_1 \sum_{i \in L} \sum_{k \in M} t_{ik} \left(x_{i0k} + \sum_{j \in Si} x_{ijk} \right)$$

Dit element wordt echter aangevuld met een optioneel doelelement. Namelijk het principe dat containers steeds binnen een bepaalde perimeter van het schip opgeslagen worden. Op deze manier wordt de reistijd beperkt. De reiskost van het behandelingsmateriaal kan als volgt geschat worden:

$$C2\sum_{i\in L} K_i Z_i$$

Hiervoor wordt wel verondersteld dat de totale reisafstand proportioneel is tot de langste afstand tussen twee *bays* waar containers voor het schip in kwestie gelokaliseerd zijn. K_i is een parameter die deze langste afstand converteert in de totale reistijd. De parameter is tevens afhankelijk van het aantal containergroepen. Z_i is de langste afstand tussen twee baaien die laadkisten voor schip *i* bevatten. (Kim en Park, 2003)

Eveneens zijn in het model dwingende en optionele beperkingen opgenomen. De eerste dwingende beperking benadrukt de beperkte beschikbare ruimte. De som van de ingenomen en toegewezen plaatsen mag op geen enkel ogenblik de totale opslagcapaciteit van een *block* overschrijden. De nood om te voldoen aan de behoefte aan ruimte van elk schip in elke stage is de tweede dwingende beperking in dit mathematisch model.

Een derde beperking is niet dwingend, maar optioneel. Deze beperking vertaalt de limiet op het aantal *blocks* die per schip mogen gebruikt worden.

$$\sum_{j\in S_i} x_{ijk} \le B\delta_{ik}, \quad voor \quad i\in L,\ k\in M,$$

waar B een zeer groot getal is

$$\sum_{k\in M} \delta_{ik} \le b_i, \quad voor\ i\in L$$

In de eerste regel van deze beperking wordt de variabele δ_{ik} gedefinieerd. Deze variabele heeft een waarde 1 als er containers van schip *i* opgeslagen liggen in *block k*. In het andere geval heeft deze variabele een 0 waarde. De tweede regel specificeert het aantal *blocks* die gebruikt mogen worden voor schip *i*. De beperkingen zijn gerelateerd aan een bepaald schip en de parameters van een beperking kunnen verschillen van schip tot schip. (Kim en Park, 2003)

Vervolgens wordt aangenomen dat X_i een tweedimensionale matrix voorstelt met als elementen alle x_{ijk}-waarden. Dit met $1 \leq j \leq n$ en $k \in M$. Het l^{de} optioneel doelelement voor schip *i* zal genoteerd worden als $f_{il}(X_i)$. In dat geval kan de doelfunctie geschreven worden als:

$$Min\, 2c_1 \sum_{i \in L} \sum_{k \in M} t_{ik} \left(x_{i0k} + \sum_{j \in Si} x_{ijk} \right) \sum_{i \in L} \sum_{l} f_{il}\left(X_i\right)$$

Laat eveneens de v^{de} optionele beperking, gerelateerd tot schip *i*, $g_{iv}(X_i) = 0$ zijn. Dan:

$$\sum_{i \in T_j} \left(x_{i0k} + \sum_{s \in S_i, s \leq j} x_{isk} \right) \leq C_k \quad voor\ j = 1, 2, ..., n, \quad k \in M \tag{1}$$

$$\sum_{k \in M} x_{ijk} = d_{ij}, \quad voor\ j = 1, 2, ..., n, \quad i \in L \tag{2}$$

$$g_{iv}(X_i) = 0, \ i \in L, \ v = 1, ..., w \quad (w\ is\ het\ aantal\ optionele\ beperkingen)$$

$$x_{ijk} \geq 0 \ \ voor\ alle\ i, j, en\ k \tag{3}$$

Voor de oplossing van dit stelsel worden twee heuristieken gegeven. De eerste methode noemt men de *least duration-of-stay* methode of DOS. In deze methode krijgen containers met een lagere verwachte verblijfsduur een hogere prioriteit toegewezen. Het algoritme loopt van stage 1 tot de laatste stage. In elke stage zullen de containers, met een behoefte voor een opslagplaats, gerangschikt worden naar toenemend vertrektijdstip van hun schip. De opslagplaats waaraan de laagste kosten gekoppeld zijn, wordt vervolgens gekoppeld aan de eerste container in de rangschikking. Dit wordt daarna voor de rest van de laadkisten herhaald tot alle behoeften aan opslagplaatsen ingewilligd zijn. Vermits deze heuristiek ruimte sequentieel toewijst, wordt geen rekening gehouden met de effecten die een allocatie voor één schip kan hebben op de allocatie voor andere schepen.

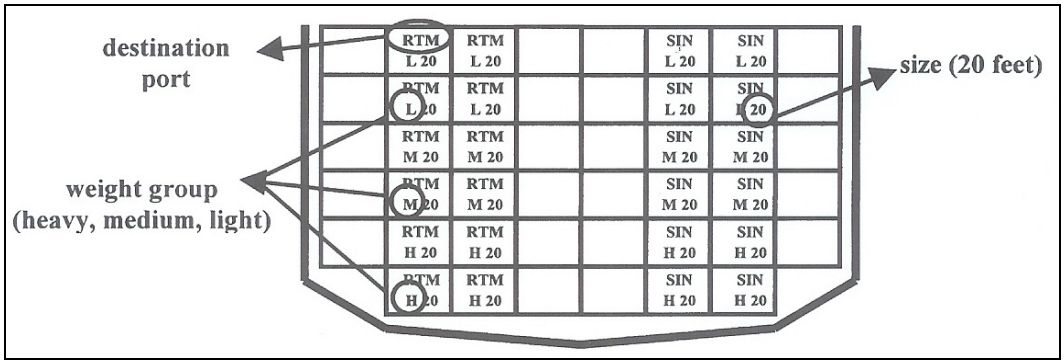
De tweede heuristiek is een *sub-gradient optimization heuristic algorithm*. Hierbij beschrijven Kim en Park nauwkeurig hoe het hierna volgende basisprobleem (P), dat onderhevig is aan de beperkingen (1), (2) en (3), kan opgelost worden via het toevoegen van een beperking, het ontspannen van een andere beperking, Lagrangemultiplicatoren, enz. Voor deze verdere uitwerking verwijs ik u graag door naar de paper "A note on a dynamic space-allocation method for outbound containers" van Kim en Park (2003).

Numerieke experimenten voor verschillende combinaties van parameters, die niet worden overgenomen uit het artikel van Kim en Park, wijzen uit dat de resultaten met de DOS-methode en de methode op basis van *sub-gradient optimization* niet sterk verschillen. Nochtans eist die tweede methode veel meer berekentijd. Verder beweren Kim en Park dat uit deze experimenten conclusies getrokken kunnen worden over de invloed van de verschillende parameters op de doelwaarde en het gebruik van de beschikbare ruimte.

II.5.1.2 Een minimum aan relocations als doelstelling

Een eerste bestaand model met als doelstelling de minimalisatie van het aantal verplaatsingen van containers tijdens het laden, is het model uit 2000 van Kap Hwan Kim, Young Man Park en Kwang-Ryul Ryu in het "Deriving decision rules to locate export containers in container yards". Dit is een dynamisch programmeringmodel dat rekening houdt met de configuratie van de containerstapeling, de gewichtsverdeling van de containers in de opslagplaats, en het gewicht van de aankomende containers. Eveneens leiden ze voor beslissingen in *real-time* een beslissingsboom af van het pakket optimale oplossing, die geleverd worden door dynamisch programmeren. (Steenken *et al.*, 2004) De efficiëntie van de beslissingsboom zal daarnaast ook geëvalueerd worden. Dit zal gebeuren aan de hand van het aantal beslissingen in de beslissingboom die verschillen van de optimale oplossingen bij de tragere oplossingsmethode op basis van dynamisch programmeren. (Kim *et al.*, 2000)

Zoals hierboven vermeld, is het uiteindelijke doel van dit model het optimaal opslaan van binnenkomende containers zodat een finale *layout* ontstaat waaruit de laadplanner een efficiënte laadsequentie kan opstellen. Men vertrekt hierbij vanuit een dwarsdoorsnede van het schip waarin de containers moeten geladen worden voor transport. Een voorbeeld van zo een dwarsdoorsnede wordt gegeven in figuur 11. Men kan duidelijk de cellen onderscheiden waartoe de containers van twee groepen (gedefinieerd door bestemmingshaven en grootte) worden toegewezen. (Kim *et al.*, 2000)



Figuur 11: voorbeeld van een dwarsdoorsnede van een schip (Kim *et al.*, 2000)

Indien het efficiënt laden van een schip het doel is, is het vanzelfsprekend dat men steeds zal proberen de in de ladingssequentie opeenvolgende containergroepen, in naast elkaar gelegen *bays* te plaatsen. Zo wordt het aantal onnodige bewegingen van een transferkraan geminimaliseerd. (Kim *et al.*, 2000)

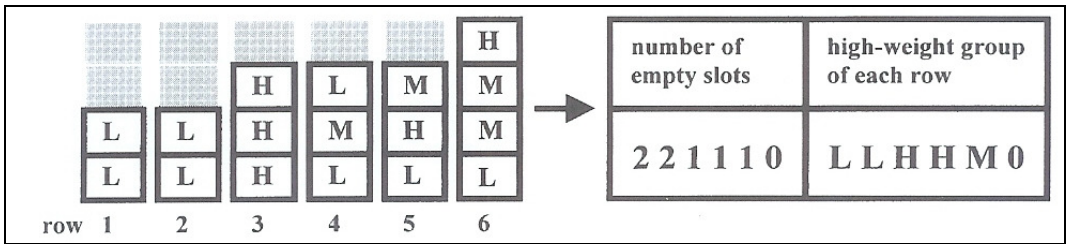
Bij het vervolg van het model gaat men uit van enkele principes. Containers van verschillende groepen worden niet samen in één baai geplaatst en de baaien die tot éénzelfde schip zijn toegewezen moeten gelokaliseerd zijn in één of twee *blocks*, die naast elkaar gelegen zijn in de werkrichting van de transferkranen. Op deze manier wordt de reistijd van de transferkranen beperkt. Merk ook op dat de containers dieper in het ruim gestapeld worden naargelang ze tot een zwaardere gewichtsklasse behoren. (Kim *et al.*, 2000)

Bij de formulering van de methodologie wordt niet enkel rekening gehouden met de 2 vermelde principes, maar er worden eveneens enkele veronderstellingen gemaakt voor de mathematische formulering:

- Omdat de aankomende laadkisten moeten worden ingedeeld in gewichtsgroepen, veronderstelt men dat het containergewicht gekend is. Zo kan men de nodige verdeling van de containers naar containergewicht maken. In de realiteit kan de actuele verdeling slechts gekend zijn als alle laadkisten aangekomen zijn. Om deze reden zal men de gewichtsverdeling schatten uit empirische data uit het verleden.
- Alle containers van een zwaardere gewichtsgroep zullen in het schip geladen worden voor de containers van een lichtere groep. In werkelijkheid zullen soms lichtere containers geladen worden voor zwaardere. Maar algemeen mag men stellen dat de veronderstelling correct is.
- De aankomende vrachtwagens zullen bediend worden op basis van *first-in-first-out* (FIFO). Deze methode is niet alleen fair ten opzichte van de chauffeurs, maar het zou eveneens onpraktisch zijn om op éénzelfde moment twee containers wachtende te hebben op een plaats binnen dezelfde containerbaai.
- Een container zal niet meer dan één keer verplaatst worden. Deze laatste veronderstelling heeft als doel het probleem niet te complex te laten worden. De auteurs gaan ervan uit dat de fout, afkomstig van deze veronderstelling, beperkt zal blijven. (Kim *et al.*, 2000)

Naast deze veronderstellingen zijn ook enkele definities noodzakelijk om het model volledig te kunnen begrijpen. De “stage” is het aantal lege plaatsen in een *yard-bay*. De “staat” van elke *stage* is het aantal lege plaatsen in elke rij en het gewicht van de zwaarste container. Dit gewicht van de zwaarste container wordt de hoog-gewicht groep (*high-weight group*) genoemd. De rijen worden gesorteerd naargelang het aantal vrije plaatsen. Dit sorteerproces zal de kwaliteit van de oplossing echter niet schaden vermits enkel het aantal verplaatsingen als doel gesteld wordt. In het geval dat er twee rijen met een gelijk aantal lege plaatsen aanwezig zijn, krijgt de rij met de hoogste gewichtsgroep voorrang. Van deze sortering wordt een voorbeeld gegeven in figuur 12. In deze figuur wordt een

systeem gegeven met vier *tiers*, zes rijen en drie gewichtsgroepen. De staat van de configuratie van dit systeem wordt uitgedrukt als (221110LLHHM0). (Kim *et al.*, 2000)



Figuur 12: Illustratie van een staat-voorstelling.

De laatste opdracht voor aan het uitwerken van het model kan begonnen worden, is het introduceren van enkele notaties (Kim *et al.*, 2000):

- N het aantal stages
- n het stage-nummer (het aantal vrije plaatsen)
- X_n de staat bij de start van de n^{de} stage
- k_n de gewichtsgroep van de aankomende container in stage n
- D_n het nummer van de rij waartoe de aankomende container wordt toegewezen in stage n
- $p_n(k_n)$ de kans op het aankomen van container van gewichtsklasse k_n
- $r_n(X_n,D_n,k_n)$ het verwachte aantal marginale verplaatsingsbewegingen bijgevoegd in stage n als een inkomende container van gewichtsklasse k_n toegewezen wordt aan rij D_n en de inputstaat X_n is.
- $t_n(X_n,D_n,k_n)$ de staat transferfunctie die X_n, k_n en D_n in X_{n-1} brengt

Aan de hand van deze notaties wordt de volgende vergelijking bekomen:

$$f_n(X_n) = \sum_{k_n} p_n(k_n) \underset{D_n}{Min}\left[r_n(X_n, D_n, k_n) + f_{n-1}(X_{n-1}) \right],$$

$$n = 1, 2, \ldots, N,$$

$$f_0(X_0) = 0$$

Aan de hand van deze vergelijking kan de optimale opslaglocatie geselecteerd worden voor een inkomende container van een bepaalde gewichtsklasse.

Een cijfervoorbeeld uit "Deriving decision rules to locate export containers in container yards" (2000) van Kap Hwan Kim, Young Man Park en Kwang-Ryul Ryu, zal verduidelijken hoe het model in werkelijkheid in zijn werk gaat. Beschouw een systeem van 4 *tiers* en zes rijen met: N = 24, k_1 = Heavy (H), k_2 = Medium (M), k_3 = Light (L), $p_n(k_i) = 1/3$ voor i = 1, 2, 3 en n = 1, 2, … , 24. In stage 1 (n = 1) is er maar één beschikbare plaats en bestaat er dus geen keuze over waar de inkomende laadkist moet geplaatst worden. Vermits de container tot drie gewichtsklassen kan behoren, kunnen er na het plaatsen drie verschillende staten optreden. Deze worden getoond in figuur 13. De eerste regel in deze tabel geeft steeds de optimale rij D^*_1 en de onderste regel geeft de waarde van $\underset{D_1}{Min}\left[r_1(X_1, D_1, k_1) + f_0(X_0)\right]$.

Inputstaat	k_1			
	H	M	L	$F_1(X_1)$
100000H00000	1	1	1	2/3 = 1/3{0+1+1}
	0	1	1	
100000M00000	1	1	1	1/3 = 1/3{0+0+1}
	0	0	1	
100000L00000	1	1	1	0 = 1/3{0+0+0}
	0	0	0	

Figuur 13: De optimale beslissingen bij n = 1 (Kim *et al.*, 2000)

In stage 2 (n = 2) zijn er twee vrije plaatsen beschikbaar en zodoende zijn negen verschillende inputstaten. De optimale beslissingen zijn opgenomen in figuur 14 met op de eerste rij de optimale rij en op de tweede rij $\underset{D_2}{Min}\left[r_2(X_2, D_2, k_2) + f_1(X_1)\right]$.

