

Strategisch plannen met BOSS



**Masterscriptie voor de masteropleiding Applied Mathematics aan de Universiteit Twente
Leerstoel: Discrete Mathematics and Mathematical Programming**

Geschreven door: Matthijs Bijl

Augustus 2011

Begeleiders:

Prof. Dr. J.L. Hurink

Dr. ir. G. Post

UNIVERSITEIT TWENTE 

Samenvatting

We richten ons op een probleem uit de logistiek waarin klanten toegeleverd moeten worden vanaf distributiedepots. Een essentiële voorwaarde is dat leveringen vanuit precies één depot moeten komen. Hierbij komt nog dat tegelijk ook de toelevering van de distributiedepots geoptimaliseerd moet worden. De toelevering van de distributiedepots mag wel vanuit meerdere productielocaties afkomstig zijn. Het doel is om de totale kosten van al het transport en productie te minimaliseren.

Het leveren aan de klanten modelleren we als het Multi-Resource Generalized Assignment Problem (MRGAP). Een oplossing voor dit probleem laat voor het toeleveren van de distributiedepots een LP over, dat opgelost moet worden.

In dit verslag ontwikkelen we een algoritme, dat beide deelproblemen oplost. Bij het oplossen van het MRGAP wordt al rekening gehouden met kosten voor de toelevering van de depots. In onze aanpak speelt Simulated Annealing een belangrijke rol. Testen voor kleinere instanties zien er veelbelovend uit, al is uitbreiding gewenst. De implementatie waarmee gewerkt is, kan en moet efficiënter om grotere instanties te testen.

Abstract

We address a specific logistic problem. Customers need deliveries from distribution centers. There's one important constraint: deliveries to customers need to be from precisely one distribution centre. At the same time we need to optimize the production and transport to get enough goods to the distribution centers. Transport to distribution centers is allowed to come from multiple locations.

We model the deliveries to customers as a Multi-Resource Generalized Assignment Problem (MRGAP). A solution for the deliveries leaves a Linear Program for the production and transport.

In this report we develop a heuristic which addresses both parts of the logistic problem. When solving the MRGAP we account for costs of transport and production. In our approach we use Simulated Annealing. Our algorithm has the potential to be suitable for very large instances up to a few thousand customers. However, a more efficient implementation is needed to test this hypothesis.

Voorwoord

Ik ben in 2008 bij ORTEC aan de slag gegaan met deze opdracht. Door persoonlijke problemen kwam er in het jaar dat ervoor gepland stond uiteindelijk niet veel van terecht. Na overleg met ORTEC en mijn begeleider van de Universiteit Twente, Johann Hurink, is besloten om de opdracht te houden, maar om deze verder buiten ORTEC om af te ronden.

Inmiddels was ik al wel naar Gouda verhuisd. Het is wel bijzonder om intern op de Universiteit Twente af te studeren, terwijl je eigenlijk bijna altijd in Gouda bent. Het is niet echt handig, overigens. Gelukkig heb ik verscheidene keren in Gouda en in Utrecht kunnen afspreken met mijn begeleiders.

Uiteindelijk is de opdracht over een hele lange termijn gespreid. Afgelopen jaar heb ik het werken aan de opdracht gecombineerd met een nieuwe baan als docent wiskunde. Dat viel lang niet altijd mee en het was vaak lastig om het overzicht te behouden nadat de opdracht weer een tijdje uit beeld geweest was.

Uiteindelijk is het nu toch gelukt en dat heb ik niet alleen aan mezelf te danken. Ik wil Johann Hurink bedanken voor zijn geduld, zijn tijd en moeite die hij in mij heeft gestoken. Het schrijven aan een verslag als dit, viel me niet mee. Zijn hulp was meer dan welkom.

Ik wil ook Gerhard Post bedanken. Bij het maken van de applicaties is hij erg waardevol geweest. Hij hielp met het opzetten van een beginstructuur waarbinnen ik uiteindelijk goed mijn weg wist te vinden.

Als laatste wil ik mijn ouders bedanken. Het laatste deel van het project heb ik bij hen in huis gedaan. Dit gaf me een goede werkplek en uitstekende catering tussendoor. Ook hun vertrouwen door de tijd heen is altijd erg aangenaam geweest.

Inhoudsopgave

SAMENVATTING	1
ABSTRACT	2
VOORWOORD	3
INHOUDSOPGAVE	4
INLEIDING	5
1.1 LOGISTIEK.....	5
1.2 DRIE NIVEAUS.....	5
1.3 BOSS.....	6
1.4 BEGRIPPEN	8
1.5 LITERATUUR.....	9
1.6 VOORUITBLIK OP DE ANDERE HOOFDSTUKKEN	10
2 PROBLEEMANALYSE	11
2.1 HET BOSS TOEWIJZINGSPROBLEEM.....	11
2.2 MULTI-RESOURCE GENERALIZED ASSIGNMENT PROBLEM	16
2.3 HET RESULTEREND PROBLEEM NA VASTLEGGEN VAN SECUNDAIR TRANSPORT.....	22
3 OPLOSSINGSMETHODE	32
3.1 HERFORMULERING VAN HET BTP	32
3.2 SIMULATED ANNEALING	35
3.3 RAAMWERK VAN HET ALGORITME	36
3.4 HET CREËREN VAN EEN BUUR.....	37
3.5 DE KOSTENFUNCTIE.....	38
3.6 HET GEHELE ALGORITME.....	41
4 RESULTATEN	44
4.1 INSTANTIES GENEREREN.....	44
4.2 TESTRESULTATEN	52
5 CONCLUSIES EN AANBEVELINGEN	55
5.1 CONCLUSIES.....	55
5.2 AANBEVELINGEN.....	55
6 LITERATUURLIJST	57

Inleiding

In dit verslag behandelen we een probleem uit de logistiek. Alvorens we daar diep op ingaan gaan we in dit hoofdstuk eerst kijken in welk deel van de logistiek ons probleem hoort. Daarna wordt ingegaan op waar ons probleem vandaan komt en hoe het er uitziet. Tot slot gaan we kijken hoe soortgelijke problemen in de literatuur aangepakt worden.

1.1 Logistiek

De wereld van tegenwoordig zit vol met logistieke vraagstukken. Consumenten hebben veel keus en voor ieder product moet de productie, het transport en de distributie georganiseerd worden. Naarmate de hoeveelheid groter wordt, wordt de uitdaging om dit optimaal te doen steeds groter. In deze context betekent optimaal bijna altijd: zo goedkoop mogelijk. In werkelijkheid blijkt vaak dat het onmogelijk is om grote processen volledig optimaal te krijgen. Bij dat soort grote processen wordt de uitdaging om zo dicht mogelijk bij een optimale oplossing in de buurt te komen. Dit soort vraagstukken vormen een belangrijk terrein waar wiskunde al jaren tot grote kostenbesparing leidt.

1.2 Drie niveaus

In veel literatuur onderscheidt men vraagstukken en beslissingen binnen de logistiek in drie niveaus. [9] De niveaus verschillen van elkaar in de lengte van tijd waarvoor de beslissing wordt genomen en de mate van detail die er aan te pas komt.

Beslissingen voor langere termijn zijn minder gedetailleerd dan beslissingen voor de kortere termijn. Nu volgt een korte beschrijving van de drie niveaus. De voorbeelden horen bij een situatie waarbij een keten van productie, transport en distributie geanalyseerd wordt.

Allereerst is er het strategische niveau. Dit is voor de hele lange termijn. Denk hierbij aan minimaal twee jaar. Onder strategische beslissingen valt het bepalen van locaties van productie, opslagdepots en distributiecentra.

Het tweede niveau is het tactische niveau. Deze beslissingen zijn voor de middenlange termijn: zes maanden tot twee jaar. Hierbij ligt het onderliggende transportnetwerk al vast. Binnen dit netwerk kunnen de stromen van goederen en diensten geoptimaliseerd worden. Denk hierbij bijvoorbeeld aan voorraadbeheer op de verschillende locaties.

Het derde niveau is het operationele niveau. Deze beslissingen zijn voor de korte termijn: 1 dag tot 6 maanden. Beslissingen van het strategische en tactische niveau liggen vast en daarbinnen moet geopereerd worden. Op het operationele niveau wordt vooral gewerkt aan plannings die moeten zorgen dat iedere individuele order op tijd op de juiste plaats is. Afhankelijk van de bedrijfsomgeving kan het plannings per dag, per week of per maand betreffen.

Dit verslag behandelt een probleem van het strategische niveau. Beslissingen op dit niveau hebben gevolgen voor de mogelijkheden op tactisch niveau die op hun beurt weer gevolgen hebben voor de mogelijkheden op operationeel niveau. Bij een beslissing voor het strategische niveau moet niet worden vergeten, dat er daarna op de lagere niveaus ook processen en vraagstukken ontstaan die opgelost moeten worden. Het is echter niet mogelijk om alle details van de onderliggende niveaus mee te nemen in die beslissingen. Van de andere niveaus zullen schattingen gemaakt moeten worden om op strategisch niveau tot goede keuzes te komen.

1.3 BOSS

Het strategische probleem dat we behandelen in dit verslag gaat over het plannen van de productie en het transport van goederen. De probleemstelling die we hier behandelen is afkomstig uit een computerapplicatie genaamd BOSS. Dit staat voor Business Optimization Strategic System. Deze applicatie is ontwikkeld door ORTEC [10]. Dit is een bedrijf dat zich o.a. bezig houdt met planning van productieprocessen, personeel en transport. Hierbij wordt naast consultancy ook vaak bijpassende software geleverd. Consultants van ORTEC gebruiken voor hun adviezen, onder andere, software die alleen binnen het bedrijf gebruikt wordt. BOSS is een voorbeeld van een stuk software dat intern gebruikt wordt bij ORTEC. Het is gemaakt om bevoorradingsnetwerken te optimaliseren op het strategische niveau.