Inputstaat	k_2			
	H	M	L	$f_2(X_2)$
200000H00000	1	1	1	4/3
	2/3	5/3	5/3	
200000M00000	1	1	1	7/9
	2/3	1/3	4/3	
200000L00000	1	1	1	1/3
	2/3	1/3	0	
110000HM0000	1	2	1	7/9
	1/3	2/3	4/3	
110000HL0000	1	2	2	4/9
	0	2/3	2/3	
110000ML0000	1	1	2	/
	0	0	1/3	
110000HH0000	1	1	1	4/3
	2/3	5/3	5/3	
110000MM0000	1	1	1	2/3
	1/3	1/3	4/3	
110000LL0000	1	1	1	0
	0	0	0	

Figuur 14: De optimale beslissingen bij n = 2 (Kim *et al.*, 2000)

Het beslissingproces voor X_2 = 110000HM0000 zal nu als illustratie verder worden toegelicht. Stel k_2 = M, dan

(1) als $D_2 = 1$ (plaats M op H),

$r_2(X_2,D_2,k_2) = r_2(110000HM0000, 1, M) = 1$,

$X_1 = t_2(X_2,D_2,k_2)$

$= t_2(110000HM0000, 1, M)$

$= 100000M00000$,

$f_1(X_1) = 1/3$

(2) als $D_2 = 2$ (plaats M op M),

$r_2(X_2,D_2,k_2) = r_2(110000HM0000, 2, M) = 0,$

$X_1 = t_2(X_2,D_2,k_2)$

$= t_2(110000HM0000, 2, M)$

$= 100000M00000,$

$f_1(X_1) = 2/3$

Dus, $\underset{D_1}{Min}\left[r_2(110000HM0000, D_2, M) + f_1(X_1)\right]$

$= \text{Min}\{1 + 1/3, 0 + 2/3\} = 2/3$

Aldus, $D^*_2 = 2$

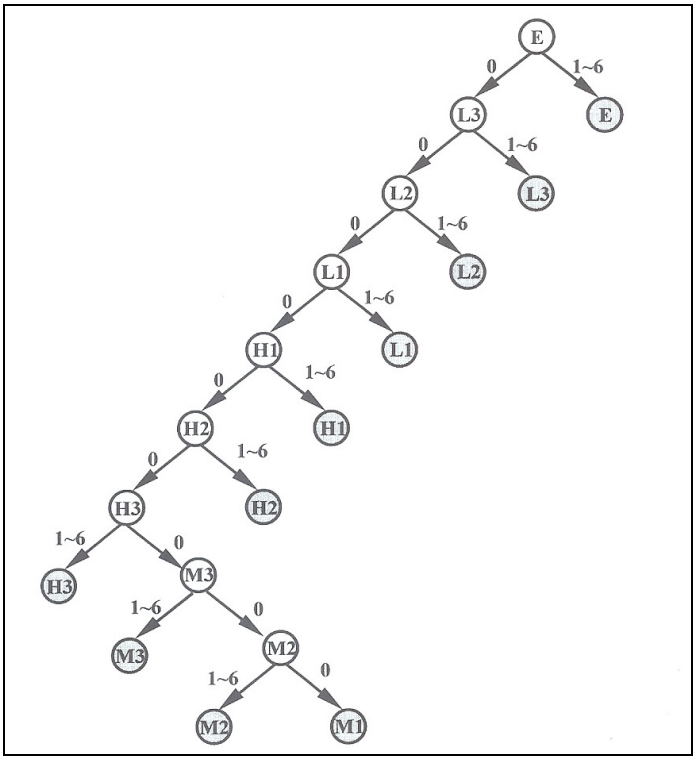
Op identieke manier wordt de optimale oplossing voor $k_2 = H$ en $k_2 = L$ bepaald. Tenslotte bekomen we $f_2(X_2) = 1/3\{1/3 + 2/3 + 4/3\} = 7/9$.

De resultaten in de tabel moeten echter nog geïnterpreteerd worden. Dit gebeurt als volgt. Als bijvoorbeeld 110000HM0000 de beginsituatie is, dan plaatst men een inkomende H-container best op de eerste rij, een inkomende M-container best op de tweede rij en een L-container best op de eerste rij. Als hieraan vastgehouden wordt, kan een totaal aantal verplaatsingbewegingen van 7/9 (= 0,777778) verwacht worden. (Kim *et al.*, 2000)

Vanzelfsprekend vereist voorgaande beslissingsmethode in realiteit te veel berekentijd. Daarom ontwikkelden de auteurs van "Deriving decision rules to locate export containers in container yards" (2000) een beslissingsboom voor de bepaling van de opslaglocatie. Voor het opstellen van deze beslissingsboom wordt gebruik gemaakt van de *branch-and-bound* methode. (Kim *et al.*, 2000) Dit is een techniek die in operationeel onderzoek met succes wordt toegepast voor het oplossen van een integer programmeringsprobleem. Het basisconcept achter de *branch-and-bound* techniek is het principe van "Divide et impera" (Verdeel en heers). Het originele, grote probleem wordt immers verdeeld in steeds kleinere problemen tot deze subproblemen makkelijk te veroveren zijn. Het verdelen, *branching*, gebeurt door de totale set van mogelijk oplossingen op te splitsen in steeds kleinere subsets. Het veroveren, *fathoming*, gebeurt deels door het beoordelen hoe goed de beste oplossing van een subset kan zijn (*bounding*) en de subset te laten vallen indien

het oordeel aantoont dat de subset onmogelijk een optimale oplossing voor het originele probleem kan bevatten. *Branching*, *bounding* en *fathoming* zijn dan ook de drie stappen in het *branch-and-bound* proces. (Hillier en Lieberman, 2001)

Voor meer informatie over hoe het opstellen van de beslissingsboom in zijn werk gaat, verwijs ik u door naar "Deriving decision rules to locate export containers in container yards" (2000) van Kap Hwan Kim, Young Man Park en Kwang-Ryul Ryu. Wel moet duidelijk vermeld worden dat voor elke gewichtsklasse een aparte beslissingsboom opgesteld moet worden omdat de beslissingsregels verschillend zijn naargelang de gewichtsklasse van de toekomende container. Figuur 15 is de beslissingsboom voor een toekomende container van gewichtsklaase L. De *critical value of pruning* is in deze figuur 0,4. Deze waarde is een drempelwaarde. De fout die afkomstig is van het *pruning* mag niet groter zijn. (Kim *et al.*, 2000)



Figuur 15: Beslissingboom voor een inkomende L container (CVP = 0,4) (Kim *et al.*, 2000)

Zoals eerder vermeld, zal de beslissingsboom geen perfecte resultaten opleveren. Kim *et al* tonen in hun artikel aan dat het aantal foute beslissingen zeer beperkt is in deze snellere methode.

II.5.2 Modellen vertrekkende vanuit het gegeven van importcontainers

In containeropslagplaatsen worden er echter niet enkel containers opgeslagen die bedoeld zijn voor uitvoer. De schepen kunnen ook containers aanleveren die bedoeld zijn voor het achterland. Deze laadkisten moeten ook opgeslagen worden tot ze door vrachtwagens of een ander transportmodus opgehaald worden.

Tot op vandaag is er nog maar weinig onderzoek verricht naar de plaatstoewijzing van importcontainers. Dit vanwege het dynamische karakter van deze klasse van containers. (Kim en Kim, 1999) De aankomsten van schepen en containers zijn immers variabel in de tijd en moeilijk exact te bepalen. Naast lijnvaart met vaste routes, stopplaatsen en tijdstippen bestaat er eveneens “wilde vaart”. Dit zijn ongeregelde vervoerdiensten over zee die niet volgens een vast schema aangeboden worden. (Witlox, c. 2004, cursus)

Het artikel van Kap Hwan Kim en Hong Bae Kim uit 1999, getiteld “Segregating space allocation models for container inventories in port container terminals”, legt zich toe op de toewijzing van opslagruimte aan importcontainers. Er wordt toegespitst op drie gevallen naargelang de opeenvolging van de containeraankomsten constant, cyclisch of dynamisch is. Het doel is de minimalisatie van het totaal aantal *rehandles*. (Steenken *et al.*, 2004) Met het oog op dit doel worden de ruimtes toegewezen per aankomend schip. Als het aantal *rehandles* kan beperkt worden, zal de *turn-around time* van de vrachtwagens verkleinen. Dit kan een belangrijk concurrentieel voordeel opleveren voor de uitbater van de containerterminal. (Kim en Kim, 1999)

Voor de ruimtetoewijzing van importcontainers bestaan er twee strategieën: de segregatiepolitiek en de niet-segregatiepolitiek. Bij de segregatiepolitiek wordt een lege plaats toegewezen wanneer de importcontainers van elk schip geladen worden. In het geval van een niet-segregatiepolitiek worden baaien met lege plaatsen geselecteerd. Daarin plaatst men dan de afgeladen containers tot ze tot een vooraf bepaald niveau gevuld zijn. (Kim en Kim, 1999)

Zoals eerder vermeld, heeft de segregatietechniek één groot voordeel. Namelijk het vlotter verloop van het afhalen van de importcontainers door vrachtwagens of een andere transportmodus. De oorzaak hiervan ligt in het voorkomen dat containers waarvan verwacht wordt dat ze snel worden opgehaald, begraven worden. Als de segregatiestrategie gebruikt wordt, moeten de volgende elementen in rekening gebracht worden: het aankomsttijdstip van de containers, het aantal containers die moeten uitgeladen worden uit elk schip en de *duration-of-stay* van elke container in de opslagplaats. Deze informatie kan onder andere uit het aanmeerschema gehaald worden. Ook beschikken containerterminals meestal enkele dagen voor de aankomst van een schip over vooruitzichten met het aantal te laden en af te laden containers. Omdat de meeste terminals een extra kost aanrekenen als een container langer blijft dan de vrije tijdslimiet, mag aangenomen worden dat slechts enkele containers langer blijven dan deze vrije tijdslimiet. De *free time limit* is een maximale grens op de *duration-of-stay* en wordt bepaald door de terminaloperatoren. (Kim en Kim, 1999)

Vooreerst wordt het geval met een constant aankomsttempo onder de loep genomen. Zoals steeds gelden ook hier enkele veronderstellingen (Kim en Kim, 1999):

- Geen enkele container blijft in de containerterminal na de vrije tijdslimiet.
- Een container die moet verplaatst worden, wordt verplaatst naar een andere plaats binnen dezelfde baai. Dit is gemakkelijk te verantwoorden als er een significante setup tijd is voor het bewegen van de transferkraan tussen twee baaien.
- Er wordt gebruik gemaakt van de segregatiepolitiek. Dit betekent dat containers die gedurende verschillende tijdsperiodes afgeladen worden, niet gemixt worden

binnen dezelfde baai. De tijdens een periode toegewezen ruimte kan enkel terug vrijgegeven worden na de vrije tijdslimiet.

- Het behandelingsmateriaal van de terminal bestaat uit een transferkraan.
- Er is geen informatie beschikbaar over de opeenvolging van de afhalingen van de importcontainers.

De notaties die zullen gebruikt worden in dit model zijn de volgende (Kim en Kim, 1999):

- b the aantal baaien
- r het aantal stapels in een baai
- t de periode-index
- h_t de gemiddelde hoogte van een containerstapel voor de container uitgeladen in periode t
- A_t het totaal aantal containers (in TEU) uitgeladen tijdens periode t
- N de planningshorizon
- n de vrije tijdslimiet (*free time limit*)
- c de lengte van de cyclische behoefte aan ruimte (hier aangenomen als c = 7)
- R(x) het verwachte totaal aantal verplaatsingen nodig om alle containers in de baai op te pikken in het geval dat het beginaantal containers in de baai x is

Vermits de initiële hoogt van de containerstapels h_t is, is het gemiddelde initieel aantal containers in een baai $h_t r$. Dus $R(h_t r)$ kan geschat door middel van volgende formule:

$$R(h_t r) = h_t r(h_t - 1)/4 + h_t(h_t + 2)/16$$

Deze formule is een toepassing van de bevindingen van het model besproken in *II.4*. Omdat we eerst uitgaan van een geval met een constant aankomsttempo kunnen we A_t voor elke periode t gelijkstellen aan A. Het verwachte totaal aantal verplaatsingen tijdens periode t kan desgevallend geschreven worden als $\{h_t r(h_t - 1)/4 + h_t(h_t + 2)/16\}A/(h_t r)$. Het aan de formule toegevoegde deel, $A/(h_t r)$, beslaat het benodigde aantal baaien om te voorzien in de behoefte van de in periode t afgeladen containers. Verder zal deze term

verkort weergegeven worden in de vorm van x_t en als gevolg hiervan kan de vorige formule herschreven worden als volgt:

$$\{1 + 1/(4r)\}(A^2/4r)/x_t + \text{constante}$$

x_t en h_t worden hier beschouwd als continue variabelen. Stel nu dat x_t een waarde heeft van 11,5. Dan zullen de containers die aankomen tijdens periode t opgeslagen worden in 11 lege baaien en de helft van een andere baai. Er wordt verondersteld dat als een baai deels toegewezen wordt, het aantal verplaatsingen proportioneel is naargelang de grootte van het toegewezen deel. De term $\{1 + 1/(4r)\}(A^2/4r)$ is bij de huidige veronderstelling van een constant aankomsttempo constant en kan dus vereenvoudigd weergeven worden bij het formuleren van de doelfunctie als de term B. De doelfunctie wordt dan:

$$Min \sum_{t=1}^{N} \frac{B}{x_t}$$

met als beperkingen:

$$\sum_{t=1}^{N} x_t \leq b$$

$$x_t \geq 0 \quad voor\ alle\ t$$

De optimale oplossing voor dit probleem wordt gevonden als $x_t^* = b / N$. Aldus wordt $h_t^* = \frac{A}{rx_t^*} = \frac{NA}{rb}$ de optimale hoogte van de stapels. (Kim en Kim, 1999)

Wordt het aankomsttempo van de importcontainers nu cyclisch, dan moeten er enkele aanpassingen in het model gemaakt worden. Met een cyclisch aankomsttempo wordt bedoeld dat het aantal aankomsten kan verschillen van bijvoorbeeld dag tot dag maar dat hetzelfde patroon herhaald wordt over een bepaalde periode. In volgend voorbeeld wordt een week genomen als tijdsduur van één patroon. Verondersteld wordt dus dat het bijvoorbeeld mogelijk is dat op een willekeurige donderdag niet hetzelfde aantal containers afgeladen wordt als op een andere weekdag, maar wel evenveel als op elke andere donderdag. Met andere woorden is $A_t = A_{t+7}$. Als nu $B_t = \{1 + 1/(4r)\}(A_t^2/4r)$, dan

geldt ook dat $B_t = B_{t+7}$. En ook voor x_t kan de lijn doorgetrokken worden naar $x_t = x_{t+7}$. Het probleem wordt dan als volgt:

$$Min \sum_{t=1}^{7} \frac{B_t}{x_t} \qquad \text{(P)}$$

en is onderhevig aan volgende beperkingen:

$$\sum_{i=t-n}^{t-1} x_{\alpha(i)} \le b, \quad t = 1, 2, ..., 7,$$
$$x_t \ge A_t / (\bar{h}r), \quad t = 1, 2, ..., 7,$$

met α(i) = 1 + de rest van i/7 en $\bar{h}$ de maximale hoogte van de stapels. Deze laatste wordt bepaald door de hoogte van de transferkraan die zich boven de containers beweegt. De eerste beperking bepaalt dat de totale beschikbare ruimte beperkt is. Stel nu bijvoorbeeld dat n = 4, dan: $x_5 + x_6 + x_7 + x_1 \le b$ voor t = 1, $x_6 + x_7 + x_1 + x_2 \le b$ voor t = 2, enzovoort. De tweede beperking begrenst de maximale hoogte op $\bar{h}$. (Kim en Kim, 1999)

Om dit probleem tot een goed einde te brengen gaat men hier gebruik maken van *sub-gradient optimization*. Deze procedure werd eerder in deze tekst al toegepast in II.5.1 voor het model dat in 2003 door Kim en Park gepubliceerd werd.

In deze toepassing is het de eerste beperking die gaat gerelaxeerd worden tot:

$$Minimize \sum_{t=1}^{7} \frac{B_t}{x_t} + \sum_{t=1}^{7} \lambda_t \left(\sum_{i=t-n}^{t-1} x_{\alpha(i)} - b \right) \qquad \text{(R)}$$

De oplossing voor (P) kan dus gevonden worden door het oplossen van volgend duaal probleem (D) en zeven oorspronkelijke problemen (DP_t). De tweede beperking zal bij deze uitwerking blijven gelden.

$$\underset{\lambda_t}{Maximize} \sum_{t=1}^{7} \left\{ \lambda_t \left(\sum_{i=t-n}^{t-1} x_{\alpha(i)} - b \right) + \frac{B_t}{x_t} \right\} \qquad \text{(D)}$$

$$\underset{x_t}{Minimize} \frac{B_t}{x_t} + x_t \sum_{i=t-1}^{t+n-2} \lambda_{\alpha(i)} \qquad (DP_t)$$

Hier moet echter nog opgemerkt worden dat voor n = 4 en t = 1, wordt:

$$\sum_{i=t-1}^{t+n-2} \lambda_{\alpha(i)} = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4$$

Dit is overeenkomstig met de beperkingen geassocieerd met x_1. (Kim en Kim, 1999)

De optimale oplossing van (DP_t) voor een gegeven set waarden voor $\lambda_{\alpha(i)}$ kan als volgt gevonden worden (Kim en Kim, 1999):

$$x_t^* = \begin{cases} \sqrt{\dfrac{B_t}{\sum_{i=t-1}^{t+n-2} \lambda_{\alpha(i)}}} & als \quad \sqrt{\dfrac{B_t}{\sum_{i=t-1}^{t+n-2} \lambda_{\alpha(i)}}} \geq A_t / (\bar{hr}) \\ A_t / (\bar{hr}) & anders \end{cases}$$

De procedure van *sub-gradient optimization* die hier toegepast werd, verliep in de vijf stappen zoals beschreven in “Segregating space allocation models for container inventories in port container terminals” (1999). In deze tekst zullen deze vijf stappen niet overgenomen worden.

Tenslotte bespreken Kim en Kim het derde en laatste geval. Namelijk indien het aankomsttempo van de importcontainers dynamisch is. Het aankomsttempo van de containers varieert nu zonder regelmaat. Er zal nu toegewezen worden op basis van een verschuivende planningshorizon. In periode 1 wordt de plaatstoewijzing gepland voor de perioden 1 tot N, welke de lengte van de planningshorizon is. Op basis van de uitkomsten hiervan worden de beschikbare plaatsen voor periode 1 toegewezen. Deze lijn wordt vervolgens doorgetrokken naar de volgende periodes. In periode twee wordt een planning opgemaakt voor periode 2 tot en met periode (N+1). Deze planning wordt nadien enkel toegepast voor de periode 2. Op deze wijze wordt het proces herhaald voor elke periode. Op het moment van elke periode kan het plaatstoewijzingsprobleem als volgt geformuleerd worden:

$$Minimize \sum_{t=1}^{N} \frac{B_t}{x_t},$$

$$waarbij\ B_t = (1+1/(4r))A_t^2/(4r),$$

$$onderhevig\ aan \sum_{i=t-n+1}^{t} x_i \le b, \quad t = 1,..., N-1,$$

$$\sum_{i=N-n+1}^{N} x_i \le b - S_N,$$

$$x_t \ge A_t/(\bar{h}r), \quad t = 1, 2, ..., N$$

Wanneer S_N de waarde is voor de grensbeperking in de laatste stage en x_t ($t \le 0$) een op voorhand vastgelegde waarde heeft die correspondeert met de in de vorige periode toegewezen hoeveelheid ruimte, kan het hierboven gestelde probleem omgezet worden in twee subproblemen.

$$\underset{\lambda_i}{Maximize}\left[\sum_{t=1}^{N-1}\left\{\lambda_t\left(\sum_{i=t-n+1}^{t} x_i - b\right) + \frac{B_t}{x_t}\right\} + \left\{\lambda_N\left(\sum_{i=N-n+1}^{N} x_i - b + S_N\right) + \frac{B_N}{x_N}\right\}\right] \quad \text{(D)}$$

$$\underset{x_t}{Minimize}\ \frac{B_t}{x_t} + x_t \sum_{i=t}^{t+n-1} \lambda_i \quad \text{(DP}_t\text{)}$$

onderhevig aan: $x_t \ge A_t/(\bar{h}r), \quad t = 1,..., N,$

met $\lambda_{N+1}, ..., \lambda_{N+n-1} = 0$

Deze kunnen vervolgens door middel van *sub-gradient optimization* opgelost worden. Dit gebeurt volgens dezelfde procedure als in het tweede geval met een cyclisch aankomsttempo.

$$x_t^* = \begin{cases} \sqrt{\dfrac{B_t}{\sum_{i=t}^{t+n-1} \lambda_i}} & als \quad \sqrt{\dfrac{B_t}{\sum_{i=t}^{t+n-1} \lambda_i}} \ge A_t/(\bar{h}r) \\ A_t/(\bar{h}r) & anders \end{cases}$$

De hierboven vermelde waarde voor x_t is de optimale oplossing voor (DP_t). (Kim en Kim, 1999)

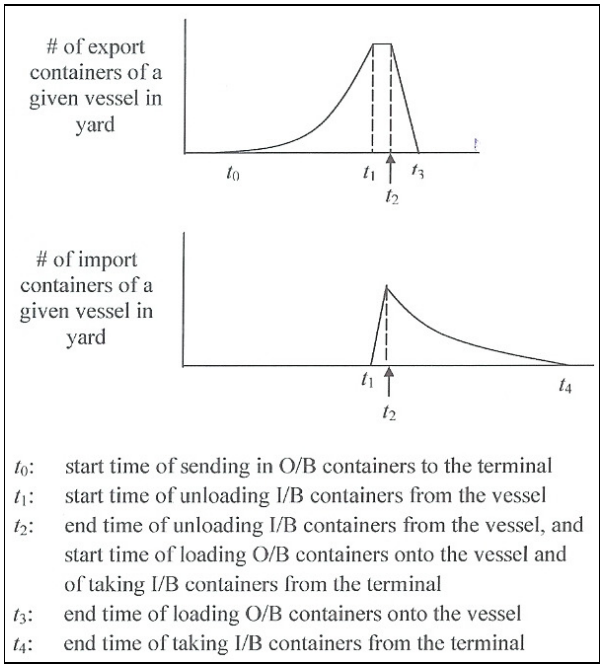
II.5.3 Modellen vertrekkende vanuit import- en exportcontainers

Zhang, Liu, Wan, Murty en Linn bestudeerden in "Storage space allocation in container terminals" (2003) het probleem van plaatstoewijzing in een complexe terminal. Dit wil zeggen dat zij een containerterminal beschouwden met zowel inkomende als uitgaande laadkisten. Zij maken hiervoor gebruik van een *rolling-horizon* aanpak waarbij in elke planningsperiode het probleem opgesplitst wordt in twee stappen en mathematische modellen. In de eerste stap wordt de werkbelasting tussen de verschillende *blocks* gebalanceerd. Het totaal aantal containers geassocieerd met elk schip en toegewezen aan elke *block* is het resultaat van de tweede stap. Deze minimaliseert de totale afstand die afgelegd moet worden om de containers te transporteren tussen de schepen en de *blocks*. (Steenken *et al.*, 2004)

Het doel van dit model is dus eenvoudigweg het beslissen in welke *blocks* van het opslagterrein de VSDS en de CYGD containers van elk schip geplaatst worden. Een VSDS of *vessel discharge* container is een inkomende of doorvoer container op een schip voordat deze afgeladen en op de *yard* gealloceerd wordt. Een uitgaande container, voordat hij de opslagplaats wordt binnen gebracht en opgeslagen wordt, noemt men een *container yard grounding* of CYGD container. Nadat deze laadkisten op het opslagterrein geplaatst zijn, worden de VSDS containers CYPI of *container yard pickup* containers. Een CYPI containers is een laadkist die op de opslagplaats wacht om afgehaald te worden door de klanten. De doorvoer VSDS containers en CYGD containers worden na het plaatsen *vessel loading* of VSLD containers. Deze zijn uitgaande of transitcontainers die zich op het terrein bevinden en wachten om geladen te worden op een schip. (Zhang *et al.*, 2003)

Een voorafgaande vereiste is het bepalen van de verwachte werkbelasting voor elke tijdsperiode. Figuur 16 toont het typische patroon van de variatie doorheen de tijd in het aantal inkomende en uitgaande containers in een containeropslagplaats. De duurtijd van het laden en lossen, (t_1,t_3), kan verondersteld worden gekend te zijn voor elk schip. De

patronen in de tijdsintervallen (t_0,t_1) en (t_2,t_4) worden bekomen door het optellen van het aantal aanwezige containers in de yard. Deze patronen zijn willekeurig voor de verschillende schepen, maar worden over de tijd wel stabiel. We veronderstellen dan ook deze stabiele patronen voor elke scheepslading.



Figuur 16: Het aantal inkomende en uigaande containers per schip in functie van de tijd (Zhang *et al.*, 2003)

Uit deze gegevens per container kunnen we de totale werkdruk vereiste bepalen. Op basis hiervan kan men de vraag naar kaaikranen, *internal trucks*, *external trucks*, enzovoort schatten. Het probleem blijft echter te complex om in zijn huidige staat op te lossen. Vandaar dat het probleem in twee niveaus zal opgesplitst worden zoals reeds in de eerste paragraaf van deze subtitel besproken werd. (Zhang *et al.*, 2003)

Omdat binnen een containerterminal 24 uur op 24 gewerkt wordt, maakten Zhang *et al.* gebruik van de *rolling-horizon* techniek. Er wordt een vaste periode in de onmiddellijke toekomst gepland en dit plan wordt uitgevoerd tot het volgende planningstijdstip. Op dat

moment wordt een nieuw plan gemaakt en uitgevoerd op basis van de huidige informatie. Dit proces herhaalt zich dan steeds verder in de toekomst. (Zhang *et al.*, 2003)

Eerst worden dus het totaal aantal VSDS en CYGD containers per *block* per planningsperiode bepaald. Hierbij wordt verondersteld dat alle containers van dezelfde grootte zijn en de nodige middelen hiervoor aanwezig zijn. Containers van verschillende grootte zouden echter geen problemen veroorzaken omdat deze in normale omstandigheden toch niet gemengd worden binnen éénzelfde *block*. (Zhang *et al.*, 2003)

Bij het begin van een planningshorizon is volgende informatie gekend:

B het totaal aantal *blocks* in de terminal

T het totaal aantal planningsperiodes binnen een planningshorizont; T=18

C_i de opslagcapaciteit van *block* i; $1 \le i \le B$

V_{i0} het aantal containers in *block* i bij het begin van de planningshorizon; $1 \le i \le B$

P_{it}^0 het verwachte aantal initiële CYPI containers opgeslagen in *block* i die opgepikt moeten worden in periode t; $1 \le i \le B$; $1 \le t \le T$

L_{it}^0 het verwachte aantal initiële VSLD containers opgeslagen in *block* i die geladen moeten worden op een schip tijdens periode t; $1 \le i \le B$; $1 \le t \le T$

$\tilde{G}_{tk}$ het verwachte aantal CYGD containers die in de containerterminal arriveren in periode t en moeten geladen worden op een schip in periode t + k; $1 \le t \le T$; $0 \le k \le T - t$

$\tilde{D}_{tk}$ het verwachte aantal inkomende VSDS containers die afgeladen worden in periode t en opgepikt worden door klanten in periode t + k; $1 \le t \le T$; $0 \le k \le T - t$

$\tilde{R}_{tk}$ het verwachte aantal transitcontainers die van schepen geladen worden in periode t en geladen worden op een ander schip op t + k; $1 \le t \le T$; $0 \le k \le T - t$

α_{it} het verwachte aantal CYGD containers die in de containerterminal aankomen in periode t, gealloceerd worden aan *block* i en geladen worden op een schip in

een periode die verder verwijderd is dan het bereik van de huidige planningshorizont; $1 \le t \le T ; 1 \le i \le B$

β_{it} het verwachte aantal inkomend VSDS containers afgeladen van een schip in periode t, gealloceerd aan *block* i en met een ongekende afhaaltijdstip of een afhaaltijdstip buiten de huidige planningshorizont; $1 \le t \le T ; 1 \le i \le B$

γ_{it} het verwachte aantal transitcontainers afgeladen van een schip in periode t, gealloceerd aan *block* i en met een ongekende laadtijdstip of een laadtijdstip buiten de huidige planningshorizont; $1 \le t \le T ; 1 \le i \le B$

De cijfers voor de laatste zes variabelen kunnen gehaald worden uit de eerder besproken patronen en uit de werkschema's van de schepen. De beslissingsvariabelen van het model van Zhang *et al.* zijn:

G_{itk} het aantal CYGD containers met volledige informatie opgeslagen in *block* i zodat ze aankomen in de terminal in periode t en op een schip geladen worden in periode t + k; $1 \le t \le T ; 1 \le i \le B ; 0 \le k \le T - t$

G_{it} het totaal aantal CYGD containers opgeslagen in *block* i die aankomen in de terminal in periode t; $1 \le t \le T ; 1 \le i \le B$

D_{itk} het aantal inkomende VSDS containers met volledige informatie opgeslagen in *block* i die afgeladen zijn van een schip in periode t en opgepikt worden in periode t + k; $1 \le t \le T ; 1 \le i \le B ; 0 \le k \le T - t$

D_{it} het totaal aantal VSDS containers (inkomend en transit) opgeslagen in *block* i die afgeladen worden in periode t; $1 \le t \le T ; 1 \le i \le B$

R_{itk} het aantal transitcontainers met volledige informatie opgeslagen in *block* i die afgeladen zijn van een schip in periode t en geladen moeten worden op een ander schip in periode t + k; $1 \le t \le T ; 1 \le i \le B ; 0 \le k \le T - t$

L_{it} het totaal aantal VSLD containers (uitgaand en transit) opgeslagen in *block* i die op een schip geladen worden in periode t, $1 \le t \le T ; 1 \le i \le B$

P_{it} het totaal aantal CYPI containers opgeslagen in *block* i die opgepikt worden door klanten in periode t; $1 \le t \le T ; 1 \le i \le B$

V_{it} de inventaris van *block* i op het einde van periode t; $1 \le t \le T$;$1 \le i \le B$

(Zhang *et al.*, 2003)

Zhang *et al.* wilden nu de scheepsgerelateerde containers (laden en afladen) en het aantal containers over de verschillende *blocks* balanceren. Deze doelstellingen vertaalden zich in de volgende doelfunctie:

$$Minimize \sum_{t=1}^{T} \left\{ w_1 \left[\max_{\{i\}} \left(D_{it} + L_{it} \right) - \min_{\{i\}} \left(D_{it} + L_{it} \right) \right] + w_2 \left[\max_{\{i\}} \left(D_{it} + L_{it} + G_{it} + P_{it} \right) - \min_{\{i\}} \left(D_{it} + L_{it} + G_{it} + P_{it} \right) \right] \right\}$$

In deze functie stelt $\left(D_{it} + L_{it} \right)$ het verwachte aantal scheepsgerelateerde containers voor die behandeld moeten worden in periode t en $\left(D_{it} + L_{it} + G_{it} + P_{it} \right)$ het verwachte totaal aantal containers die in periode t behandeld moeten worden. w_1 en w_2 zijn de gewichten van de twee termen van de functie en worden aangepast naargelang het relatieve belang van de scheepsgerelateerde containers binnen een totaal aantal containers zoals deze door de terminal geïnterpreteerd worden. (Zhang *et al.*, 2003)