BOSS is ontwikkeld in samenwerking met een producent van olie en gas. Het doel daarbij was om de levering aan benzinstations vanuit verscheidene raffinaderijen en depots zo efficiënt mogelijk te laten verlopen. BOSS werd hierbij gebruikt om te beslissen welke stations vanuit welk depot toegeleverd zouden gaan worden. BOSS kan niet gebruikt worden voor gedetailleerde planning van het vrachtvervoer.

BOSS kan niet alleen worden ingezet voor distributie van olie en gas. Zo is bijvoorbeeld ook het logistieke netwerk, dat behoort bij het bevoorraden van supermarkten, met BOSS goed te analyseren en te optimaliseren.

Het onderliggende model waar BOSS mee werkt is op te delen in twee stukken. In het eerste deel, het primaire transport, worden goederen vanuit een productielocatie naar hun distributiecentra getransporteerd. Dit kan gebeuren via één of meerdere overslagdepots. Het transport van productielocatie naar distributiecentrum kan ook rechtstreeks, dus zonder overslagdepots ertussen. Het woord productie kan hier ook nog als import geïnterpreteerd worden. Het is dus even goed mogelijk om binnenlands transport te modelleren waarbij import op meerdere plaatsen kan binnenkomen.

In het tweede deel, het secundaire transport, vindt het transport plaats van het distributiedepot naar de klant. Dit transport is altijd zonder verdere overslag ertussen. In het model wordt rekening gehouden met verscheidene capaciteitsrestricties in de depots. Een belangrijke voorwaarde in het model is dat klanten niet willen dat de bevoorrading van een product vanuit twee verschillende depots gebeurt.

BOSS lost het probleem alleen op voor één tijdsperiode. In de cases waarbij BOSS gebruikt wordt gaat het vaak om een lange periode (minstens een jaar). Zoals gezegd: het doel is niet om een planning voor alle ritten van de vrachtwagens te leveren. Het gaat om toewijzingen van klanten aan depots en de goederenstromen die deze toewijzingen met zich mee brengen.

De centrale vraag bij berekeningen is steeds: welke klanten moeten aan welke depots gekoppeld moeten worden voor een zo goed mogelijke distributie van de goederen. Het is ook mogelijk om de vraag te analyseren of een depot beter gesloten kan worden of niet. Het probleem kan dan twee maal worden doorgerekend. Een keer met en een keer zonder dat depot om zo te zien wat de beste optie is. Dit kan worden uitgebreid naar deelverzamelingen van de depots en berekeningen voor alle situaties die dan mogelijk zijn. Omdat BOSS ondersteunt in dit soort grote en belangrijke beslissingen, is het belangrijk dat het onderliggende algoritme, oplossingen levert die van goede kwaliteit zijn.

Voor de uitleg van het model waarop BOSS is gebaseerd, zijn een serie begrippen van belang. Deze worden in de volgende sectie nader toegelicht. Het gaat hier om toelichting van het wiskundige probleem dat opgelost moet worden. In dit probleem kunnen begrippen een beperktere betekenis hebben dan in de totale omgeving van BOSS. In hoofdstuk 2 volgt een formelere beschrijving van het wiskundige probleem.

1.4 Begrippen

In deze sectie worden een aantal belangrijke begrippen nader gespecificeerd.

1.4.1 Producten

Dit is de verzameling van alle producten die vervoerd worden. Kosten voor vervoer kunnen per product verschillen.

1.4.2 Depot

Een depot is een plaats waar goederen in vaak grote hoeveelheden aanwezig kunnen zijn. Bij een depot ligt vast welke deelverzameling van de producten op dat depot aanwezig kunnen zijn. De locatie van een depot ligt ook vast.

Een depot heeft per product een, geen of meerdere van de volgende functies: productie, overslag en distributie.

- Productie betekent dat het depot het product kan produceren en het kan transporteren naar andere depots.
- Overslag betekent dat het depot het product kan ontvangen van andere depots en het weer kan transporteren naar verdere depots.
- Distributie betekent dat het depot het product kan ontvangen andere depots en het kan distribueren naar klanten.

1.4.3 Klant

Iedere klant heeft vraag naar één of meerdere producten. De gevraagde hoeveelheid kan per product verschillend zijn. De locatie van een klant ligt vast. Klanten kunnen alleen producten ontvangen vanuit distributiedepots.

1.4.4 Throughput

Het totaal aan goederen dat een depot verlaat noemen we de throughput van een depot. Het is ook mogelijk om over de throughput van een enkel product of over de throughput van een deelverzameling van de producten te praten. Het gaat dan dus om de hoeveelheid van het product, respectievelijk de deelverzameling van de producten, die het depot verlaten.

1.4.5 Klant-productset

Klant-productsets worden gebruikt om te modelleren dat klanten in het echt bepaalde groepen producten van één leverancier willen ontvangen.

Een Klant-productset is een deelverzameling van de producten waar een klant naar vraagt. Alle Klant-productsets van een klant vormen een partitie van zijn totale

vraag. Deze partitie is altijd gegeven. Alle producten in een klant-productset moeten vanaf hetzelfde distributiedepot worden aangeleverd.

De samenstellingen van klant-productsets kunnen per klant verschillen.

1.4.6 Depot-productset

Depot-productsets worden gebruikt om te modelleren dat een aantal producten in een depot in een speciale ruimte opgeslagen moeten worden. Denk hierbij bijvoorbeeld aan producten die bevroren bewaard moeten worden.

Een Depot-productset is een deelverzameling van de producten die op een depot aanwezig kunnen zijn. De depot-productsets van een depot vormen een partitie van alle producten die op dat depot aanwezig kunnen zijn. Deze partitie is altijd gegeven. Bij een depot-productset wordt altijd een maximum throughput gegeven.

1.4.7 Primair transport

Primair transport is het transport van producten van productiedepots naar distributiedepots. Dit kan via een of meerdere overslagdepots zijn, maar ook direct. Bij de beschrijving van depots is vastgelegd tussen welke depots transport kan plaatsvinden. De hoeveelheid transport tussen twee depots wordt alleen beperkt door beperkingen op de throughput van het depot dat de goederen verzend.

Zoals bij de beschrijving van depot vermeld is, kunnen depots meerdere functies hebben. Het kan dus ook zo zijn dat een productiedepot ook een distributiefunctie heeft. Primair transport kan dan binnen een depot plaatsvinden. Dit soort transport levert in het model geen kosten op.

1.4.8 Secundair transport

Secundair transport is het transport van producten van distributiedepots naar klanten. Bij het analyseren en later ook bij het oplossen van het probleem wordt vaak eerst naar het secundaire transport gekeken en daarna naar het primaire. Dit komt doordat pas bekend is welk primair transport nodig is, op het moment dat het secundaire transport vast ligt. Bij analyses en berekeningen bekijken we daarom vaak eerst het secundaire transport.

1.5 Literatuur

Bij de analyse van het probleem, die in hoofdstuk 2 wordt behandeld, blijkt dat het moeilijkste deel van het oplossen van het probleem zit in het bepalen hoe de toelevering van de klanten moet plaatsvinden, dus in het secundaire transport. Op het moment dat het transport van distributiedepots naar klanten vastligt, blijkt dat

het oplossen van het primaire transport te modelleren is als een Linear Program (LP). Verder is het vinden van een toegelaten oplossing voor het toewijzen van de klanten goed te modelleren als Multi-Resource General Assignment Problem [2]. Dit probleem is NP-hard [2]. Hoofdstuk 2 gaat hier dieper op in.

Bij het bestuderen van literatuur richten we ons op het secundaire transport. Problemen die vergelijkbaar zijn met het secundair transport zijn in de literatuur eerder onderzocht. Gavish [7] maakt gebruik van lagrangerelaxaties en van een branch and bound methode. Mazzola[8] heeft een heuristiek ontwikkeld, waarin onderscheid wordt gemaakt in drie fasen. In deze heuristiek wordt eerst een oplossing opgebouwd. Later wordt deze oplossing nog verbeterd door gebruik te maken van Lagrange relaxaties. Verder bestaat ook nog een vernieuwde versie van het algoritme van Gavish [7] en wordt de eigen heuristiek weer gecombineerd met de vernieuwde versie van Gavish.

Yagiura [2] maakt gebruik van Tabu search. Hij gebruikt daarbij erg grote buurtstructuren. In hoofdstuk 4 komen we hier nog op terug. Goudvis [4] heeft aan dezelfde probleem gewerkt als die ook in dit verslag aan de orde zijn. Hij maakt gebruik van de LP-relaxatie van het probleem.

Ons doel is om een methode te maken die grote instanties aan kan, met een paar duizend klanten. Met instanties van die omvang is nog geen van de genoemde methodes echt getest.

1.6 Vooruitblik op de andere hoofdstukken

In hoofdstuk 2 gaan we de begrippen uit de inleiding en de bijbehorende probleemstelling verder uitdiepen. We gaan het probleem opsplitsen in het toewijzen van de klanten aan de distributiedepots enerzijds en het organiseren van het primair transport en de productie anderzijds.

In hoofdstuk 3 wordt beschreven hoe de deelproblemen opgelost worden en hoe de oplossingen van de deelproblemen op elkaar worden aangepast. De methode die we gebruiken, is gebaseerd op Simulated Annealing.

In hoofdstuk 4 gaan we resultaten van een test van de ontwikkelde methode beschrijven. Voor deze test worden instanties gebruikt die we zelf genereren. De manier, waarop deze instanties gegenereerd worden, zal ook worden uitgelegd.

In hoofdstuk 5 staan conclusies en aanbevelingen.