Om de praktische toepassingen van dit model veilig te stellen, moeten er eveneens beperkingen aangebracht worden. Deze beperkingen kunnen in 3 groepen ingedeeld worden: beperkingen ter handhaving van de containerstroom, beperkingen op de CYPI en VSLD containers en beperkingen op de dichtheid van de *block*. Daarnaast is er nog de beperking die stipuleert dat alle variabelen niet-negatieve integere waarden aannemen. (Zhang *et al.*, 2003)

De beperking in verband met de handhaving van de containerstroom dient vooreerst om te verzekeren dat het verwachte totaal aantal inkomende VSDS containers met volledige informatie en wachten op hun allocatie ($\tilde{D}_{tk}$), de som is van al de containers van dit type toegekend aan alle *blocks*. De tweede en derde beperking in deze categorie hebben een gelijkaardige betekenis maar dan voor CYGD en transit VSDS containers. De vierde van deze beperkingen zorgt ervoor dat het verwachte totaal aantal VSDS containers gealloceerd aan *block* i tijdens periode t, D_{it} , de som is van het totaal aantal VSDS

containers (inkomend en transit) met volledige informatie en van die containers met een ongekend vertrektijdstip ($\beta_{it}+\gamma_{it}$). Deze beperking voor CYGD containers wordt uitgedrukt in de laatste regel van volgende beperkingen (Zhang *et al.*, 2003):

$$\tilde{D}_{tk}=\sum_{i=1}^{B}D_{itk},\quad t=1,2,...,T;\ k=0,1,...,T-t$$

$$\tilde{G}_{tk}=\sum_{i=1}^{B}G_{itk},\quad t=1,2,...,T;\ k=0,1,...,T-t$$

$$\tilde{R}_{tk}=\sum_{i=1}^{B}R_{itk},\quad t=1,2,...,T;\ k=0,1,...,T-t$$

$$D_{it}=\sum_{k=0}^{T-t}(D_{itk}+R_{itk})+\beta_{it}+\gamma_{it},\quad i=1,2,...,B;\ t=1,2,...,T$$

$$G_{it}=\sum_{k=0}^{T-t}G_{itk}+\alpha_{it},\quad i=1,2,...,B;\ t=1,2,...,T$$

De eerste van de volgende beperkingen stelt dat het aantal VSLD containers behandeld in *block* i tijdens periode t, L_{it}, bestaat uit twee delen. Het eerste deel zijn de VSLD containers die bij het begin in *block* i opgeslagen waren en op schepen geladen moeten worden tijdens periode t in de huidige planningshorizont, L_{it}^{0}. Het tweede deel zijn de containers getransfereerd van de overeenkomstige CYGD en transitcontainers die aankomen tijdens de planningshorizont. De tweede beperking heeft de zelfde functie als de vorige maar in het geval van de CYPI containers. (Zhang *et al.*, 2003)

$$L_{it}=L_{it}^{0}+\sum_{k=0}^{t-1}\left(G_{i(t-k)k}+R_{i(t-k)k}\right),\quad i=1,2,...,B;\ t=1,2,...,T$$

$$P_{it}=P_{it}^{0}+\sum_{k=0}^{t-1}D_{i(t-k)k},\quad i=1,2,...,B;\ t=1,2,...,T$$

Het derde type beperking in dit model zijn de beperkingen in verband met de dichtheid van de *block*. De eerste van deze twee beperkingen zorgt ervoor dat de inventaris (V_{it}) periode per periode geüpdate wordt. Het voorkomen dat de inventaris van elke *block* de maximale dichtheid overschrijdt gebeurt door de tweede beperking in deze categorie.

$$V_{it} = V_{i(t-1)} + \left[\left(G_{it} + D_{it} \right) - \left(P_{it} + L_{it} \right) \right], \quad i = 1, 2, ..., B;\ t = 1, 2, ..., T$$
$$V_{it} \le \eta C_i, \quad i = 1, 2, ..., B;\ t : 1, 2, ..., T$$

met η de maximaal toegelaten dichtheid binnen elke *block*. (Zhang *et al.*, 2003)

Omdat het hierboven besproken model niet-lineair is omwille van zijn doelfunctie wordt het volgende gedefinieerd:

$$A_t = \max_{\{i\}}(D_{it} + L_{it}), \quad B_t = \min_{\{i\}}(D_{it} + L_{it}),$$
$$M_t = \max_{\{i\}}(D_{it} + L_{it} + G_{it} + P_{it}), \quad N_t = \min_{\{i\}}(D_{it} + L_{it} + G_{it} + P_{it})$$

Als deze gebruikt worden in de doelfunctie, wordt het volgende lineair integer programmeringsprobleem bekomen:

$$Minimize \sum_{t=1}^{T} \left[w_1 (A_t - B_t) + w_2 (M_t - N_t) \right]$$

Dit is buiten de eerder genoemde beperkingen ook onderhevig aan enkele bijkomende restricties:

$$D_{it} + L_{it} \le A_t, \quad i = 1, 2, ..., B;\ t = 1, 2, ..., T$$
$$D_{it} + L_{it} \ge B_t, \quad i = 1, 2, ..., B;\ t = 1, 2, ..., T$$
$$G_{it} + D_{it} + L_{it} + P_{it} \le M_t, \quad i = 1, 2, ..., B;\ t = 1, 2, ..., T$$
$$G_{it} + D_{it} + L_{it} + P_{it} \ge N_t, \quad i = 1, 2, ..., B;\ t = 1, 2, ..., T$$

(Zhang *et al.*, 2003)

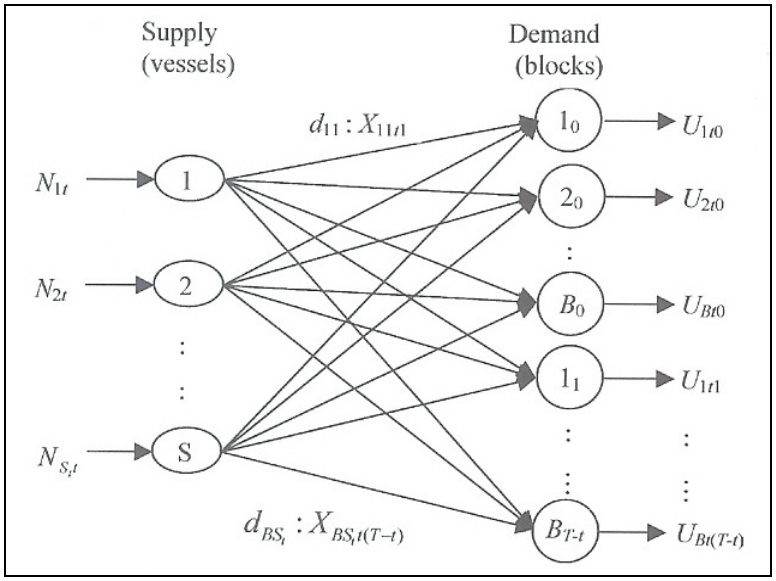
Als dit lineair probleem opgelost is, kan aan het tweede niveau begonnen worden. Dit bepaalt het aantal containers geassocieerd met elk schip, die bijdragen tot het totaal aantal containers in elke *block*. Het doel hierbij is het minimaliseren van de totale container verplaatsingskosten. De meting hiervan gebeurt door het optellen van de totaal afgelegde afstand door interne trucks tussen de schepen en de opslagblokken.

Op het tweede niveau zullen VSDS en CYGD containers gelijk behandeld worden en de parameters en beslissingsvariabelen voor dit probleem zijn weergegeven in volgende tabel.

	Betekenis voor VSDS containers	Betekenis voor CYGD Containers
d_{ij}	De afstand tussen *block* i en de ligplaats of schip j	De afstand tussen *block* i en de ligplaats of schip j
N_{jt}	Het totaal aantal VSDS containers afgeladen van schip j in periode t	Het totaal aantal CYGD containers aangekomen in periode t die op schip j geladen moeten worden
S_t	Het totaal aantal schepen aangemeerd in de terminal in periode t die dienen afgeladen te worden	Het totaal aantal schepen waarop de aankomende CYGD containers, die in periode t arriveren, geladen moeten worden
U_{itk}	$D_{itk} + R_{itk}$	G_{itk}
X_{ijtk}	Het aantal VSDS containers afgeladen van schip j in periode t die moeten worden opgepikt door klanten of geladen op een ander schip in periode t + k en opgeslagen zullen worden in *block* i	Het aantal CYGD containers aangekomen in de terminal in periode t die geladen moeten worden op schip j, in periode t + k en opgeslagen zullen worden in *block* i

Figuur 17: Notaties van de parameters en beslissingsvariabelen voor het probleem op niveau twee (Zhang et al., 2003)

Het probleem op niveau twee kan nu periode per periode opgelost worden aan de hand van hetzelfde model. Dit is een transportmodel zoals voorgesteld in figuur 18.



Figuur 18: Voorstelling van het probleem op het tweede niveau (Zhang et al., 2003)

Mathematisch ziet dit transportprobleem er als volgt uit:

$$Minimize \sum_{i=1}^{B} \sum_{j=1}^{S_t} \sum_{k=0}^{T-t} d_{ij} X_{ijtk}$$

$$\sum_{j=1}^{S_t} X_{ijtk} = U_{itk}, \quad i = 1,2,...,B;\ k = 0,1,...,T-t$$

$$\sum_{i=1}^{B} \sum_{k=0}^{T-t} X_{ijtk} = N_{jt}, \quad j = 1,2,...,S_t$$

$$X_{ijtk} \geq 0, \quad i = 1,2,...,B;\ k = 0,1,...,T-t;\ j = 1,2,...,S_t$$

De eerste van de drie beperkingen stelt de capaciteitsbeperking voor elke *block* of *demand node* voor. De tweede vertegenwoordigt de beperking op de hoeveelheid die afgeladen kan worden van een schip voor elke *supply node*. De laatste beperking zorgt ervoor dat X_{ijtk} geen negatieve waarden kan aannemen. Het transportmodel garandeert ook een integere oplossing voor X_{ijtk} omdat U_{itk} en N_{jt} integer zijn. (Zhang *et al.*, 2003)

Verder geven Zhang, Liu, Wan, Murty en Linn nog een numerische studie van dit model, maar deze zal hier niet verder overgenomen worden.

Hoofdstuk III: Praktijkstudie met als leidraad "An approach to determine storage locations of containers at seaport terminals"

III.1 "An approach to determine storage locations of containers at seaport terminals" van Preston en Kozan (1999)

III.1.1 Verantwoording van de keuze

Na voldoende artikels doorgenomen te hebben tijdens de literatuurstudie, werd "An approach to determine storage locations of containers at seaport terminals" (1999) van Preston en Kozan gekozen om verder onder de loep te nemen en aan de hand van deze publicatie bijkomende lessen te trekken over de plaatstoewijzing van containers.

Het artikel sprak mij aan omwille van de link tussen het operationeel onderzoeksmodel en informatica. In de huidige samenleving is de hulp van hard- en software toepassingen niet meer weg te denken. Dit geldt eveneens voor het vakdomein *Operations Research*. Zoals in dit artikel het geval is, kunnen softwareprogramma's modellen oplossen die anders een eeuwigheid aan berekeningstijd zouden vragen.

Daarnaast was het model in dit artikel aantrekkelijk omwille van zijn twee mogelijke gebruikstoepassingen, voorgesteld door Preston en Kozan. Het model zou naast de plaatstoewijzing van containers eveneens gebruikt kunnen worden als hulpmiddel bij de beslissingen over toekomstige investeringen in de containerterminal.

Tenslotte was dit artikel een mooie uitdaging vermits ik nog nooit eerder van genetische algoritmen gehoord had.

III.1.2 Bespreking van het gekozen artikel

In "An approach to determine storage locations of containers at seaport terminals" (1999) beschrijven P. Preston en E. Kozan hun model met het objectief een optimale opslagstrategie te bepalen, die de noodzakelijke tijd minimaliseert die nodig is om de containers van de opslagplaats naar de laadplaats binnen de perimeter van het schip te verplaatsen. Ze gaan ervan uit dat als deze tijd geminimaliseerd wordt, ook de *turn-around time* van het containerschip zal verkleinen. Dit wil zeggen dat het schip niet zo lang aangemeerd moet liggen. Preston en Kozan willen dit doel bereiken door de ontwikkeling van een *container location model* (CLM) en dit model op te lossen door middel van genetische algoritmen. Genetische algoritmen zijn een vorm van metaheuristieken, met als onderliggend principe *survival of the fittest*, die kunnen gebruikt worden voor het oplossen van integere niet-lineaire programmeringsproblemen. (Hillier en Lieberman, 2001)

Het doel van het model van Preston en Kozan is de minimalisatie van de aanmeertijd van een schip in de haven door het minimaliseren van de volgende doelfunctie:

$$Minimise \underset{Mac}{Max} \sum_{\{i|mac_i=mac\}} (travelling\ time_i + set\text{-}up\ time_i)$$

Deze formule bestaat uit twee grote componenten:

- travelling time$_i$: de benodigde tijd om container I te transporteren tussen de opslagplaats en de laadplaats (*marshalling area*)

$$travelling\ time_i = lock + \frac{x_i RW + y_i CW}{V^m} + lock, \text{ waar m = mac}_i$$

met:

lock: de tijd die de machines van de opslagplaats nodig hebben om een container vast te maken voor transport. Deze wordt verondersteld gelijk te zijn aan de tijd die nodig is om een container los te maken.

CW: breedte van een kolom in de opslagplaats

RW: breedte van een rij in de opslagplaats

X_i: de rij waar container i is opgeslagen

Y_i: de kolom waar container i is opgeslagen

V^m: de snelheid van de behandelingsmachine m

mac_i: de machine waarmee de behandeling van container i gepland is

set-up $time_i$: de benodigde tijd om toegang te krijgen tot de gewenste container i

$$\textit{set-up time}_i = 0 \quad \text{als } z_{i,t} = 0$$

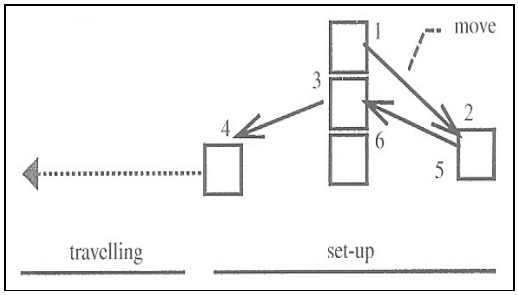
$$= z_{i,t}\,(4\textit{lock} + \textit{move}) + 2\textit{lock} + \textit{move} \quad \text{als } z_{i,t} \neq 0$$

met:

move: de tijd die nodig is om een container te verplaatsen naar een aangrenzende positie zodat men bij onderliggende containers kan komen.

$Z_{i,t}$: de verticale opslagpositie van container i op tijdstip t (= $time_i$); dit wordt gemeten door het aantal containers die bovenop container i zijn opgeslagen en zodoende de toegang van de behandelingsmaterialen tot container i vertragen.

Figuur 19 toont de benodigde behandelingen in het geval dat de gewenste container zich op het tweede niveau bevindt van een stapel van 3 containers. Eerst moet de bovenste laadkist verplaatst worden (1→2). Deze container zal later teruggeplaatst moeten worden (5→6). Vervolgens wordt de gewenste container behandeld (3→4).



Figuur 19: De behandeling van een container op het 2de van 3 niveaus (Preston en Kozan, 1999)

Dit model is echter ook onderhevig aan enkele beperkingen. Een eerste fysische beperking bestaat erin dat twee containers zich niet op hetzelfde moment op dezelfde plaats kunnen bevinden. Deze beperking wordt weergegeven in volgende formule:

Als $x_i = x_{i'}$ en $y_i = y_{i'}$ dan $z_{i,t} \neq z_{i',t}$ $\quad \forall i \neq i'$

Eveneens bestaat er een machinebeperking. Deze beperking weerspiegelt het feit dat een behandelingsmachine maar één container tegelijk kan oppikken en verplaatsen.

Als $mac_i = mac_{i'}$ dan $time_i \neq time_{i'}$ $\quad \forall i \neq i'$

Een derde beperking zorgt ervoor dat de parameter $z_{i',t}$ aangepast wordt wanneer container i, die opgeslagen is boven i´, verplaatst wordt voor container i´.

Als $x_i = x_{i'}$ en $y_i = y_{i'}$ en $z_{i,t} < z_{i',t}$ dan $z_{i',t'} = z_{i',t} - 1$ $\quad \forall i \neq i'$

De laatste beperking die men aan het model toevoegt, bepaalt dat alle containers die op een bepaald schip moeten geladen worden, aan boord gebracht worden tussen de aankomsttijd en de vertrektijd van het schip.

$$arrive^s + \sum_{\{i|ship_i = s\}} (travelling\ time_i + set\text{-}up\ time_i) \leq depart^s \qquad \forall s$$

met: $ship_i$ het schip waarop container i moet vertrekken

$depart^s$ de tijdstip van vertrek van schip s

$arrive^s$ het tijdstip van aankomst van schip s

Hierboven werd het probleem als een *mixed integer linear programming problem* (de minimalisatie of maximalisatie van een lineaire functie onderhevig aan lineaire beperkingen) gedefinieerd en de berekeningscomplexiteit ervan zal toenemen naarmate

het aantal containers in het schema toeneemt. Dit maakt het model moeilijk te gebruiken in reële situaties. Daarom zal men gebruik moeten maken van heuristieken om het model op te lossen. Preston en Kozan kozen in “An approach to determine storage locations of containers at seaport terminals” (1999) voor genetische algoritmen als techniek en dit leverde relatief goede resultaten oplevert, zelfs met de simpelste genetische algoritmes.

Dit model werd door Preston & Kozan geprogrammeerd, getest en vervolgens concludeerden zij het volgende. De resultaten varieerden weinig naargelang men een FCFS, LCFS of *random* schema gebruikte. Dit wil zeggen dat het type van ladingssequentie bijna geen invloed heeft op de gemiddelde transfertijd in de simulaties. Het aantal gebruikte terminalmachines heeft daarentegen wel een invloed op die transfertijd. Als het aantal behandelingsmachines afneemt, zal de transfertijd exponentieel toenemen. Veranderingen in de *storage area utilisation* geven een lineaire verandering in de transfertijd binnen het interval van 10% tot 50% *storage area utilisation*. Dus vanaf dat 50% van de opslagplaats ingenomen is door containers, zal een toename van het aantal containers in de opslagplaats bijna geen negatief effect meer hebben op de transfertijd. (Preston en Kozan, 1999)

III.2 Inleiding tot genetische algoritmen

Zowel in het artikel van Preston en Kozan als in het softwareprogramma, dat het onderwerp vormt van deze praktijkstudie, wordt gebruik gemaakt van genetische algoritmen. Daarom is het aangewezen om beknopt te beschrijven wat genetische algoritmen zijn.

Genetische algoritmen zijn zoekalgoritmen gebaseerd op de werking van natuurlijke selectie. In tegenstelling tot de meeste zoekalgoritmen opereren genetische algoritmen op een populatie van oplossingen in plaats van op één oplossing. Om gebruik te maken van genetische algoritmen moeten de oplossingen tot het probleem gecodeerd worden in een structuur die leesbaar is door een computer. Dit object wordt een chromosoom genoemd. Het genetische algoritme creëert een populatie van chromosomen en past vervolgens *crossovers* en mutaties toe op de *individuals* van de populatie om tot nieuwe *individuals* te komen. (Goldberg, 1989)

Aan *crossovers* en mutaties gaat meestal de stap "reproductie" vooraf. Dit is een proces waarbij individuele ketens of chromosomen gekopieerd worden naar een *mating pool* op basis van hun doelfunctiewaarden. Zo wordt bekomen dat de ketens met een betere doelfunctiewaarde een grotere kans hebben om bij te dragen tot de volgende generatie aan chromosomen. Het is immers op deze *mating pool* dat *crossover* toegepast wordt. De term *crossover* kan gedefinieerd worden als het combineren van twee *individuals* ('ouders') om twee *individuals* ('kinderen') bij te creëren. Op deze manier komt het 'genetisch materiaal' van de voorgaande generatie dus terug in de daaropvolgende generatie. Een *crossover* bestaat eigenlijk uit twee stappen. Eerst worden de ketens in de *mating pool* willekeurig gepaard. Vervolgens gaat elk paar van ketens kruisen. Een integere positie k wordt uniform en willekeurig genomen tussen 1 en de ketenlengte min één. Hierna worden alle karakters tussen de posities k+1 en het einde van de keten uitgewisseld, om zo 2 nieuwe ketens of chromosomen te bekomen. (Goldberg, 1989)

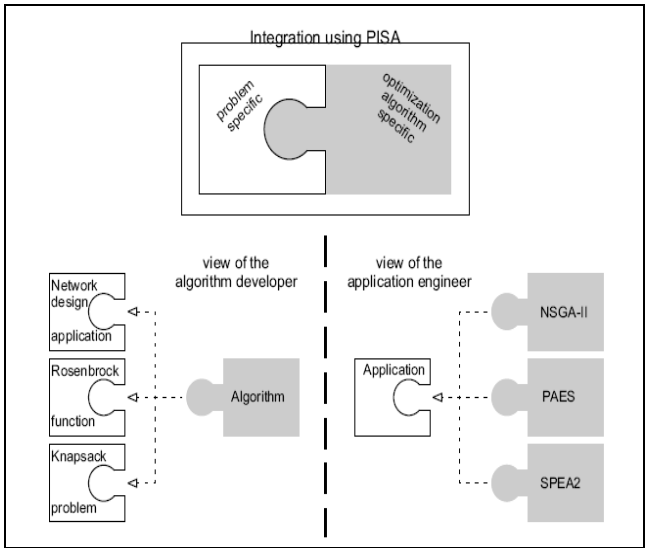
Mutaties zorgen voor een bepaalde hoeveelheid aan willekeur. Het zijn occasionele, willekeurige wijzigingen in de waarde op een positie in een keten. Het gebruik van mutaties kan bijdragen tot het vinden van oplossingen die *crossover* alleen niet zou kunnen bereiken. (Goldberg, 1989)

III.3 Implementatie van het algoritme

Voor de implementatie van het algoritme bracht mijn promotor mij in contact met de heer Jose Maria Pangilinan (docent aan de St. Louis University te Baguio op de Filippijnen en docotoraatsstudent als VLIR-bursaal aan de Universiteit Hasselt). De heer Pangilinan ontwikkelde het programma met behulp van de softwares PISA, SPEA2 en zijn kennis van genetische algoritmen. In de volgende punten worden deze componenten en de werking van de softwaretoepassingen verder uitgelegd.

III.3.1 PISA

PISA is een tekst-gebaseerde *interface* voor zoekalgoritmen die het optimalisatieproces opsplitst in twee componenten of modules. Enerzijds is er de algoritmische module, genaamd *selector*, en anderzijds de toepassings- of probleemspecifieke module *variator*. Deze twee modules worden als afzonderlijke programma's geïmplementeerd en communiceren onderling door middel van tekstbestanden. Het voordeel hiervan is dat deze data-uitwisseling eenvoudig te bekijken is door de gebruiker of programmeur. Het eigenlijke doel van PISA is om een gestandaardiseerd, uitbreidbaar en makkelijk te gebruiken raamwerk te bouwen voor de implementatie van algoritmen voor optimalisatie met meervoudige doelstellingen. (Bleuler *et al.*, 2003)

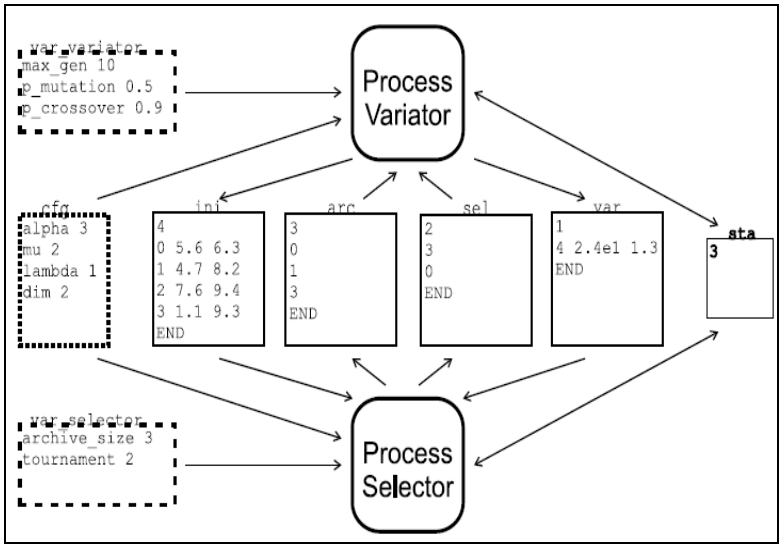


Figuur 20: Illustratie van het PISA-concept (Bleuler *et al.*, 2003)

Selector is de module die de delen van het optimalisatieproces bevat welke onafhankelijk zijn van het optimalisatieprobleem. (Computer Engineering and Networks Laboratory, 2006, online) In het geval van de toepassing die hier besproken wordt, bestaat dit algoritmische gedeelte uit SPEA2 waar de selectie en rangschikking gebeurt. Meer informatie over SPEA2 kan teruggevonden worden in het puntje III.3.2. De *variator* module bevat alle delen die specifiek zijn voor het optimalisatieprobleem zoals bijvoorbeeld de evaluatie van de oplossingen, de voorstelling van het probleem, enz.. (Computer Engineering and Networks Laboratory, 2006, online) Deze module bevat hier, in navolging van Preston en Kozan (1999), het genetische algoritme met de initiële populatie, *crossover*, mutatie en evaluatie van de doelfunctie.

Verschillende parameters zijn vereist om het gedrag van de verschillende modules te bepalen. In onderstaande figuur 21 zijn de drie uiterst linkse kaders met gestippelde randen de bestanden met de parameters noodzakelijk voor de werking van de PISA interface. Het cfg bestand specificeert de parameters die gemeenschappelijk zijn voor beide modules. De twee andere zijn modulespecifiek. De gemeenschappelijke parameters bestaan uit het aantal doelstellingen (dim) en de grootte van de drie verschillende

collecties van *individuals* die tussen de twee modules uitgewisseld worden. De betekenis van de drie overige parameters in het 'cfg' bestand wordt verder in deze tekst verklaard.



Figuur 21: De communicatie tussen de modules via tekstbestanden (Bleuler *et al.*, 2003)

Ook de tekstbestanden die gebruikt worden voor de data-uitwisseling zijn in bovenstaande figuur 21 terug te vinden en kunnen nu gespecificeerd worden naargelang de informatie die ze uitwisselen:

- PISA_ini initiële populatie
- PISA_sel *individuals* geselecteerd voor variatie ('ouders')
- PISA_var gevarieerde *individuals* (offspring)
- PISA_arc *individuals* in het *archive*

Een voordeel aan het gebruik van de PISA interface is dat door een maximale onafhankelijkheid tussen beide gedeelten slechts aandacht moet besteed worden aan het deel waarin de gebruiker geïnteresseerd is. Het andere gedeelte kan beschouwd worden als een gebruiksklare *black box*. Ten tweede hoeft bij het eigenlijke gebruik van een programma op basis van PISA, de gebruiker geen informatica achtergrond te bezitten. De behandeling van de *input* en *output* data en het instellen van de parameters is eenvoudig en makkelijk te begrijpen. (Bleuler *et al.*, 2003)

III.3.2 SPEA2 (*selector*)

SPEA2 of *Strength Pareto Evolutionary Algorithm 2* is een bouwblok voor de *selector* module binnen PISA. SPEA2 werkt met een populatie met een vaste grootte, waaruit veelbelovende kandidaten geselecteerd worden als 'ouders' voor de volgende generatie. De resulterende 'kinderen' nemen het vervolgens op tegen de leden van de oude populatie om opgenomen te worden in die populatie. (Zitzler *et al.,* 2001)

In het geval van SPEA2 wordt de betekenis van de gemeenschappelijke parameters binnen "PISA_cfg" als volgt ingevuld:

Alpha (α)	Populatiegrootte
Mu (μ)	Aantal parent individuals of het aantal 'ouders'
Lambda (λ)	Aantal offspring individuals of het aantal 'kinderen'
Dim	Het aantal doelstellingen

De twee locale parameters voor het *selector* deel SPEA2 zijn verschillend met het voorbeeld weergeven in figuur 21. De locale parameters bij SPEA2 heten 'seed' en 'tournament' . Als de waarde van 'seed' behouden blijft is het mogelijk om verschillende simulaties uit te voeren met dezelfde reeks toevalsgetallen (*random numbers).* 'Tournament' geeft het aantal *individuals* per paringsactie. De waarde voor 'tournament' blijft in dit geval constant op 2. Dit betekent dat twee 'ouders' worden gecombineerd om twee 'kinderen' te creëren. (Zitzler *et al.,* 2001)

III.3.3 De *variator* module

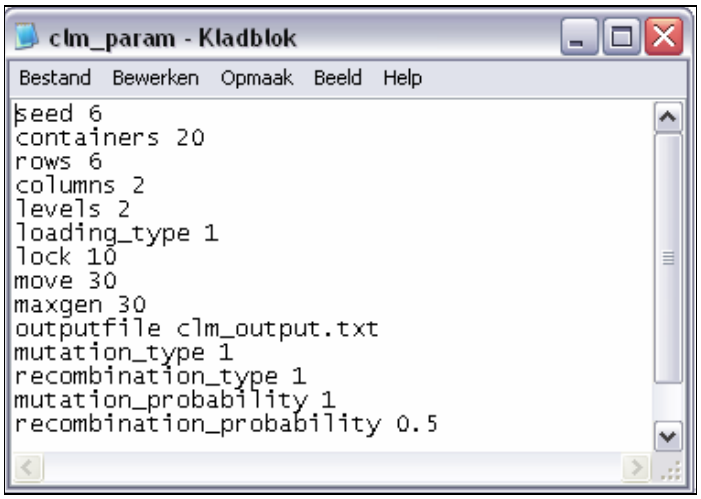
Deze module is in dit geval gebaseerd op het *Container Location Model* (CLM) uit "An approach to determine storage locations of containers at seaport terminals" (2001) van P. Preston en E. Kozan. Deze applicatiezijde van de PISA interface werd hier dan ook ingevuld door het genetische algoritme. Wel moet opgemerkt worden dat de functie voor

de *set-up time* licht werd aangepast ten opzichte van het artikel van Preston en Kozan en de bespreking van dit artikel in III.1. De functie die geprogrammeerd werd is de volgende:

$$\begin{aligned} \textit{set-up time}_i &= 0 && \text{als } z_{i,t} = 0 \\ &= z_{i,t}\,(4\textit{lock} + \mathbf{2}\textit{move}) + 2\textit{lock} + \textit{move} && \text{als } z_{i,t} \neq 0 \end{aligned}$$

Deze aanpassing gebeurde omdat een container die uit de weg gezet wordt en nadien teruggeplaatst wordt, volgens de definitie van de variabele *move*, twee *move*-bewegingen maakt in plaats van één.

De locale parameters voor de *variator* worden, voor het starten van de toepassing, ingegeven in het tekstdocument “clm_param”. Dit tekstdocument wordt weergegeven in Figuur 22. Niet enkel de parameters die het genetisch algoritme specificeren zijn in dit tekstdocument opgenomen. Het aantal containers die moeten geplaatst worden, kan hier ingegeven worden. Net als de parameters die de fysieke *layout* van de containerterminal bepalen (aantal *rows,* aantal *columns* en de maximale hoogte). Ook de tijdsduur van de eerder omschreven *lock* en *move* bewegingen zijn variabel gelaten. Daarnaast kan het beleid via “loading_type” nog gewijzigd worden. (1 = FCFS; 2 = LCFS) De parameters die het genetisch algoritme bepalen en die hier gewijzigd kunnen worden zijn: het aantal generaties die het genetisch algoritme doorloopt, de kans op een mutatie (mutation_probability) en de kans dat een gegeven paar *individuals* een *crossover* ondergaat (recombination_probability). Net al bij de locale parameters bepaalt ‘seed’ de reeks *random numbers* zodat herhaling op basis van dezelfde cijfers mogelijk is. ‘Mutation_type’ en ‘recombination_type’ zullen in de volgende simulaties steeds de waarde 1 behouden. Dit betekent dat er enkel 1-bit mutaties en 1-point crossover zullen op treden. Hierover vindt u meer in III.3.4.3.