2 Probleemanalyse

In dit hoofdstuk richten we ons in de eerste plaats op een formele omschrijving van het probleem dat we gaan analyseren. We noemen dit het BOSS

Toewijzingsprobleem (BTP). Zoals al kort werd aangestipt in hoofdstuk 1, kan dit probleem opgedeeld worden in het toewijzen van de klanten (het secundaire transport) en het organiseren van het transport om voldoende goederen op de distributiedepots te krijgen (het primaire transport).

Na de formele definitie volgt een analyse van beide deelproblemen. Het secundaire transport wordt eerst behandeld. In hoofdstuk 1 werd genoemd dat dit te schrijven is als het Multi-Resource General Assignment Problem (MRGAP). We onderzoeken hoe dat MRGAP eruit ziet en wat naar de overeenkomsten en verschillen met het BTP zijn.

Als laatste volgt een analyse van het primaire transport bij een gegeven keuze voor het secundaire transport. Hierbij analyseren wij ook hoe het primair transport afhangt van het secundaire transport en onder welke voorwaarden dit probleem efficiënt oplosbaar is.

2.1 Het BOSS Toewijzingsprobleem

In deze sectie wordt het BOSS Toewijzingsprobleem (BTP) formeel gedefinieerd. We starten met een overzicht van de gegeven verzamelingen die een rol spelen. Daarna volgt een omschrijving van de beslissingsvariabelen. Vervolgens komen de voorwaarden en de doelfunctie aan bod.

2.1.1 Gegevens

Bij een instantie van het Boss Toewijzingprobleem (BTP) zijn de volgende elementen gegeven.

- De verzameling producten, genoteerd als P
- De verzameling klanten, genoteerd als K
- De gevraagde hoeveelheid v_{kp} van een product $p \in P$ door klant $k \in K$
 - o De verzameling van producten $p \in P$ die klant $k \in K$ vraagt, dat wil zeggen de producten p , waarvoor $v_{kp} > 0$; deze wordt genoteerd met P_k .
- De locatie van iedere klant $k \in K$
- Iedere klant $k \in K$ heeft een verzameling klant-productsets Q_k . Klant-productset $q \in Q_k$ bevat een deelverzameling van P_k . De klant-productsets vormen een partitie van P_k , dus

$q_1 \cap q_2 = \emptyset$ voor alle $q_1, q_2 \in Q_k$ met $q_1 \neq q_2$ en $\bigcup_{q \in Q_k} q = P_k$.

De verzameling van alle klant-productsets wordt aangeduid met Q .

- De verzameling producten in klant-productset $q \in Q_k$ ($k \in K$); deze wordt aangeduid met P_q .
- De verzameling depots, genoteerd als D .
- Bij ieder depot $d \in D$ ligt voor ieder product $p \in P$ vast of het depot het product kan produceren, overslaan en/of distribueren.
 - o Met $P_d^D \subseteq P$ wordt de verzameling producten aangeduid die gedistribueerd kunnen worden vanaf depot $d \in D$.
 - o Met $P_d^O \subseteq P$ wordt de verzameling producten aangeduid die overgeslagen kunnen worden vanaf depot $d \in D$.
 - o Met $P_d^P \subseteq P$ wordt de verzameling producten aangeduid die geproduceerd kunnen worden in depot $d \in D$.
 - o Met P_d wordt de verzameling producten aangeduid die geproduceerd, overgeslagen of gedistribueerd kan worden vanaf depot $d \in D$. Dus $P_d = P_d^P \cup P_d^O \cup P_d^D$.
 - o Voor een product kunnen meerdere locaties geschikt zijn voor de productie, overslag of distributie. Verder kan een depot voor een product $p \in P$ meerdere functies hebben. De verzamelingen P_d^P, P_d^O en P_d^D hoeven dus niet disjunct te zijn.

In plaats van per depot te beschouwen welke producten er gedistribueerd (overgeslagen, geproduceerd) kan worden, is het ook mogelijk om per product te beschouwen welke depots dit product kunnen distribueren (overslaan, produceren).

- Met $D_p^D \subseteq D$ wordt de verzameling aangeduid van alle depots die product $p \in P$ kunnen distribueren.
- Met $D_p^O \subseteq D$ wordt de verzameling aangeduid van alle depots die product $p \in P$ kunnen overslaan.
- Met $D_p^P \subseteq D$ wordt de verzameling aangeduid van alle depots die product $p \in P$ kunnen produceren.

Depots kunnen per product meerdere functies hebben. De verzamelingen D_p^D, D_p^O en D_p^P voor een vaste $p \in P$ hoeven dus niet disjunct te zijn.

Verder moet een depot ook voor minstens één product een functie hebben. Dus voor ieder depot $d \in D$ geldt dat $P_d^P \cup P_d^O \cup P_d^D$ niet leeg is.

Alle producten in een klant-productset moeten vanaf één distributiedepot geleverd worden aan de klant. Het is daarom zinvol om D_q^D te definiëren. Met $D_q^D \subseteq D$ wordt de verzameling van depots aangeduid die alle producten uit klant-productset q kunnen leveren. Er geldt $D_q^D = \bigcap_{p \in q} D_p^D$.

- Ieder depot $d \in D$ heeft een verzameling depot-productsets R_d . Depot-productset $r \in R_d$ bevat een deelverzameling van P_d .

De depot-productsets vormen een partitie van P_d , dus
 $r_1 \cap r_2 = \emptyset$ voor alle $r_1, r_2 \in R_d$ met $r_1 \neq r_2$ en $\bigcup_{r \in R_d} r = P_d$.

Bij de depots zijn er drie typen voorwaarden voor de maximum throughput. Er kan bij een depot $d \in D$ een maximum throughput worden opgelegd per product, per depot-productset en per depot als geheel.

- De maximum throughput van een product $p \in P$ bij depot $d \in D$ is t_{dp}^{max} .
- De maximum throughput bij een depot $d \in D$ van een depot-productset $r \in R_d$ is t_{dr}^{max} .
- De maximum throughput bij een depot is t_d^{max} .

In het onderliggende model van BOSS zitten vele parameters die met de kosten van het transport te maken hebben. Er wordt o.a. rekening gehouden met reizen, laden, lossen en administratie. Voor al die onderdelen zijn er verscheidene parameters die variabele kosten en vaste kosten voorstellen. De kostenmodellen in het model van BOSS zijn uitgebreid en zullen hier niet besproken worden. Voor meer informatie, zie [4].

Voor het hier behandelde model is het alleen van belang dat wordt verondersteld dat er een lineair verband bestaat tussen de kosten voor het transport en de getransporteerde hoeveelheid. Dit leidt tot de volgende kostenparameters.

- De kostenparameters voor het secundaire transport c_{dkq}^S geven aan wat het kost om alle goederen van klant-productset $q \in Q_k$ van depot $d \in D_q^D$ naar klant $k \in K$ te transporteren.
- De kostenparameters voor het primaire transport $c_{dd'p}^P$ geven aan wat het kost op een eenheid van product $p \in P$ te transporteren van depot $d \in D_p^P \cup D_p^O$ naar depot $d' \in D_p^O \cup D_p^D$.
- De kosten voor het produceren van een eenheid van product $p \in P$ in depot $d \in D$ zijn c_{dp}^{PR} .

2.1.2 Beslissingsvariabelen

Er moeten in principe drie verschillende beslissingen worden gemaakt. Er moet besloten worden hoeveel er van ieder product geproduceerd moet worden in de depots. Verder moet besloten worden hoe het transport van productie naar de distributie gaat. En als laatste moet besloten worden welke klant-productsets aan welke depots gekoppeld worden.

- Voor het secundaire transport moet besloten worden welke klant-productsets door welke distributiedepots geleverd gaan worden. Hiervoor worden beslissingsvariabelen $x_{kqd} \in \{0,1\}$ gebruikt. Als $x_{kqd} = 1$, dan wordt klant-productset $q \in Q_k$, behorend bij klant $k \in K$ door depot $d \in D$ geleverd. Als $x_{kqd} = 0$, dan is dat niet het geval. Indien een depot niet in staat

is om klant-productset $q \in Q_k$ te leveren, dus $d \notin D_q^D (q \in Q_k)$, dan geldt dus dat $x_{kqd} = 0$.

- Voor het primaire transport moet voor ieder depot worden besloten hoeveel er geproduceerd wordt van ieder product.

Hiervoor worden variabelen $z_{dp} \in [0, \infty)$ gebruikt. De variabele z_{dp} geeft aan hoeveel van product $p \in P$ geproduceerd wordt in depot $d \in D$. Indien geldt dat $d \notin D_p^P$ voor een product $p \in P$, dan volgt dat $z_{dp} = 0$.

- Verder moet voor het primaire transport ook worden besloten welke hoeveelheden van elk product tussen ieder tweetal depots getransporteerd wordt.

Hiervoor worden de variabelen $y_{dd'p}$ gebruikt. De variabele $y_{dd'p}$ geeft aan hoeveel van product $p \in P$ van depot $d \in D$ naar depot $d' \in D$ vervoerd wordt. Als voor een trippel (d, d', p) niet geldt dat voor $d \in D_p^P \cup D_p^O$ en $d' \in D_p^O \cup D_p^D (p \in P)$, dan is $y_{dd'p} = 0$.

Het toewijzen van de klant-productsets aan de depots, d.w.z. het kiezen van de x -variabelen, zorgt ervoor dat de distributiedepots goederen moeten ontvangen. Dit wordt geregeld in het primaire transport met behulp van de y variabelen. Maar als de x -variabelen en de y -variabelen gekozen zijn, dan liggen de z -variabelen vast, omdat in het model niet met voorraden gewerkt wordt. Als besloten is hoe al het transport gaat verlopen dan is bij ieder depot dat produceert duidelijk hoeveel er geproduceerd moet worden. Dat is precies het verschil tussen de hoeveelheid die vertrekt vanaf het depot en de hoeveelheid die binnenkomt in het depot. De z -variabelen zijn afhankelijke variabelen.