Figuur 22: Het tekstdocument “clm_param” bedoeld voor het ingeven van parameters

III.3.4 De werking van het programma op basis van het CLM model

III.3.4.1 De chromosoomvoorstelling

Het is belangrijk om een goed begrip te hebben van de codering van de chromosomen. De chromosomen in het hier besproken genetisch algoritme bestaan uit een reeks van positieve integers inclusief 0. De lengte van zo een chromosoom is gelijk aan het aantal aankomende containers ingegeven in “clm_param” bij ‘containers’. Een integer vertegenwoordigt een 3D-ruimte ter grootte van een opslaglocatie voor 1 container (1 TEU). Er zullen immers enkel container met een grootte van 1 TEU in beschouwing genomen worden. Ter illustratie: de integer of positie 0 vertegenwoordigt de opslagplaats (0,0,0) ofwel rij 0, kolom 0 en niveau 0. Veronderstel nu een containeropslagplaats met 1 rij, 4 kolommen en 3 niveaus. Deze *layout* is goed voor 12 opslaglocaties (1x4x3) gaande van (0,0,0) tot (0,3,2). Gegeven het chromosoom in Figuur 23 zal container 2 of de container die als derde arriveert, toegewezen worden aan locatienummer 9 en container 5 aan opslaglocatie 1. Locatienummer 9 komt overeen met rij 0, kolom 3 en niveau 0 ofwel (0,3,0). Locatienummer 1 duidt op (0,0,1). Dit is weergegeven in Figuur 24.

0	1	2	3	4	5	6	7	8	9
6	0	9	10	7	1	3	4	8	5

Figuur 23: Een voorbeeld van een chromosoom (10 containers)

2	5	8	11	Tier 2
1	4	7	10	Tier 1
0	3	6	9	Tier 0
Kolom 0	Kolom 1	Kolom 2	Kolom 3	

Figuur 24: Zijaanzicht van een rij

III.3.4.2 De waarden voor enkele constanten

Om het model uit “An approach to determine storage locations of containers at seaport terminals” (1999) te kunnen gebruiken, was het noodzakelijk om in het software programma enkele waarden voor bepaalde constanten op te nemen in het deel van het genetisch algoritme. Voor de breedte van een kolom (CW) werd de waarde 2,4384m ingegeven. Dit is de breedte van de kortste zijde van een container. De lengte van de lange zijde van een container werd genomen voor de breedte van een rij. Deze lengte bedraagt 6,09625m

Voorts moest ook een waarde voor de snelheid van het behandelingsmateriaal gespecificeerd worden. Hiervoor werd de snelheid van 5,65 m/s ingegeven. Deze snelheid van 20 km/h voor een *straddle carrier* van de producent Kalmar werd gevonden op de website van Siemens, www.siemens.com. Voor de specifieke link naar de webpage wil ik u doorverwijzen naar de bibliografie. Op de website van de producent, Kalmar, was echter

geen notie van een gemiddelde snelheid van een *straddle carrier* terug te vinden. Het aantal *straddle carriers* ligt vast op 1. Die ene *straddle carrier* wordt dus toegekend aan een bepaalde ruimte, die ingegeven wordt in “clm_param”, om een aantal container hierin weg te plaatsen.

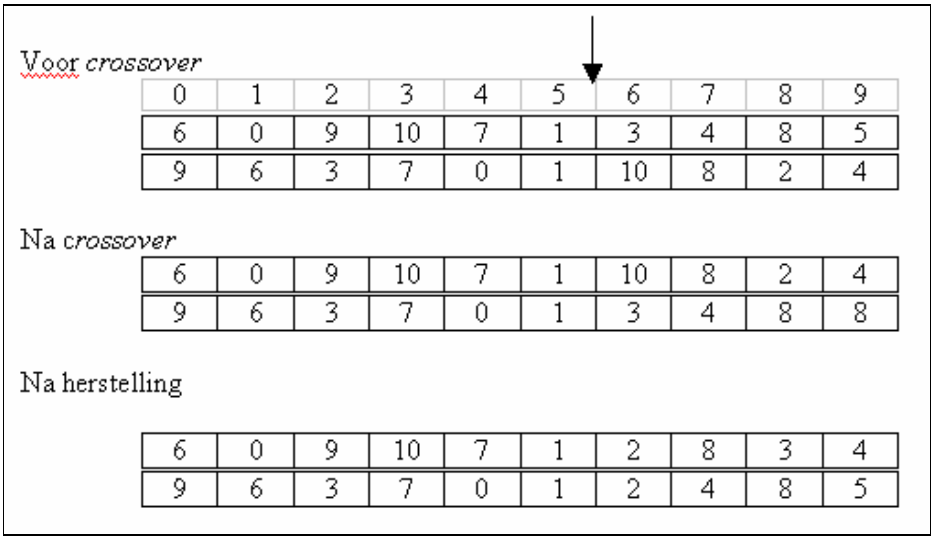
Ondanks het feit dat de parameters *lock* en *move* instelbaar zijn in de “clm_param” bestand, zijn deze twee termen constanten waarvan de waarde vast ligt. De reden waarom deze twee echter nog makkelijk te wijzigen zijn in “clm_param” is de moeilijkheid van de zoektocht naar deze waarden. Om een aanvaarbare tijd te bekomen voor elk van deze twee parameters contacteerde ik onder andere uitbaters van containerterminals zoals P&O en de Katoen natie, het Antwerpse havenbedrijf, het opleidingscentrum OCHA voor havenarbeiders en het Vlaams Instituut voor de Logistiek. De uitbaters wezen mij erop dat deze twee waarden, zoals omschreven in “An approach to determine storage locations of containers at seaport terminals”, niet gekend zijn omdat zij zelf geen interesse hebben in het kennen van deze tijdsintervallen. Dit vooral omdat deze tijd sterk afhankelijk is van verschillende factoren zoals de weersomstandigheden, de kwaliteiten van de bestuurder, enz.. Uiteindelijk vond ik iemand bereid om een schatting te maken. André Nollet bij de Hessenoordnatie zei mij telefonisch dat het geheel ongeveer 4 minuten zou duren. Vervolgens gaf ik de 60 seconden in voor de *lock*-actie en 180 seconden voor de *move*-actie.

III.3.4.3 De procedure

Nadat in “clm_param” en “spea2_param” de parameters voor een simulatie ingegeven zijn, is het programma klaar om uitgevoerd te worden door achtereenvolgens te dubbelklikken op de batchbestanden “run_spea2” en “clm”. Bij het ingeven van de parameters in “clm_param” moet wel opgelet worden of de ingestelde *layout* voldoende groot is voor het aantal aankomende containers dat ingesteld werd.

Eerst kent de procedure een opslaglocatie toe aan elke container en vormt zo de initiële populatie. Hierbij wordt opgelet dat container C_{t1} niet boven C_{t2} geplaatst wordt als t1 en t2 de aankomsttijden zijn van de respectievelijke containers en t1 < t2. Een container mag dus niet toegekend worden aan positie P_t als positie P_{t-1} leeg is.

Het algoritme in deze toepassing maakt gebruik van een 1-point *crossover* met herstellingsmechanisme. Beschouw het voorbeeld in figuur 25. Dit voorbeeld toont aan dat de *crossover* onuitvoerbare oplossingen kan produceren. Zowel in het eerste en tweede chromosoom na *crossover* zijn er containers die toegewezen worden aan dezelfde opslaglocatie. Het herstellingsmechanisme gaat na of dit probleem zich voordoet en als dit het geval is, kent het mechanisme toegelaten locaties toe.

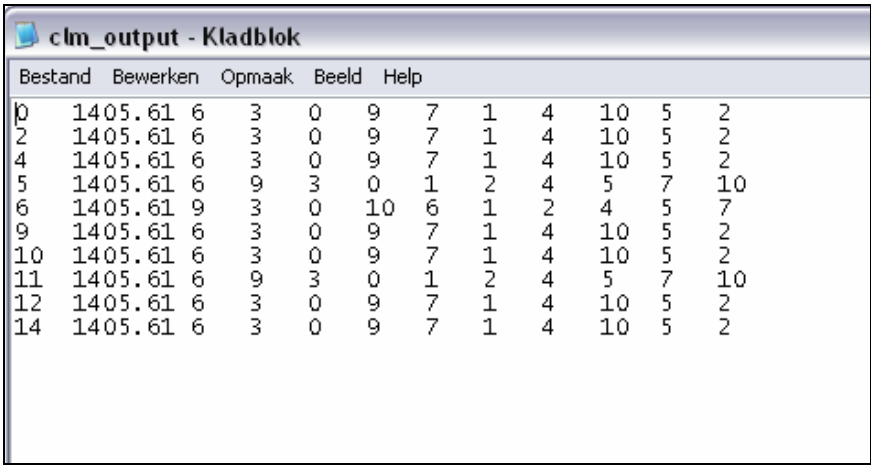


Figuur 25: Voorbeeld van *crossover* en herstellingsfunctie

De procedure maakt, zoals eerder vermeld, gebruik van een 1-bit mutatie. Deze selecteert willekeurig een positie op een chromosoom en verandert de waarde willekeurig. Daarna worden de overige posities binnen het chromosoom, net als bij een *crossover,* aangepast totdat een toegelaten oplossing bekomen wordt.

III.3.4.4 De output

De interpretatie van de outputfile “clm_output” zal nu beknopt uitgelegd worden aan de hand van het eenvoudig voorbeeld in figuur 26.



Figuur 26: Voorbeeld van de outputfile in geval van een simulatie met 10 containers

Er wordt steeds de volledige eindpopulatie gegeven. Het aantal lijnen in de outputfile is dus steeds gelijk aan alpha (α). Dit wil niet zeggen dat er α verschillende oplossingen zijn die allemaal dezelfde optimale doelfunctiewaarde als resultaat hebben. In dit geval zijn er drie verschillende oplossingen die dezelfde minimale tijd voor de uitvoering van de handelingen als gevolg hebben. Deze tijd wordt gegeven in de tweede kolom en is in dit voorbeeld dus 1405,61 seconden. In elke rij stellen de laatste x aantal kolommen samen een chromosoom voor dat lid is van de eindpopulatie. De lengte x van dit chromosoom is gelijk aan het aantal containers dat moet geplaatst worden. In dit geval is x dus gelijk aan 10.

III.4 De experimenten met de software op basis van genetische algoritmen

III.4.1 Instelling van de parameters in verband met het genetische algoritme

Het uiteindelijke doel is om het aantal containers, de *layout* en het behandelingsbeleid te laten variëren tijdens verschillende simulaties aan de hand van de software en daaruit vervolgens conclusies te trekken. Om bij deze testen zo goed mogelijke resultaten te bekomen, moet het genetisch algoritme zo optimaal mogelijk werken. Daarom gaan er eerst verschillende instellingen voor het genetische algoritme uitgeprobeerd worden op een eenvoudige instelling voor het aantal containers, de *layout* en het behandelingsbeleid. Tijdens de eigenlijke experimenten later in de tekst zal de instelling voor het genetisch algoritme gebruikt worden die bij deze testen het beste resultaat oplevert.

De testen om een goede instelling van het genetische algoritme te bekomen, werden uitgevoerd op volgend eenvoudig voorbeeld:

Containers (#)	10
Rows	1
Columns	4
Levels	3
Loading type	FCFS (1)

De resultaten en de 16 verschillende instellingen die getest werden, worden in volgende tabel samengevat. Tevens wil ik nog opmerken dat voor mu en lambda steeds de helft van de populatiegrootte werd ingesteld. Bij een populatie van bijvoorbeeld 50 werden mu (μ) en lambda (λ) dus op 25 gezet.

Nr.	Populatie-grootte (α)	Aantal generaties	Kans op *crossover*	Kans op een mutatie	Totale geschatte tijdsduur van de voorgestelde handelingen (in seconden)
1	10	30	0,5	1	8405.61
2	10	30	0,5	0,5	8405.61
3	10	30	0,8	1	8405.61
4	10	30	0,8	0,5	8405.61
5	10	60	0,5	1	8405.61
6	10	60	0,5	0,5	8405.61
7	10	60	0,8	1	8405.61
8	10	60	0,8	0,5	8405.61
9	50	30	0,5	1	8405.61
10	50	30	0,5	0,5	8405.61
11	50	30	0,8	1	8405.61
12	50	30	0,8	0,5	8405.61
13	50	60	0,5	1	8405.61
14	50	60	0,5	0,5	8405.61
15	50	60	0,8	1	8405.61
16	50	60	0,8	0,5	8405.61

Figuur 27: Instellingen en resultaten in zoektocht naar goede parameters voor genetisch algoritme

Het feit dat steeds dezelfde oplossingen terugkeren, ondanks de instellingen van populatiegrootte, *crossover* en mutatie, doet vermoeden dat reeds na enkele generaties de optimale oplossing bekomen wordt. Daarom werd de simulatie met volgnummer 8 nog enkele keren herhaald, maar telkens met minder generaties. De resultaten hiervan zijn weergegeven in figuur 28.

Aantal generaties	Totale geschatte tijdsduur van de voorgestelde handelingen (in seconden)
60	8405.61
30	8405.61
10	8405.61
6	8405.61
5	8405.61 en 8406.04
4	8405.61, 8406.04 en 8406.47
3	8405.61, 8406.04 en 8406.47
2	8405.61, 8406.04 en 8406.47
1 (initiële populatie)	10 verschillende tijden

Figuur 28: Aflopend aantal variaties en resultaten voor nr. 8 uit figuur 27

Uit figuur 28 blijkt dat bij zes generaties en meer, alle leden van de populatie dezelfde waarde geven voor de doelfunctie, al dan niet via verschillende chromosoomketens. We zullen nu nagaan of dit ook het geval is bij andere instellingen en 6 generaties. Het kiezen voor slechts 6 generaties kort ook de verwerkingstijd van het programma in en dit met behoud van de optimale oplossing die na 30 of 60 generaties ook als eindoplossing zou gegeven worden.

Nr.	Populatie-grootte (α)	Kans op *crossover*	Kans op een mutatie	Totale geschatte tijdsduur van de voorgestelde handelingen (in seconden)
1	10	0,8	0,5	8405.61
2	10	0,8	1	8405.61 en 8406.04 (1x)
3	10	0,5	0,5	8405.61
4	10	0,5	1	8405.61 en 8406.04 (2x)
5	50	0,8	0,5	8405.61
6	50	0,8	1	8405.61 en 8406.04 (16x)
7	50	0,5	0,5	8405.61
8	50	0,5	1	8405.61 en 8406.04 (16x)

Figuur 29: 6 generaties bij verschillende instellingen

Uit deze tabel kunnen we vooreerst afleiden dat de kans op mutatie best op 0,5 gehouden wordt, vermits bij deze instelling slechts enkel de optimale oplossing in de eindpopulatie voorkomt. Tevens lijkt het beter om 0,8 als kans op *crossover* in te geven dan 0,5. Bij regel 2 komt de niet-optimale oplossing van 8406.04 slechts 1 keer voor in de eindpopulatie, terwijl dit in regel 4 twee keer het geval is. Tussen de kleine populatiegrootte in regel 1 tot en met 4 en de grootte α-waarde in regel 5 tot en met 8 is geen verschil waarneembaar dat zou aanleiding geven tot een voorkeur voor 1 van de twee groottes. In de verdere experimenten zullen echter toch 60 generaties gebruikt worden in plaats van 6. Dit omdat bij grotere problemen, met een grotere opslagplaats en meer containers, de optimale tijd niet steeds na 6 generaties gekend is. De chromosomen bij deze leden zijn immers langer en we willen geen risico lopen dat optimale oplossingen over het hoofd worden gezien. De verwerkingstijd van het programma bedraagt immers nog steeds slechts enkele seconden.

In de volgende experimenten zullen dus volgende instellingen gebruikt worden voor het genetische algoritme:

Populatiegrootte (α)	10
Aantal generaties	60
Kans op *crossover*	0,8
Kans op mutatie	0,5

III.4.2 Experimenten

III.4.2.1 FCFS versus LCFS

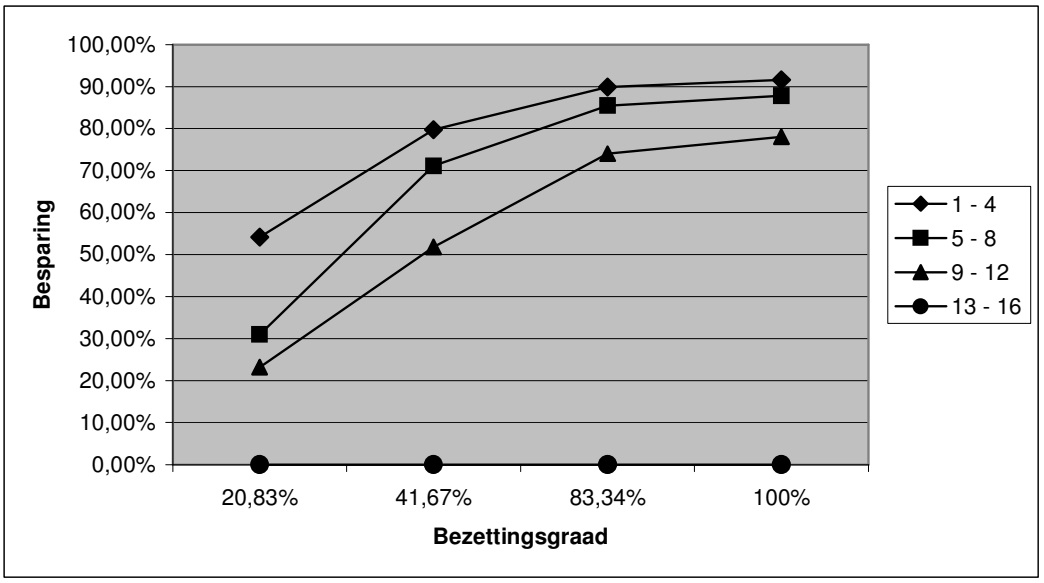
In de publicatie van Preston en Kozan wordt geconcludeerd dat er geen noemenswaardig verschil bestaat tussen het toepassen van het *First Come First Serve* (FCFS) beleid en het toepassen van het *Last Come First Serve* (LCFS) beleid. Hier plaats ik enkele vraagtekens bij vermits bij LCFS geen *rehandles* nodig zullen zijn vermits de gewenste container zich steeds op de bovenste positie van een stapel zal bevinden. Dit zal niet zo zijn bij een FCFS beleid. De hypothese dat FCFS hogere doelfunctiewaarden als gevolg heeft dan LCFS, wordt getest aan de hand van enkele willekeurige *layouts* en containerhoeveelheden. De tests zijn samengevat in figuur 30.

Nr.	Aantal containers	*row*	*column*	*level*	Maximum aantal containers	Bezettings-graad	Resultaat FCFS (in sec.)	Resultaat LCFS (in sec.)	Besparing [(FCFS-LCFS)/FCFS]x100
1	50	6	10	4	240	20,83%	13407.38	6143.72	54,18%
2	100	6	10	4	240	41,67%	60989.07	12359.51	79,73%
3	200	6	10	4	240	83,34%	245319.80	24805.98	89,89%
4	240	6	10	4	240	100%	353809.91	29809.91	91,57%
5	50	8	10	3	240	20,83%	8943.20	6169.83	31,01%
6	100	8	10	3	240	41,67%	43066.98	12444.96	71,10%
7	200	8	10	3	240	83,34%	171687.47	24943.45	85,47%
8	240	8	10	3	240	100%	246017.07	30017.07	87,80%
9	50	6	20	2	240	20,83%	8122.61	6239.53	23,18%
10	100	6	20	2	240	41,67%	26104.43*	12579.18	51,81%
11	200	6	20	2	240	83,34%	97270.79	25260.65	74,03%
12	240	6	20	2	240	100%	138379.59	30379.59	78,05%
13	50	6	40	1	240	20,83%	6359.72	6359.72	0%
14	100	6	40	1	240	41,67%	12977.53	12977.53	0%
15	200	6	40	1	240	83,34%	26122.94	26122.94	0%
16	240	6	40	1	240	100%	31441.27	31441.27	0%
* Er waren 80 generaties nodig omdat na 60 generaties nog geen duidelijke optimale tijd zich aftekende.									

Figuur 30: Experimenten FCFS versus LCFS

Uit deze figuur kan duidelijk geconcludeerd worden dat een beleid op basis van LCFS veel kortere tijden aflevert dan een beleid op basis van FCFS. Door de voorbeelden heen ziet men de tijdsbesparing bij LCFS ten opzichte van FCFS wijzigen. De procentuele tijdsbesparing stijgt naarmate de bezettingsgraad stijgt. Hierbij wordt de bezettingsgraad gedefinieerd als de mate waarin een bepaalde ruimte voor containeropslag gevuld is als alle aankomende containers geplaatst zijn ([*row* x *column* x *level*] / Aantal containers). De verklaring van het fenomeen kan als volgt gegeven worden. Naarmate de containeropslagplaats drukker bezet is, moet er hoger gestapeld worden en zal het effect van *rehandles* op de tijd bij een FCFS beleid toenemen, terwijl een LCFS beleid hier geen hinder van ondervindt.

Als nu voor de 4 *layouts* de besparing uitgezet wordt in een grafiek ten opzichte van de bezettingsgraad, dan valt ook grafisch op dat de besparing stijgt wanneer de bezettingsgraad stijgt. Omdat de grafieken afbuigen naarmate een hogere bezettingsgraad bereikt wordt, kan geconcludeerd worden dat het effect van de bezettingsgraad op de besparing steeds kleiner wordt. De richtingscoëfficiënten van de grafieken nemen af.



Figuur 31: De besparingen ten opzichte van de bezettingsgraad voor gegevens in Figuur 30

Er kan echter nog een tweede opmerking gemaakt worden als naar figuur 31 gekeken wordt. De positie van de grafieken onderling zal bepaald worden door de *layout*. De overige parameters werden immers herhaald voor iedere *layout* die getest werd. Hierbij zal het verschil voornamelijk gedragen worden door de maximum hoogte van een stapeling. De besparing naarmate de bezettingsgraad stijgt, wordt sterker naarmate er hoger gestapeld mag worden. Ook dit is logisch te verklaren. Als er hoger gestapeld mag worden, zullen de kans op *rehandles* en het aantal verplaatsingen zelf toenemen. Dit doet vervolgens op zijn beurt de transporttijd bij FCFS sterk stijgen, terwijl LCFS hier geen invloed van ondervindt omdat hierbij de gewenste containers zich steeds bovenaan een stapel bevinden. Omdat het verschil tussen de tijden bij FCFS en LCFS zijn oorsprong vindt in de verplaatsingen die nodig zijn bij FCFS, zal er geen verschil zijn tussen de tijden die behaald worden als containers niet gestapeld worden en het aantal *levels* dus 1 is.

Als nu ook de tests in figuur 32 gedaan worden voor 200 containers, kan men besluiten dat het verschil tussen FCFS en LCFS procentueel kleiner wordt naarmate een *layout* uit minder rijen en meer kolommen bestaat. Deze tests zijn gelijk met regels 3, 7, 11 en 15 bij de vorige experimenten, alleen zal hier het aantal rijen en kolommen omgekeerd worden. Vervolgens wordt de procentuele besparing van 3 vergeleken met die in 3b, 7 met 7b, enz. De besparingen bij de nieuwe experimenten liggen steeds lager dan die bij de eerdere experimenten. Het verschil is telkens echter niet groot. De belangrijkste factor in de *layout* met betrekking tot het verschil tussen FCFS en LCFS, zal dus *level* of de maximale hoogte zijn.

Nr.	*row*	*column*	*level*	Maximum aantal containers	Bezettings-graad	Resultaat FCFS (in sec.)	Resultaat LCFS (in sec.)	Besparing
3b	10	6	4	240	83,34%	245240.17	24697.88	89,92%
7b	10	8	3	240	83,34%	171733.22	24984.88	85,45%
11b	20	6	2	240	83,34%	96700.90	24706.94	74,45%
15b	40	6	1	240	83,34%	24716.00	24716.00	0%

Figuur 32: Testen met het aantal kolommen en rijen omgekeerd

III.4.2.2 Invloeden op de resultaten

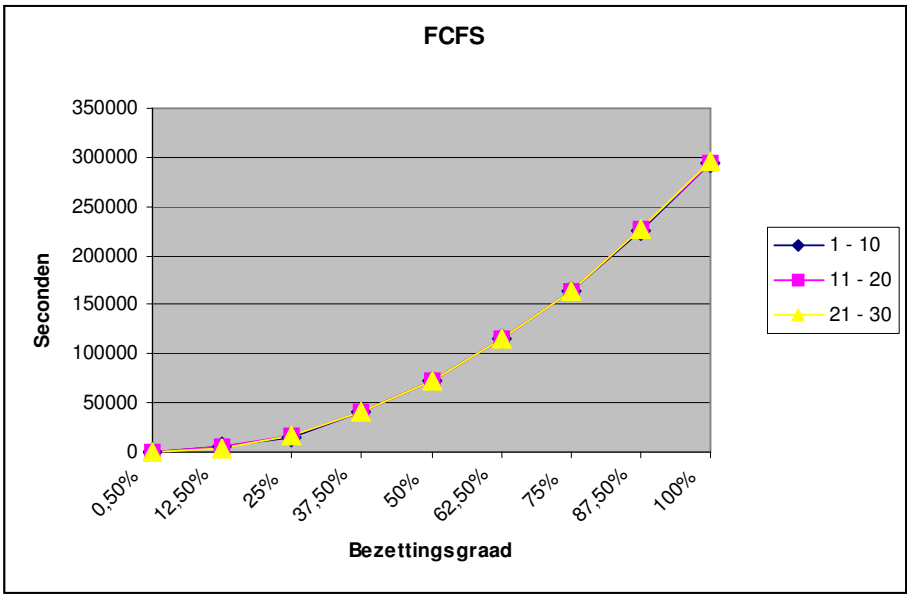
Om de invloeden op de resultaten te onderzoeken werd een reeks van experimenten uitgevoerd. Om de verschillende *layouts* eenvoudig te kunnen vergelijken bieden ze telkens plaats aan maximum 200 containers.

Nr.	Aantal containers	*row*	*column*	*level*	Maximum aantal containers	Bezettings-graad	Resultaat FCFS (in sec.)	Resultaat LCFS (in sec.)
1	1	10	5	4	200	0,5%	120	120
2	10	10	5	4	200	5%	1212.08	1200.00
3	25	10	5	4	200	12,5%	4874.02	3030.21
4	50	10	5	4	200	25%	15148.03	6101.21
5	75	10	5	4	200	37,5%	41629.60	9188.82
6	100	10	5	4	200	50%	72582.47	12247.73
7	125	10	5	4	200	62,5%	115259.51	15320.24
8	150	10	5	4	200	75%	164236.55	18398.79
9	175	10	5	4	200	87,5%	225813.59	21489.42
10	200	10	5	4	200	100%	294604.22	24604.22
11	1	5	10	4	200	0,5%	120	120

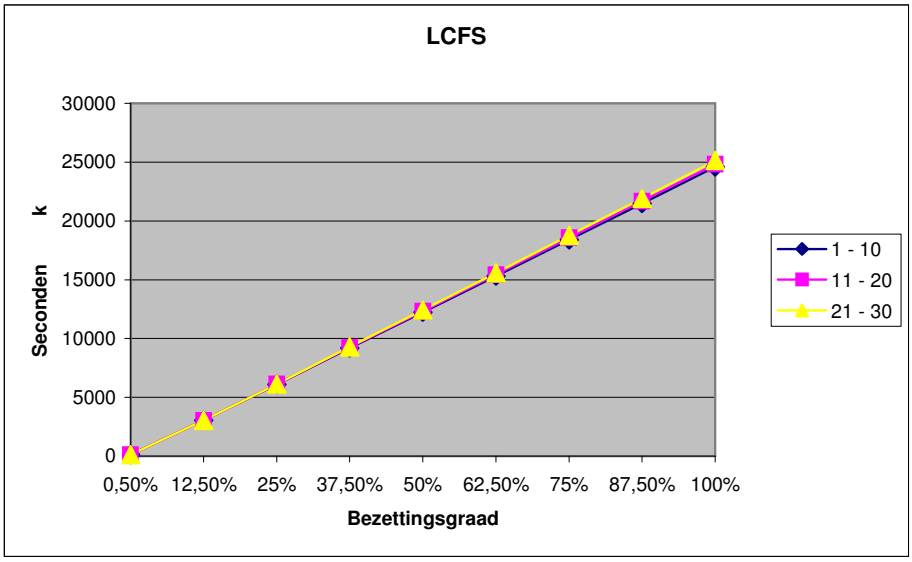
12	10	5	10	4	200	5%	1220.72	1204.53
13	25	5	10	4	200	12,5%	4895.60	3045.75
14	50	5	10	4	200	25%	16104.36	6133.79
15	75	5	10	4	200	37,5%	41711.17	9241.04
16	100	5	10	4	200	50%	72688.21	12332.10
17	125	5	10	4	200	62,5%	116286.83	15442.37
18	150	5	10	4	200	75%	164398.39	18560.84
19	175	5	10	4	200	87,5%	226887.73	21674.57
20	200	5	10	4	200	100%	294820.01	24820.01
21	1	2	25	4	200	0,5%	120	120
22	10	2	25	4	200	5%	1235.17	1207.12
23	25	2	25	4	200	12,5%	4045.44	3037.33
24	50	2	25	4	200	25%	16180.31	6129.91
25	75	2	25	4	200	37,5%	41801.37	9277.29
26	100	2	25	4	200	50%	71926.09	12418.63
27	125	2	25	4	200	62,5%	114672.83	15587.81
28	150	2	25	4	200	75%	164618.05	18781.37
29	175	2	25	4	200	87,5%	226237.17	21904.15
30	200	2	25	4	200	100%	295139.36	25139.36

Figuur 33: Experimenten om de invloed van de bezettingsgraad, het aantal rijen en kolommen op de transporttijd te bestuderen

Als de resultaatwaarden voor de verschillende voorbeelden uitgezet worden ten opzichte van de bezettingsgraad zoals in figuren 34 en 35, kan opgemerkt worden dat bij FCFS de transporttijd in seconden exponentieel stijgt naarmate de bezettingsgraad stijgt en dat bij LCFS deze stijging slechts lineair is. De verschillende voorbeelden bij FCFS en LCFS overlappen ook steeds. De *layout* heeft dus geen invloed op deze relatie tussen de bezettingsgraad en de doelfunctiewaarde.