2.1.3 Voorwaarden

Om sommige voorwaarden overzichtelijk weer te kunnen geven volgen nu eerst nog enkele definities.

Het beschrijven van throughput

Met t_{dp}^- wordt al het transport van product $p \in P$ dat binnenkomt bij depot $d \in D$ aangeduid. Transport dat binnenkomt bij een depot komt altijd vanaf een ander depot.

$$t_{dp}^- = \sum_{d' \in D} y_{d'dp} \quad (p \in P, d \in D)$$

Met t_{dp}^+ wordt al het transport van product $p \in P$ dat vertrekt bij depot $d \in D$ aangeduid, de throughput dus. Dit is de som van het transport naar andere depots en het transport naar klanten.

$$t_{dp}^+ = \sum_{d \in D} y_{ddr} + \sum_{k \in K} \sum_{q \in Q_k | p \in P_q} v_{kp} x_{kqd} \quad (p \in P, d \in D)$$

In de rechtersommatie moet alleen over de klant-productset gesommeerd worden waar product p in zit.

Er zijn voorwaarden die over de throughput van een depot-productset gaan en over de throughput van een depot. Het is daarom zinvol om t_{dr}^+ en t_d^+ te definiëren als de throughput van een depot-productset respectievelijk de throughput van een depot.

De throughput bij een depot $d \in D$ van een depot-productset $r \in R_d$ is gelijk aan de som van de throughputs van de producten in die depot-productset.

$$t_{dr}^+ = \sum_{p \in r} t_{dp}^+ \quad (d \in D, r \in R_d)$$

Op soortgelijke manier volgt:

$$t_d^+ = \sum_{p \in P} t_{dp}^+ \quad (d \in D)$$

Transport

In het model wordt niet met voorraden gewerkt. Een depot kan dus per product niet meer transporteren naar klanten of andere depots dan er binnenkomt vanuit andere depots plus de geproduceerde hoeveelheid.

$$t_{dp}^- + z_{dp} = t_{dp}^+ \quad (\forall p \in P, \forall d \in D)$$

Toewijzing

Iedere klant-productset moet aan precies één depot worden toegewezen.

$$\sum_{d \in D} x_{kqd} = 1 \quad (\forall k \in K, q \in Q_k)$$

In plaats van sommeren over alle depots (D), had ook gesommeerd kunnen worden over alle depots die klant-productset $q \in Q_k$ kunnen distribueren (D_q). Maar omdat de x -variabelen voor depots die dat niet kunnen toch al op nul zijn gezet, levert een sommatie over alle depots exact hetzelfde resultaat. Ook bij andere voorwaarden zal er op soortgelijke manier gebruik worden gemaakt van variabelen die op 0 zijn gezet. We kiezen voor deze schrijfwijze om de notatie niet moeilijker leesbaar te maken dan nodig is.

Throughput

Bij ieder depot $d \in D$ kan per product $p \in P$ een maximum throughput worden opgelegd.

$$t_{dp}^+ \leq t_{dp}^{max} \quad (\forall d \in D, p \in P)$$

Bij ieder depot $d \in D$ kan per depot-productset $r \in R_d$ een maximum throughput worden opgelegd.

$$t_{dr}^+ \leq t_{dr}^{max} \quad (\forall d \in D, r \in R_d)$$

Bij ieder depot $d \in D$ kan een maximum throughput worden opgelegd.

$$t_d^+ \leq t_d^{max} \quad (\forall d \in D)$$

2.1.4 De doelfunctie

Het doel is het vinden van een toegelaten oplossing die de volgende kostenfunctie minimaliseert. De kosten zijn opgebouwd uit kosten voor productie, primair transport en secundair transport.

$$Kosten = \sum_{d \in D} \sum_{p \in P} c_{dp}^{PR} z_{dp} + \sum_{d \in D} \sum_{d' \in D} \sum_{p \in P} c_{dd'p}^P y_{dd'p} + \sum_{k \in K} \sum_{q \in Q_k} \sum_{d \in D} c_{kqd}^S x_{kqd}$$

2.1.5 Model

Het hierboven beschreven optimaliseringsprobleem leidt tot een Mixed Integer Program (MIP), waarvan de x variabelen binair en de y en z variabelen continu en niet-negatief zijn. MIP's kunnen met behulp van softwarepakketten, zoals CPLEX[11] opgelost worden, maar voor grotere instanties met veel binaire variabelen, wordt de rekentijd erg lang. Dit leidt ertoe dat voor onze doelstelling om instanties met veel klanten te kunnen oplossen, het oplossen van het MIP geen haalbare optie is.

2.2 Multi-Resource Generalized Assignment Problem

Het oplossen van het MIP wordt niet het doel. We gaan methodes zoeken die goede oplossingen kunnen vinden voor het BTP in een acceptabele rekentijd. In de scriptie van Goudvis[4] wordt het BTP vergeleken met het Multi-Resource Generalized Assignment Problem (MRGAP). Voor het oplossen van het MRGAP zijn verscheidene methodes ontwikkeld, zoals beschreven werd aan in sectie 1.6.

Het MRGAP blijkt belangrijke gelijkenissen te vertonen met het BTP. Alvorens we naar die gelijkenissen gaan kijken, zorgen gaan we eerst het MRGAP duidelijk formuleren. Daarna volgt een analyse van de overeenkomsten van en verschillen tussen het MRGAP en BTP.

2.2.1 Formulering MRGAP

Het Multi-Resource Generalized Assignment Problem (MRGAP) wordt in de literatuur als volgt omschreven[2]. Er zijn een aantal taken die verdeeld moeten worden over een aantal agenten. Om een taak uit te kunnen voeren heeft een agent middelen nodig. Aan iedere toewijzing zijn kosten verbonden. Het doel is om alle taken zo te verdelen dat de kosten geminimaliseerd worden.

Gegevens

- De verzameling middelen, genoteerd als M .
- De verzameling taken, genoteerd als J .
- De verzameling agenten, genoteerd als I .
- De hoeveelheid van middel $m \in M$ die agent $i \in I$ nodig heeft om taak $j \in J$ uit te voeren is a_{ijm} . Er wordt aangenomen dat $a_{ijm} \geq 0 \forall i \in I, j \in J, m \in M$.
- De hoeveelheid van middel $m \in M$ die agent $i \in I$ beschikbaar heeft is b_{im} .
- De kosten voor het toewijzen van taak $j \in J$ aan agent $i \in I$ zijn f_{ij} .

Beslissingsvariabelen

Er moet besloten worden welke taak wordt toegewezen aan welke agent. Dit wordt gemodelleerd met behulp van variabelen w_{ij} , waarbij $w_{ij} = 1$ als taak $j \in J$ wordt toegewezen aan agent $i \in I$ en $w_{ij} = 0$ indien dit niet het geval is.

Voorwaarden

De agenten kunnen niet meer middelen gebruiken dan ze beschikbaar hebben.

$$\sum_{j \in J} w_{ij} a_{ijm} \leq b_{im} \quad (\forall i \in I, m \in M)$$

Iedere taak moet aan één agent toegewezen worden.

$$\sum_{i \in I} w_{ij} = 1 \quad (\forall j \in M)$$

Doel

Het doel is het om een toewijzing te vinden die aan de voorwaarden voldoet en daarbij de totale kosten,

$$\sum_{i \in I} \sum_{j \in J} w_{ij} f_{ij},$$

te minimaliseren.

2.2.2 NP-hard

MRGAP is een generalisatie van het General Assignment Problem (GAP). Het GAP wordt verkregen uit het MRGAP door het aantal middelen gelijk aan één te nemen. Omdat het partitieprobleem gereduceerd kan worden tot een instantie van GAP met twee agenten is het GAP, en dus ook het MRGAP NP-hard.

Het GAP is op zijn beurt weer een generalisatie van het Assignment Problem (AP). In het AP geldt dat het aantal taken en het aantal agenten precies aan elkaar gelijk zijn. Verder geldt dat iedere agent in staat is om precies één taak uit te voeren. Voor het

AP bestaan wel algoritmes die het probleem in polynomiale tijd kunnen oplossen. Een voorbeeld hiervan is de Hongaarse methode[5].

2.2.3 Van het MRGAP naar het BTP

Om te laten zien dat het BTP NP-hard is, kunnen we laten zien dat iedere instantie van het MRGAP te schrijven is als een instantie van het BTP.

- De verzameling middelen in het MRGAP wordt de verzameling producten in het BTP.
- De verzameling agenten in het MRGAP wordt de verzameling van distributiedepots in het BTP.
- De verzameling van taken in het MRGAP wordt de verzameling van klant-productsets in het BTP.
- In het MRGAP is er de voorwaarde dat iedere agent voldoende middelen moet hebben voor al zijn toegewezen taken. In het BTP wordt dat de voorwaarde dat ieder distributie voldoende throughput aan moet kunnen voor de toegewezen klant-productsets.
- In het MRGAP moet elke taak aan precies één agent toegewezen worden. In het BTP moet elke klant-productset aan precies één distributiedepot worden toegewezen.
- Doel in het MRGAP is minimaliseren van de totale kosten, veroorzaakt door de toewijzingen van taken aan agenten. In het BTP wordt dat het minimaliseren van de totale kosten die veroorzaakt worden door toewijzingen van klant-productsets aan distributiedepots.

Om te zorgen dat de oplossing van het BTP dat ontstaat uit een instantie van het MRGAP ook de oplossing van die instantie van het MRGAP is, mag het primaire transport geen extra voorwaarden opleveren en geen rol in de kosten spelen. Hier is voor te zorgen door voor ieder distributiedepot voor ieder product de maximale productie gelijk te stellen aan de maximale throughput van dat product. De productiekosten moeten gelijk worden gekozen aan 0. Primair transport is in zo een instantie van het BTP niet nodig.

Iedere instantie van het MRGAP kan op deze manier geschreven worden als een instantie van het BTP. Het MRGAP is dus een vereenvoudiging van het BTP, en daarmee is het BTP NP-hard.

2.2.4 Van het BTP naar het MRGAP

Om te onderzoeken hoe bruikbaar oplosmethoden voor MRGAP zullen zijn bij het oplossen van het MRGAP, gaan we kijken onder welke voorwaarden een instantie van het BTP geschreven kan worden als een instantie van het MRGAP. We beginnen met een instantie waar veel extra eisen gelden. Sommige eisen blijken later niet noodzakelijk te zijn om een instantie van het BTP als een instantie van het MRGAP te schrijven.

De eisen

We gaan nu een speciaal geval van het BTP beschouwen. We zullen laten zien dat dit speciale geval te schrijven is als een instantie van het MRGAP. Neem aan dat ieder distributiedepot ieder product kan produceren waarbij de kosten van de productie overal gelijk zijn. Al het primaire transport is dan overbodig geworden. Neem verder aan dat er geen voorwaarden zijn over de totale throughput van een depot of over de throughput van een depot-productset.

De transformatie

Na deze vereenvoudiging is het BTP bijna identiek aan het MRGAP. De klant-productsets in het BTP worden de taken in MRGAP. De depots in het BTP worden de agenten in het MRGAP en de producten in het BTP worden de middelen in het MRGAP. De kostenparameters van toewijzingen moeten alleen worden aangepast in verband met de extra kosten van de productie. De kosten van een toewijzing van een taak (klant-productset) aan een agent (distributiedepot) stijgen alle agenten even veel, omdat de productiekosten per product overal gelijk zijn. De waarden van de variabelen bij de optimale oplossing verandert dus niet.

Wat onder de genoemde aannames overblijft, is het toewijzen van klant-productsets (taken) aan depots (agenten) waarbij ieder depot voldoende hoeveelheden van de producten (middelen) moet hebben voor alle klant-productsets die toegewezen worden. Aan iedere toewijzing zijn kosten verbonden en het doel is om alle klant-productsets toe te wijzen met minimale kosten.

2.2.5 Depot-productsets van het BTP naar het MRGAP

Hierboven werden condities omschreven waarin een instantie van het BTP getransformeerd kan worden naar een instantie van het MRGAP. Eén van die condities was dat er geen voorwaarden waren voor de throughput van een depot-productset. Hieronder wordt behandeld of een instantie met throughputvoorwaarden voor depot-productsets ook gereduceerd kan worden tot MRGAP.

Impliciete throughputvoorwaarden

Voor het transformeren van de depot-productsets straks, onderzoeken we eerst throughputvoorwaarden voor deelverzamelingen in het algemeen. Neem een willekeurige instantie van het BTP. Bij ieder depot $d \in D$ geldt dat er impliciet een maximale throughput vast ligt voor iedere deelverzameling van de producten $P^* \subseteq P$. Bij een depot $d \in D$ is van deelverzameling P^* niet meer throughput mogelijk dan de som van de maximale throughput van de producten in de deelverzameling. Daarbij is het ook niet mogelijk dat er van die deelverzameling meer throughput is dan de maximale throughput van het depot. Als laatste is het ook niet mogelijk dat er meer throughput van die deelverzameling P^* is dan de maximale throughput van iedere depot-productset waar P^* een deelverzameling van is. We noemen $t_{dp^*}^{max}$ de hieruit resulterende maximale throughput van deelverzameling $P^* \subseteq P$ bij depot $d \in D$. Er geldt:

$$t_{dp^*}^{max} = \min \left[\sum_{p \in P^*} t_{dp}^{max}, t_d^{max}, \min_{r | P^* \subseteq r} t_{dr}^{max} \right]$$

Het wegwerken van een depot-productset

We willen de verzameling van instanties van het BTP die we kunnen reduceren tot een instantie van het MRGAP gaan uitbreiden. We bekijken daartoe wederom een speciaal geval van het BTP, maar met minder eisen dan het eerdere speciale geval dat we gereduceerd hebben. We gaan daarna kijken of dit speciale geval ook te reduceren is tot een MRGAP. De instantie die we gaan bekijken heeft de volgende eigenschap:

- Ieder depot kan ieder product produceren en de kosten van de productie zijn overal gelijk.

Over throughputvoorwaarden voor depot-productsets en depots stellen we nu dus geen eisen. We gaan hieronder laten zien dat een depot-productset vervangen kan worden door het toevoegen van een extra product.

De verzameling van alle depot-productsets noteren we als R . De deelverzameling van depots waarbij er een throughputvoorwaarde bestaat voor depot productset $r \in R$ noteren we als D_r .

We nemen een zekere depot-productset $r^* \in R$. Bij de depots $d \in D_{r^*}$ is er een throughputvoorwaarde voor depot-productset r^* . Om r^* te vervangen introduceren we product p^{r^*} . Nu gaan we voor product p^{r^*} de throughputvoorwaarden zo opstellen, zodat ze equivalent zijn aan de throughputvoorwaarden voor r^* .

Voor de depots $d \in D_{r^*}$ moet nu dus gelden dat de maximum throughput $t_{dp^{r^*}}^{\max}$ voor het nieuwe product p^{r^*} gelijk is aan de maximum throughput van de depot-productset r^* .

$$t_{dp^{r^*}}^{\max} = t_{dr^*}^{\max} \quad (\forall d \in D_{r^*})$$

Voor de andere depots gebruiken als maximale throughput het impliciete maximum dat al bestond voor depot-productset r^* .

$$t_{dp^{r^*}}^{\max} = \min \left[\sum_{p \in P^*} t_{dp}^{\max}, t_d^{\max}, \min_{r | r^* \subseteq r} t_{dr}^{\max} \right] \quad (\forall d \in D \setminus D_{r^*})$$

Om de throughputvoorwaarden voor product p^{r^*} echt equivalent te maken aan de throughputvoorwaarden voor depot-productset r^* , moet er bij de klant-productsets ook een aanpassing wat betreft de gevraagde hoeveelheid naar het nieuwe product p^{r^*} gedaan worden. Anders zouden de transportvoorwaarden voor het nieuwe product p^{r^*} verschillen van de impliciete transportvoorwaarden voor depot-productset r^* :

$$t_{dr^*}^- + z_{dr^*} = t_{dr^*}^+ \quad (\forall d \in D)$$

Algemeen geldt dat de impliciete vraag v_{qr} van klant-productset $q \in Q$ naar depot-productset $r \in R$ is te berekenen als de som van de gevraagde hoeveelheden v_{qp} van alle producten p in r .

$$v_{qr} = \sum_{p \in r} v_{qp} \quad (\forall q \in Q)$$

We kiezen voor de vraag $v_{qp^{r^*}}$ van een klant-productset $q \in Q$ naar product p^{r^*} voor de impliciete vraag naar depot-productset r^* .

$$v_{qr^*} = \sum_{p \in r^*} v_{qp} \quad (\forall q \in Q)$$

De voorwaarden die nu gelden voor product p^{r^*} maken de voorwaarden voor depot-productset r^* overbodig. Om te zorgen dat het nieuwe product geen invloed zal hebben op waarden van de doelfunctie, worden alle kostenparameters voor product p^{r^*} gelijk aan nul gekozen. De depot-productset r^* is nu volledig vervangen. Deze kan dus verwijderd worden. Door het toevoegen van een extra product aan een instantie kan dus een nieuwe, equivalente instantie worden gecreëerd met een depot-productset minder.

Meer transformaties

De hierboven beschreven methode kan herhaald worden toegepast voor iedere depot-productset in een instantie. Ook het transformeren van voorwaarden voor een depot als totaal werkt als zodanig. Op deze manier kunnen bij een instantie alle throughputvoorwaarden voor depot-productsets en depots vervangen worden door het invoeren van extra producten. Zo komen we weer uit op een instantie van het BTP, waarvan we al hadden gezien dat deze te transformeren is naar het MRGAP. Het aantal instanties van het BTP dat we kunnen transformeren naar het MRGAP hebben we dus uitgebreid.

2.2.6 Bruikbaarheid van MRGAP

We hebben nu gezien dat het MRGAP en het BTP grote gelijkenissen vertonen. Indien het primaire transport wegvalt, kan een instantie van het BTP worden getransformeerd naar een instantie van het MRGAP.

2.3 Het Resultierend Probleem na vastleggen van secundair transport

Hiervoor hebben we het secundaire transport geanalyseerd en vergeleken met het MRGAP. Hierbij kwam het primaire transport eigenlijk niet aan bod. Nu gaan we ons juist richten op dat primaire transport. Bij het bestuderen daarvan beschouwen we het secundaire transport als gegeven.

Als gegeven is hoe het secundaire transport eruit ziet, blijft een probleem over waarin gezorgd moet worden dat de distributiedepots voldoende van ieder product binnenkrijgen of zelf produceren om aan de klanten te kunnen leveren. Dit probleem noemen we het Resultierend Probleem (RP). Het bespreken van het RP gebeurt aan de hand van een voorbeeld. Voordat we daarmee beginnen kijken we eerst naar noodzakelijke voorwaarden voor oplosbaarheid van het BTP.

2.3.1 Productie toegelaten

We gaan enkele noodzakelijke voorwaarden voor oplosbaarheid bestuderen.

Eerst kijken wij naar de totale vraag naar een product. Dit is de som van de vraag van alle klanten naar dit product. Voor een toegelaten instantie moet het voor de depots mogelijk zijn om voor ieder product deze totale vraag te produceren. Omdat de productie van een product alleen beperkt wordt door throughputvoorwaarden, moeten deze bij elkaar opgeteld voor alle depots, die het product kunnen produceren dus groter of gelijk aan de totale vraag naar het product zijn. Als het in een instantie voor de productiedepots mogelijk is om voor ieder product de totale vraag te produceren dan noemen we zo een instantie *productie toegelaten*.

Voor oplosbaarheid van een instantie geldt een soortgelijke eis bij de distributie. Voor ieder product moeten alle depots die het product kunnen distribueren in staat zijn om de totale vraag van het product te distribueren. Indien in een instantie alle distributiedepots voor ieder product in staat zijn om de totale vraag te distribueren, dan noemen we zo een instantie *distributie toegelaten*.

Het is voor een instantie een noodzakelijke voorwaarde voor oplosbaarheid om zowel productie als distributie toegelaten als te zijn. Het is echter geen garantie voor oplosbaarheid als een instantie productie en distributie toegelaten is. Een eenvoudig voorbeeld om dit aan te tonen is een klant die van een product een grotere gevraagde hoeveelheid heeft dan de maximale throughput van ieder distributiedepot.

Algemener: de voorwaarde dat iedere klant-productset door één depot toegeleverd moet worden in combinatie met throughputvoorwaarden bij de distributiedepots kan het probleem onoplosbaar maken, zonder dat dit altijd eenvoudig in te zien is.

Stel nu dat voor een productie toegelaten instantie het secundaire transport al opgelost is. De vraag die blijft is of er dan altijd een oplossing bestaat voor de productie en het primair transport, passend bij de gegeven toewijzing van het secundair transport. Deze vraag wordt hieronder behandeld.

2.3.2 Een voorbeeld

De uitleg hoe het Resultierend Probleem (RP) na het vastleggen van het secundaire transport eruit ziet volgt hieronder aan de hand van een voorbeeld. In dit voorbeeld zijn er twee producten (p_1 en p_2), vier depots (d_1, d_2, d_3 en d_4) en twee klanten (k_1 en k_2).

Beide klanten hebben een vraag naar beide producten van 50 eenheden. Voor beide klanten geldt dat de producten geen onderdeel zijn van klant-productsets. De bevoorrading van de verschillende producten mag dus vanuit verschillende depots plaatsvinden.

De gegevens van de depots staan in tabel 1.

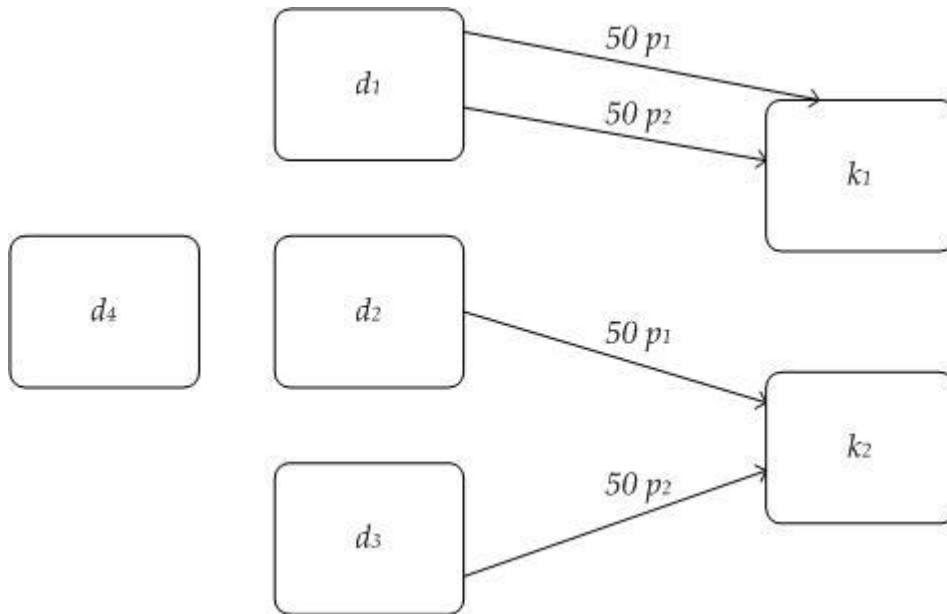
Depot	d_1	d_2	d_3	d_4
Productie	p_1, p_2			p_1
Overslag				
Distributie	p_1, p_2	p_1	p_2	
$t_{p_1}^{max}$	100	100	0	100
$t_{p_2}^{max}$	100	0	100	0
t^{max}	150	100	100	100

Tabel 1: gegevens van de depots uit het voorbeeld

De eerste drie rijen geven aan welke producten het depot kan produceren, overslaan en distribueren. De twee rijen daaronder geven bij ieder depot aan wat de maximale

throughput is van de afzonderlijke producten. De onderste rij geeft weer wat de maximale throughput van het depot als geheel is. Parameters over de kosten blijven in het voorbeeld achterwege. Er wordt alleen gekeken naar de oplosbaarheid van het probleem.

Een mogelijke oplossing voor het secundaire transport staat grafisch weergegeven in figuur 1. Geen enkel depot distribueert een product dat niet toegestaan is en aan alle throughputvoorwaarden wordt voldaan.



Figuur 1: een grafische oplossing voor het voorbeeld

De vraag die nu overblijft is: bestaat er bij deze keuze voor het secundaire transport een invulling van het primaire transport die het volledige probleem oplost? Voordat het oplossen van dit voorbeeld verder gaat, gaan we eerst het algemene geval van dit probleem verder beschouwen.

2.3.3 Algemeen

Indien het secundaire transport vastligt, dan betekent dat, dat alle x -variabelen gekozen zijn. Voor alle depots is hiermee de throughput voor het secundaire transport vastgelegd. We noemen dit t_{dp}^S . Er geldt:

$$t_{dp}^S = \sum_{k \in K} \sum_{q \in Q_k | p \in P_q} v_{kp} x_{kqd} \quad (p \in P, d \in D)$$

Zoals eerder vermeld is de totale throughput van een product bij een depot:

$$t_{dp}^+ = \sum_{d' \in D} y_{dd'p} + \sum_{k \in K} \sum_{q \in Q_k | p \in P_q} v_{kp} x_{kqd} = \sum_{d' \in D} y_{dd'p} + t_{dp}^S \quad (p \in P, d \in D)$$

Nu het secundaire deel van het transport vastligt, is alleen het primaire deel, t_{dp}^P nog variabel. Er geldt:

$$t_{dp}^P = t_{dp}^+ - t_{dp}^S \quad (p \in P, d \in D)$$

Dit resulteert in:

$$t_{dp}^P = \sum_{d' \in D} y_{dd'p} \quad (p \in P, d \in D)$$

Op identieke wijze krijgen we waardes $t_{dr}^S, t_{dr}^P, t_d^S$ en t_d^P voor de throughput in het secundaire en primaire transport van een depot-productset en die van een depot.

Voor de beschrijving van het RP spelen de klanten dus geen rol meer. Alle variabelen die daarmee te maken hebben, liggen immers vast. De y - en z -variabelen moeten nog wel gekozen worden. Deze variabelen komen terug in de voorwaarden voor het transport en in de voorwaarden over de maximale throughput bij depots.

Voor transport zijn de voorwaarden voor het probleem als altijd:

$$t_{dp}^- + z_{dp} = t_{dp}^+ \quad (\forall p \in P, \forall d \in D)$$

Nu het secundaire transport bekend is, kan de throughput worden opgesplitst in een deel dat vastligt en een deel dat nog bepaald moet worden. Hiervoor gebruiken we de zojuist ingevoerde notatie:

$$t_{dp}^- + z_{dp} = t_{dp}^S + t_{dp}^P \quad (\forall p \in P, \forall d \in D)$$

Een overeenkomstige substitutie kan ook worden toegepast in de throughput-voorwaarden.

Bij ieder depot $d \in D$ kan per product $p \in P$ een maximum throughput worden opgelegd.

$$t_{dp}^+ \leq t_{dp}^{max} \quad (\forall d \in D, p \in P)$$

Na substitutie wordt dit

$$t_{dp}^S + t_{dp}^P \leq t_{dp}^{max} \quad (\forall d \in D, p \in P)$$

Bij ieder depot $d \in D$ kan per depot-productset $r \in R_d$ een maximum throughput worden opgelegd.

$$t_{dr}^+ \leq t_{dr}^{max} \quad (\forall d \in D, r \in R_d)$$

Na substitutie wordt dit

$$t_{dr}^S + t_{dr}^P \leq t_{dr}^{max} \quad (\forall d \in D, r \in R_d)$$

Bij ieder depot $d \in D$ kan een maximum throughput worden opgelegd.

$$t_d^+ \leq t_d^{max} \quad (\forall d \in D)$$

Na substitutie wordt dit

$$t_d^S + t_d^P \leq t_d^{max} (\forall d \in D)$$

De doelfunctie van het RP bevat alleen de kosten uit de oorspronkelijke doelfunctie die met het primaire transport te maken hebben.

$$Kosten = \sum_{d \in D} \sum_{p \in P} c_{dp}^{PR} z_{dp} + \sum_{d \in D} \sum_{d' \in D} \sum_{p \in P} c_{dd'p}^P y_{dd'p}$$

2.3.4 Voorbeeld (vervolg)

We vervolgen nu het voorbeeld. In tabel 2 staan de relevante parameters welke waarde ze in het voorbeeld hebben.

Depot	d_1	d_2	d_3	d_4
Productie	p_1, p_2			p_1
Overslag				
Distributie	p_1, p_2	p_1	p_2	
$t_{p_1}^{max}$	100	100	0	100
$t_{p_2}^{max}$	100	0	100	0
t^{max}	150	100	100	100
$t_{dp_1}^S$	50	50	0	0
$t_{dp_2}^S$	50	0	50	0
t_d^S	100	50	0	0

Tabel 2: parameters na vastleggen van secundair transport in het voorbeeld

2.3.5 Gevolgen van een toewijzing van het secundaire transport (algemeen)

We gaan nu kijken naar de mogelijke gevolgen van een toewijzing van het secundaire transport voor het RP. Het belangrijkste is de vraag of een toewijzing zou kunnen zorgen dat het RP onoplosbaar zou kunnen worden.

Er zijn instanties waarin depots zowel kunnen produceren als distribueren. In het behandelde voorbeeld is dit ook het geval. Indien een dergelijk depot goederen gaat distribueren, dan heeft dat invloed op de bovengrenzen van de productie van bepaalde goederen.

De maximale productie van een product p in een depot d is begrensd door de volgende bovengrenzen:

- I. De maximale throughput van product p , t_{dp}^{max}
- II. De maximale throughput van de depot-productset $r \in R_d$ waar het product p in zit met daarvan afgetrokken alle throughput van de andere producten in die depot-productset. Dat is:

$$t_{dr}^{max} - \sum_{p' \in r (p' \neq p)} t_{p'}^S$$

III. De maximale throughput van het depot met daarvan afgetrokken alle throughput van alle andere producten. Dat is:

$$t_d^{max} - \sum_{(p' \neq p)} t_{p'}^S$$

Door een toewijzing van het secundaire transport zal I niet veranderen, maar II en III uiteraard wel. Als de waarde bij II of III lager wordt dan de waarde bij I, dan zijn er in het RP minder productiemogelijkheden, dan in de oorspronkelijke instantie.

Het begrip productie toegelaten is behandeld in de context van een volledige instantie. Het is ook zinvol om te kijken of een instantie van het RP productie toegelaten is. De definitie blijft dezelfde: een instantie van het RP is productie toegelaten indien alle productiedepots in staat zijn om de totale vraag te produceren.

Voor een volledige instantie is het een noodzakelijke voorwaarde dat deze productie toegelaten is. We hebben laten zien dat deze voorwaarde dan niet voldoende was, omdat er onoplosbaarheid kon ontstaan bij het secundair transport. De vraag is of bij een toewijzing van het secundair transport een productie toegelaten RP altijd oplosbaar is.

Een depot produceert nooit meer dan er aan throughput mogelijk is, dus kan alle productie ook vervoerd worden naar de klant of het depot dat het nodig heeft. Dus als het RP productie toegelaten is, dan is het RP oplosbaar.

2.3.6 Voorbeeld (vervolg)

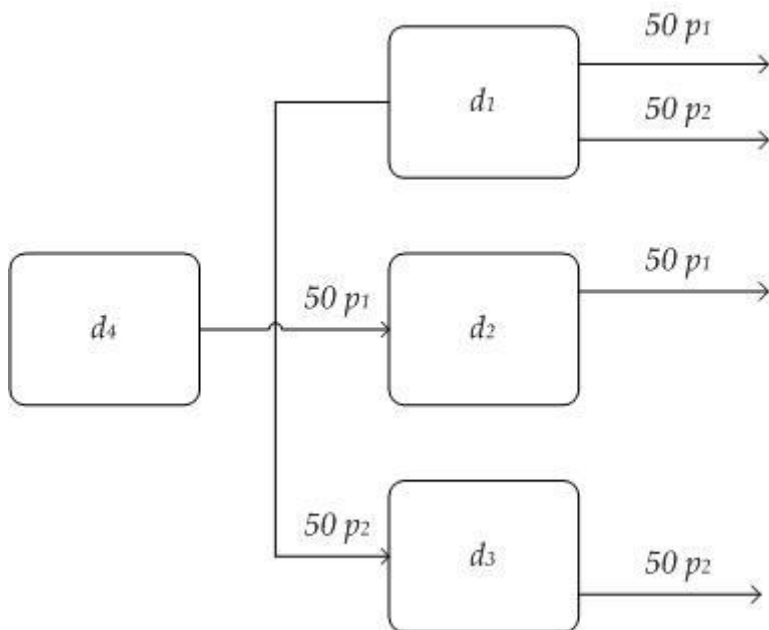
Het enige depot in het voorbeeld dan zowel kan produceren als distribueren is d_1 . Voor dit depot kunnen op basis van het bovenstaande nieuwe grenzen voor de productie bepaald worden.

Dit geeft als maximale productie van p_1 :

$$\max [t_{d_1 p_1}^{max}, t_{d_1}^{max} - t_{d_1 p_2}^S] = \max [100, 150 - 50] = 100$$

Er is dus geen nieuwe bovengrens. Voor p_2 blijft de bovengrens ook gelijk aan 100. Bij deze toewijzing zijn er dus geen veranderingen in de bovengrenzen van de productie. De instantie was productie toegelaten. De toewijzing heeft geen nieuwe grenzen op de productie gelegd. Het RP is dus productie toegelaten en dus oplosbaar.

Een grafische voorstelling van de oplossing voor het RP in het voorbeeld staat in figuur 2.



Figuur 2: een grafische weergave van een oplossing voor het primair transport in het voorbeeld

In deze oplossing vindt er primair transport plaats van d_1 naar d_3 ($50 p_2$) en tussen d_4 naar d_2 ($50 p_1$). Uit alle transportgegevens is nu op te maken wat de productiehoeveelheden zijn bij depot d_1 en depot d_4 . Deze productie is gelijk aan het verschil van de throughput en het binnenkomende transport. In het voorbeeld hebben depot d_1 en depot d_4 geen binnenkomend transport. Ze produceren dus precies hun throughput. De producties staan in tabel 3.

Depot	d_1	d_4
Productie van p_1	50	50
Productie van p_2	100	0

Tabel 3: de producties van de productiedepots uit het voorbeeld

2.3.7 Oplosbaarheid van het RP

Zoals genoemd in sectie 2.4.5 is een productie toegelaten RP, dat is ontstaan na een toegelaten toewijzing van het secundair transport, voldoende voor oplosbaarheid van de hele instantie.

Verderop zal aan de hand van een voorbeeld toegelicht worden dat de oplosbaarheid van een instantie niet betekent dat iedere toewijzing van het secundaire transport een RP oplevert dat oplosbaar is.

Er zijn echter condities waaronder iedere oplossing voor het secundaire transport wel een RP oplevert dat productie toegelaten is.

Conditie I: De instantie is productie toegelaten en er zijn geen depots die zowel kunnen distribueren als produceren.

Conditie I is een voldoende voorwaarde voor een productie toegelaten RP na een willekeurige oplossing van het secundaire transport. De toewijzing van het secundaire transport heeft dan namelijk geen invloed op de bovengrenzen van de productie.

Conditie I kan nog worden verzwakt tot conditie II.

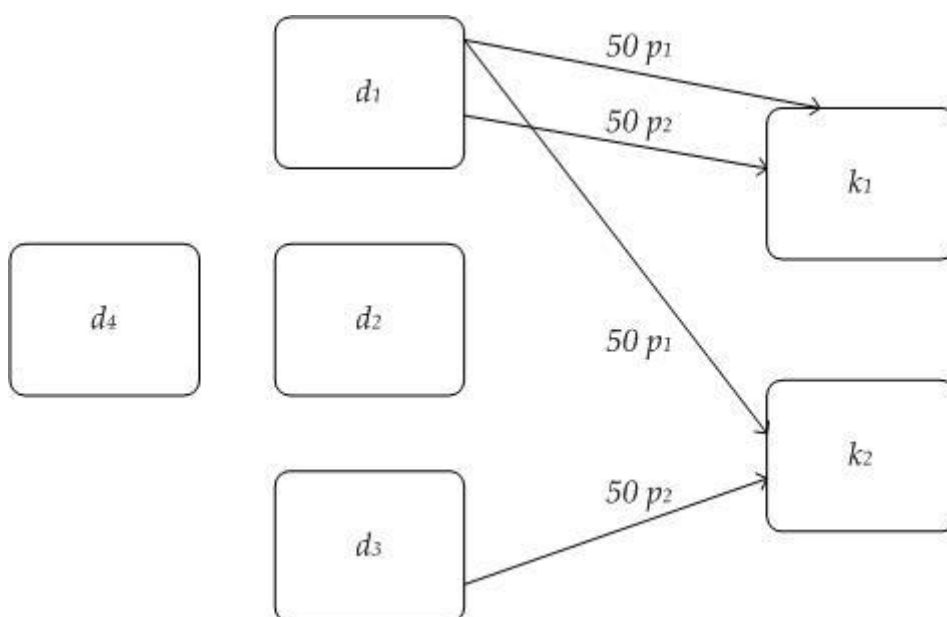
Conditie II: De instantie is productie toegelaten en de verzameling van productiedepots, die niet kunnen distribueren, is in staat om voor ieder product de totale vraag te produceren.

Ook als conditie II geldt, dan is het zeker dat een oplossing van het secundaire transport de bovengrenzen voor de productie niet zo kan verlagen dat het RP onoplosbaar wordt.

Conditie II is geen noodzakelijke voorwaarde, er bestaan immers instanties die niet aan conditie II voldoen, maar waarbij ook iedere toewijzing tot een productie toegelaten RP leidt. Echter is het niet duidelijk hoe eenvoudig te controleren condities voor noodzakelijke voorwaarden eruit zouden zien. Conditie I en Conditie II zijn in het algemene geval goed te controleren door een LP op te lossen.

2.3.8 Voorbeeld (vervolg)

Bij de eerder gekozen toewijzing blijft het RP productie toegelaten. De vraag is of er ook een toewijzing bestaat die zorgt dat het RP niet meer productie toegelaten is. Hieronder staat een dergelijke toewijzing



Figuur 3: een alternatieve toewijzing

De gegevens van het bijbehorende RP staan in tabel 4.

Depot	d_1	d_2	d_3	d_4
Productie	p_1, p_2			p_1
Overslag				
Distributie	p_1, p_2	p_1	p_2	
$t_{p_1}^{max}$	100	100	0	100
$t_{p_2}^{max}$	100	0	100	0
t^{max}	150	100	100	100
$t_{d_1}^S$	100	0	0	0
$t_{d_2}^S$	50	0	50	0
t_d^S	150	0	0	0

Tabel 4: gegevens voor het RP na de alternatieve toewijzing

De bovengrenzen voor de productie van p_2 worden nu anders dan bij de eerdere toewijzing.

De maximale productie van p_2 is nu gelijk aan:

$$\max [t_{d_1 p_2}^{max}, t_{d_1}^{max} - t_{d_1 p_1}^S] = \max [100, 150 - 100] = 50$$

Depot d_1 is het enige depot dat product p_2 kan produceren. Er wordt 100 gevraagd door beide klanten samen, maar er kan maar 50 worden geproduceerd. Het RP is nu niet meer productie toegelaten en dus niet oplosbaar.

Er bestaan aanpassingen aan de instantie die zorgen dat de instantie aan één van de genoemde condities zou voldoen.

- Indien depot d_1 geen enkel product meer zou kunnen distribueren, dan zou de instantie aan conditie I voldoen.
- Indien depot d_4 beide producten zou kunnen produceren, een maximale throughput van minimaal 100 zou hebben voor beide producten en in staat zou zijn tot een totale throughput van minimaal 200, dan zou de instantie voldoen aan conditie II.
- Indien depot d_1 product p_1 niet meer zou kunnen distribueren, dan zou de instantie niet aan conditie II voldoen, maar bij iedere oplossing van het secundaire transport zou het RP wel productie toegelaten zijn.

2.3.9 Kosten

Het vinden van oplossingen bij instanties die aan conditie II voldoen kan in twee stappen. Zodra het secundair transport toegewezen is, kan daarbij de beste (goedkoopste) oplossing van het primaire transport worden gezocht. Een instantie zou opgelost kunnen worden door eerst het secundaire transport op te lossen en een zo goedkoop mogelijke oplossing te zoeken, zonder daarbij te kijken naar het primaire transport. Voor deze oplossing, kan het daarbij behorende RP opgelost worden. Deze strategie zal in het algemeen niet tot optimale oplossingen leiden voor

de instantie. Indien bij het secundair transport een depot wordt gebruikt dat bij het primaire transport juist erg duur is, zal voor het totale probleem een betere oplossing kunnen bestaan dan de samenvoeging van de optimale oplossingen van de twee deelproblemen.

2.3.10 Samenvatting

Het BTP kan worden opgedeeld in twee stukken: het secundaire transport en het primaire transport. Het optimaliseren van het secundaire transport is NP-hard en is tevens te formuleren als een instantie van het MRGAP.

Bij het vastleggen van het secundaire transport blijft het Resultierend Probleem over. Dit probleem is te schrijven als Linear Program. De oplosbaarheid hiervan is vooraf onder bepaalde condities te garanderen, maar in het algemeen hangt de oplosbaarheid af van het secundaire transport.

De kosten van de optimale oplossing van het RP hangen ook af van het secundaire transport.

3 Oplossingsmethode

In dit hoofdstuk formuleren we hoe we het beschreven BTP uit hoofdstuk 2 gaan aanpakken. Hierbij maken we onder andere gebruik van de local search methode simulated annealing. Deze methode gebruiken we voor het secundaire transport. We hebben gezien dat het RP als een LP geschreven kan worden. Dit LP lossen we op met behulp van CPLEX[11]. Aan het einde van het vorig hoofdstuk zagen we al dat de oplossing van het secundaire transport invloed heeft op de oplossing van het RP. Met deze invloed wordt in het gebruikte algoritme rekening gehouden. Voordat we gaan starten met het uitleggen van het algoritme, gaan we het probleem eerst nog herformuleren.

3.1 Herformulering van het BTP

Voor we de oplossingsmethode beschrijven, gaan we het BTP eerst herformuleren. De instanties die we kunnen oplossen zijn onveranderd, maar na de probleemanalyse blijkt dat het formuleren van het probleem eenvoudiger kan.

3.1.1 Veranderingen

De veranderingen in de formulering zitten op drie gebieden. Bij de throughputvoorwaarden wordt er nu voor gekozen om aan te nemen dat er alleen depot-product throughputvoorwaarden zijn. Bij het toewijzen wordt alleen nog naar klant-productsets gekeken en niet meer naar klanten. Tot slot wordt bij depots geen onderscheid meer gemaakt in de mogelijke functies van het depot.

Throughputvoorwaarden

In het vorige hoofdstuk is aan bod gekomen dat het secundaire transport van het BTP omgeschreven kan worden naar het MRGAP. Hierbij kunnen voorwaarden voor de throughput van depot-productsets en voorwaarden voor de totale throughput van een depot worden omgeschreven naar voorwaarden van een product bij een depot.

Vanwege de beschreven manier van het omschrijven van dit soort voorwaarden richt het algoritme in dit hoofdstuk zich enkel op instanties waarin voorwaarden voor de throughput van een depot-productset of voorwaarden voor totale throughput niet bestaan.

Klant-productsets en klanten

Bij het oplossen van het secundaire transport, is het bestaan van klanten eigenlijk niet relevant. Het is voldoende om te weten dat er klant-productsets zijn die toegewezen moeten worden aan een distributiedepot. Voor iedere mogelijke toewijzing kunnen vooraf de kosten voor het transport van alle gevraagde hoeveelheden in een klant-productset worden berekend. De gevolgen voor de kosten van het primaire transport van een toewijzing zijn vooraf niet exact te berekenen, want het is immers het geheel aan toewijzingen dat de kosten van het primaire transport beïnvloedt.

Functies van depots

Tot zover is onderscheid gemaakt in depots wat betreft hun functie(s) om aan te geven welke vormen van transport wel en niet toegestaan zijn. Hetzelfde resultaat is te bereiken door voor verboden transportlijnen de kosten oneindig groot te maken. Verboden productie wordt opgelost door een extra parameter die voor ieder depot de maximale productie van een product aangeeft. In dit hoofdstuk zal verder alleen nog over depots worden gesproken. Het onderscheid in functies wordt gemaakt door de productieparameters en kostenparameters van de depots.

3.1.2 Nieuwe formulering

Zo komen we nu tot de onderstaande formulering van het BTP.

Verzamelingen en parameters

- De verzameling producten noemen we P .
- De verzameling klant-productsets noemen we Q .
- De verzameling depots noemen we D .
- Voor iedere klant $q \in Q$ is bekend wat de gevraagde hoeveelheid $v_{qp} \geq 0$ is van een product $p \in P$.
- Voor ieder depot $d \in D$ is bekend wat de maximale throughput t_{dp}^{max} is van product $p \in P$.
- Voor ieder depot $d \in D$ is bekend wat de maximale productie u_{dp}^{max} is van product $p \in P$.
- Voor iedere toewijzing van een klant-productset $q \in Q$ aan depot $d \in D$ is bekend wat de kosten c_{dq}^S van het bijbehorende secundaire transport zijn.
- Voor ieder tweetal depots $d, e \in D$ is bekend wat de kosten c_{dep}^P zijn van het transport van één eenheid van product $p \in P$.
- Voor ieder depot $d \in D$ is bekend wat de kosten c_{dp}^{PR} zijn voor het produceren van één eenheid van product $p \in P$.

Beslissingsvariabelen

- De variabele $x_{qd} \in \{0,1\} = 1$ als klant-productset $q \in Q$ wordt toegewezen aan depot $d \in D$ en 0 anders.

- De variabele $z_{dp} \in [0, \infty)$ is gelijk aan het aantal eenheden van product $p \in P$ die bij depot $d \in D$ geproduceerd wordt.
- De variabele y_{dep} is gelijk aan het aantal eenheden van product $p \in P$ van depot $d \in D$ naar depot $e \in D$ getransporteerd wordt.

Voorwaarden

Voor de throughput gebruiken we weer de eerder ingevoerde notatie.

Met t_{dp}^- wordt al het transport van product $p \in P$ dat binnenkomt bij depot $d \in D$ aangeduid.

$$t_{dp}^- = \sum_{e \in D} y_{edp} \quad (p \in P, d \in D)$$

Met t_{dp}^+ wordt al het transport van product $p \in P$ dat vertrekt bij depot $d \in D$ aangeduid, de throughput dus. Dit is de som van het transport naar andere depots en de vraag van de klant-productsets.

$$t_{dp}^+ = \sum_{e \in D} y_{dep} + \sum_{q \in Q} v_{qp} x_{qd} \quad (p \in P, d \in D)$$

De transportvoorwaarden zijn:

$$t_{dp}^- + z_{dp} = t_{dp}^+ \quad (\forall p \in P, \forall d \in D)$$

De toewijzingsvoorwaarden zijn:

$$\sum_{d \in D} x_{qd} = 1 \quad (q \in Q)$$

De maximum-throughputvoorwaarden zijn:

$$t_{dp}^+ \leq t_{dp}^{\max} \quad (\forall d \in D, p \in P)$$

De maximum-productievoorwaarden zijn:

$$z_{dp} \leq u_{dp}^{\max} \quad (\forall d \in D, p \in P)$$

De doelfunctie

De kosten zijn opgebouwd uit kosten voor productie, primair transport en secundair transport. De kosten willen we nog steeds minimaliseren.

$$\text{Kosten} = \sum_{d \in D} \sum_{p \in P} c_