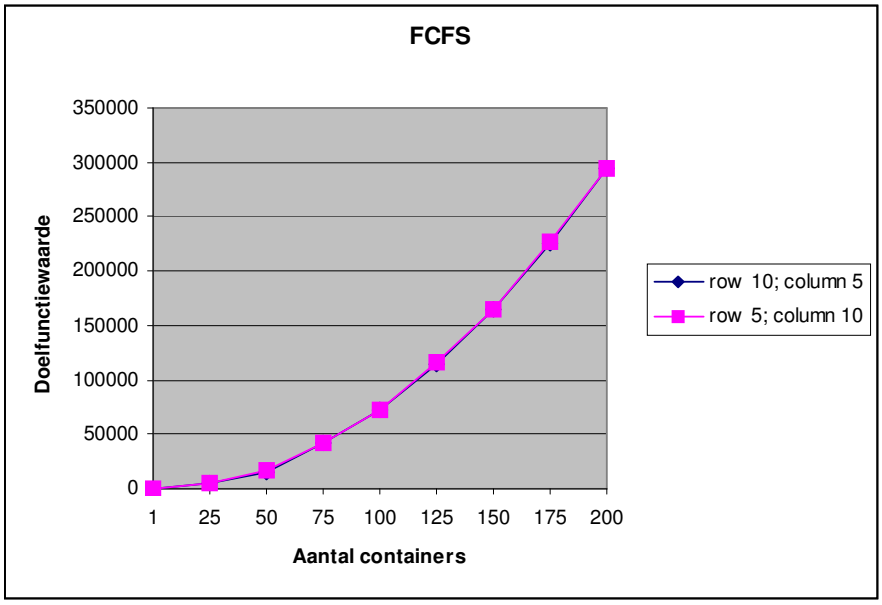
Figuur 34: Effecten van een stijgende bezettingsgraad op de doelfunctiewaarde



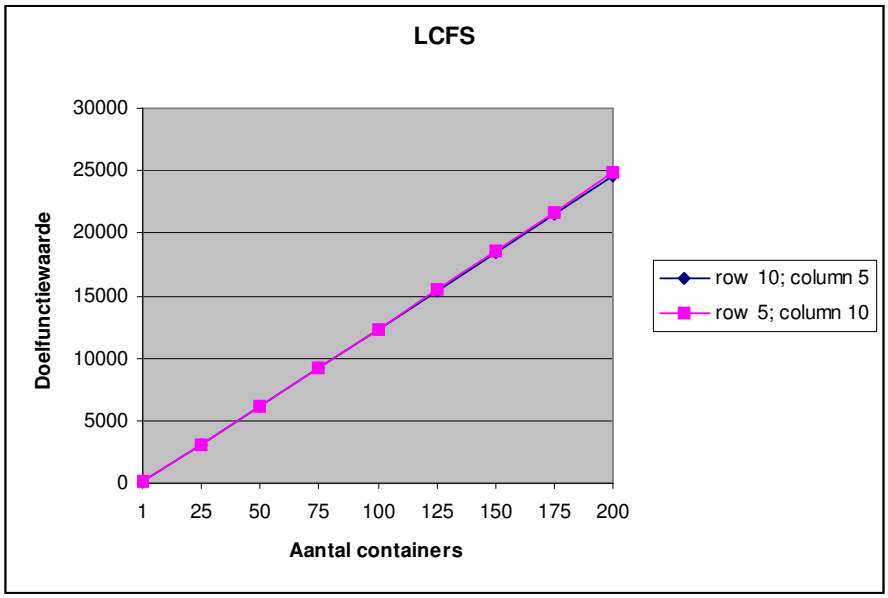
Figuur 35: Effecten van een stijgende bezettingsgraad op de doelfunctiewaarde

Als nu de resultaten voor regels 1-10 (bij verschillende containeraantallen), met *row* = 10 en *column* = 5, en voor regels 11-20, met *row* = 5 en *column* = 10, uitgezet worden in figuren 36 en 37, kan geconcludeerd worden dat het aantal rijen en kolommen bijna geen

rol speelt bij zowel FCFS en LCFS. Op grotere oppervlakken zal het verschil groter worden omdat dan het verschil in de breedte van een rij en de breedte van een kolom sterker tot uiting zal komen. Deze gedachtegang werd geverifieerd aan de hand van een éénvoudig voorbeeldje waarbij de functiewaarden bij 40 rijen en 80 kolommen en de functiewaarden bij 40 kolommen en 80 rijen vergeleken werd.



Figuur 36: Experimenten om de invloed van het aantal kolommen en rijen op de doelfunctiewaarde te achterhalen bij FCFS

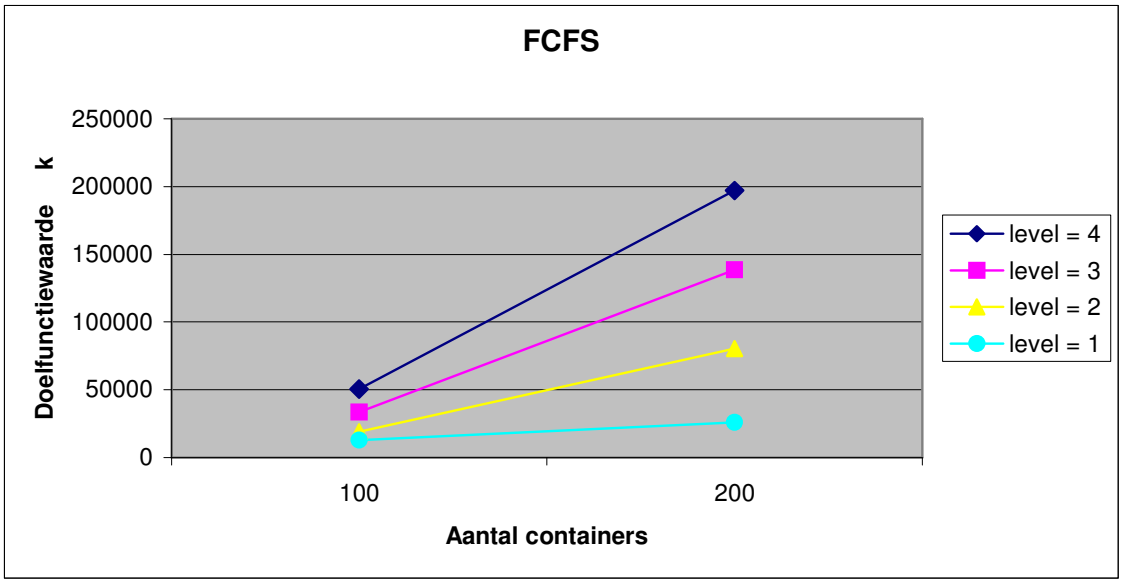


Figuur 37: Experimenten om de invloed van het aantal kolommen en rijen op de doelfunctiewaarde te achterhalen bij LCFS

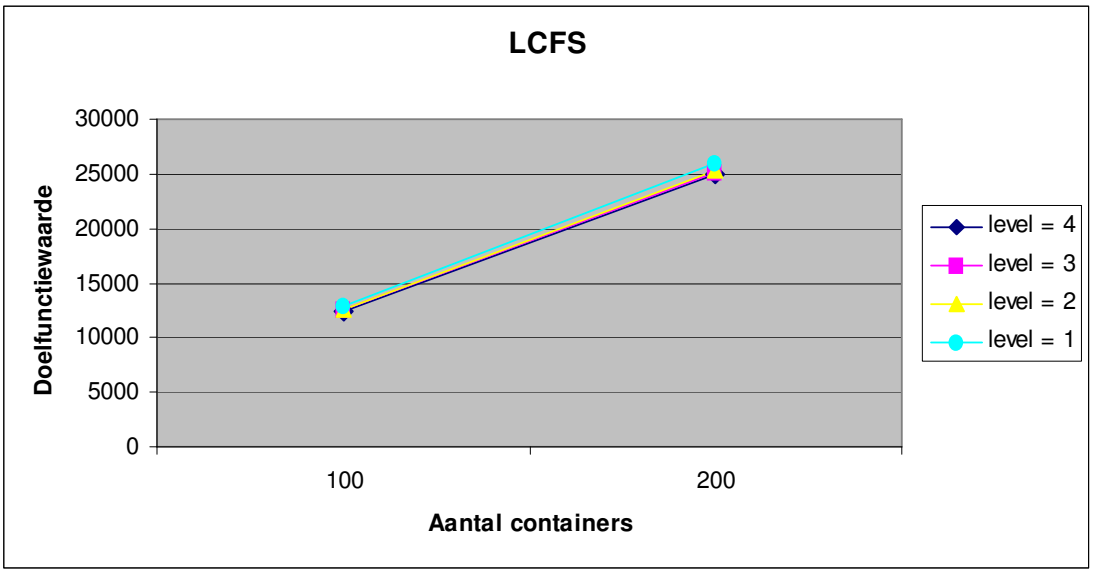
Vervolgens zal nu aan de hand van enkele voorbeelden een afweging gemaakt worden tussen hoger stapelen en een grotere ruimte bestrijken. Vergelijken van verschillende maximale hoogten met behoud van hetzelfde aantal opslagplaatsen gaat niet anders dan met een uitbreiding van de oppervlakte voor opslag.

Aantal containers	*row*	*column*	*level*	Maximum aantal containers	Bezettings-graad	Resultaat FCFS (in sec.)	Resultaat LCFS (in sec.)
100	3	25	4	300	33,33%	50414.57	12367.92
200	3	25	4	300	66,66%	197060.73	25052.84
100	4	25	3	300	33,33%	33334.85	12498.04
200	4	25	3	300	66,66%	138651.80	25201.95
100	6	25	2	300	33,33%	19026.13	12611.12
200	6	25	2	300	66,66%	80351.62	25374.37
100	12	25	1	300	33,33%	12941.07	12941.07
200	12	25	1	300	66,66%	25936.30	25936.30
In deze tabel werden 140 generaties gebruik omdat vaak na 60 generaties nog geen optimale oplossing bereikt was.							

Figuur 38: Experimenten om de wisselwerking tussen oppervlakte en *levels*



Figuur 39: De substitutie van oppervlakte en hoogte (bij FCFS)



Figuur 40: De substitutie van oppervlakte en hoogte (bij LCFS)

De grafische figuren 39en 40 tonen duidelijk de fenomenen die ook in de cijfer van figuur 38 te achterhalen zijn. In het geval van FCFS ziet men duidelijk dat een afname in maximale hoogte en de daarmee samengaande toename in oppervlakte duidelijk positieve gevolgen heeft op de totale transporttijd. Bij LCFS is dit duidelijk niet het geval.

III.4.3 Samenvattende conclusies uit de experimenten

Wanneer LCFS gebruikt wordt in plaats van FCFS treedt er een sterke besparing op in totale transporttijd. Als de bezettingsgraad stijgt, stijgt ook de besparing die kan behaald worden en omgekeerd. Dit effect neemt wel af naarmate men 100% bezetting nadert. De besparing door overschakeling van FCFS op LCFS wordt eveneens groter naarmate er hoger gestapeld mag worden op de opslagplaats. Het verschil in transporttijd loont zeker de moeite om bij de ontvangst aan de landzijde en het wegzetten van containers op de *yard* te pogen om een LCFS beleid toe te passen.

Op de doelfunctiewaarden zelf bij respectievelijk FCFS en LCFS hebben bezettingsgraad, aantal kolommen, aantal rijen en maximale stapelhoogte ook een invloed. Bij FCFS stijgt de transporttijd in seconden exponentieel naarmate de bezettingsgraad stijgt en dat gebeurt bij LCFS slechts lineair. Bij grotere opslagplaatsen kan opgemerkt worden dat men beter meer rijen en minder kolommen kan hebben. De maximale hoogte heeft een sterke invloed op de doelfunctiewaarden bij FCFS, maar geen effect op de totale transporttijd bij LCFS.

Bibliografie

Boeken

Benson, D., Bugg, R. en Whitehead, G. (1994) *Transport and logistics*, New York, Woodhead-Faulkner

Button, K.J. (1993) *Transport economics*, Aldershot, Elgar

Coeck, C., Merckx, J.-P. en Verbeke, A. (2006) *Haveneconomie en –logistiek*, Antwerpen – Apeldoorn, Garant

Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading, Massachusetts, Addison-Wesley Publishing Company

Hillier, F.S. en Lieberman, G.J. (2001) *Introduction to Operations Research*, Singapore, McGraw-Hill Book Co

Kuiler, H.C. (1973) *Inleiding tot de vervoers- en haveneconomie*, Rotterdam, Universitaire Pers Rotterdam

Cursussen

Renders, L. (c. 2001) *Methoden van onderzoek en rapportering 1*, cursus 'Methoden van onderzoek en rapportering 1' UHasselt 2001-2002

Van Breedam, A. (c. 2006) *European Logistics: Trends and evolutions*, cursus 'Logistiek: deel materiaalbeheer' UHasselt 2005-2006

Witlox, F. (c. 2004) *Vervoersbeleid: expeditie – verzending van goederen*, cursus 'Vervoersbeleid' UHasselt 2004-2005

Artikels

Bleuler, S., Laumanns, M., Thiele, L. en Zitzler, E. (2003) 'PISA – A platform and programming Language independent interface for search algorithms', Lecture Notes in Computer Science, volume 2632 / 2003, 494 - 508

Kim, K. H. (1997) 'Evaluation of the number of rehandles in container yards', Computers & Industrial Engineering, volume 32, 701-711

Kim, K. H. en Bae J. W. (1998) 'Re-marshaling export containers in port container terminals', Computers & Industrial Engineering, volume 35, 655-658

Kim, K. H. en Kim, H. B. (1999) 'Segregating space allocation models for container inventories in port container terminals', International journal of production economics, volume 59, 415-423

Kim, K. H., Park, Y. M. en Ryu, K.-R. (2000) 'Deriving decision rules to locate export containers in container yards', European Journal of Operational Research, volume 124, 89-101

Kim, K. H. en Park, K.T. (2003) 'A note on a dynamic space-allocation method for outbound containers', European Journal of Operational Research, volume 148, 92-101

Preston, P. en Kozan, E. (2001) 'An approach to determine storage locations of containers at seaport terminals', Computers & Operations Research, volume 28, 983-995

Steenken, D., Voss, S. en Stahlbock, R. (2004) 'Container terminal operation and operations research – a classification and literature review', OR Spectrum, volume 26, 3-49

Zhang, C., Liu, J., Wan, Y., Murty, K. G. en Linn, R. J. (2003) 'Storage space allocation in container terminals', Transportation Research Part B, volume 37, 883-903

Zitzler, E., Laumanns, M. en Thiele, L. (2001) 'SPEA2: Improving the strength Pareto evolutionary algorithm', Swiss Federal Institute of Technology (ETH) ,Technical report 103, 1-21

Internetbronnen

Computer Engineering and Networks Laboratory (online) (geciteerd augustus 2006) beschikbaar op www.tik.ee.ethz.ch (http://www.tik.ee.ethz.ch/pisa/)

Port of Antwerp (online) (geciteerd 13 mei 2006) beschikbaar op www.portofantwerp.be (http://www.portofantwerp.be/html/05_PORTBROCHURES/AGHApdfNEW/Cargo_Companion_Ned.pdf)

Siemens (online) (geciteerd juli 2006) beschikbaar op www.siemens.com (http://www.automation.siemens.com/_en/mc/cranes/en/c5eb153c-a15e-11d7-b54c-0050da4caaa9/index.aspx)

The transportation institute (online) (geciteerd februari 2006) beschikbaar op www.trans-inst.org (http://www.trans-inst.org/seawords.htm#mg)

Krantenartikels

De Standaard (2005) 'Containers stuwen Antwerpse trafiekcijfers', *De Standaard*, Groot-Bijgaarden, 30 december

Auteursrechterlijke overeenkomst

Opdat de Universiteit Hasselt uw eindverhandeling wereldwijd kan reproduceren, vertalen en distribueren is uw akkoord voor deze overeenkomst noodzakelijk. Gelieve de tijd te nemen om deze overeenkomst door te nemen en uw akkoord te verlenen.

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:
Bepaling van een strategie voor het plaatsen van containers op een containeryard
Richting: **Handelsingenieur** Jaar: **2006**
in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Deze toekenning van het auteursrecht aan de Universiteit Hasselt houdt in dat ik/wij als auteur de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij kan reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

U bevestigt dat de eindverhandeling uw origineel werk is, en dat u het recht heeft om de rechten te verlenen die in deze overeenkomst worden beschreven. U verklaart tevens dat de eindverhandeling, naar uw weten, het auteursrecht van anderen niet overtreedt.

U verklaart tevens dat u voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen hebt verkregen zodat u deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal u als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze licentie

Ik ga akkoord,

Kevin VERELST

Datum: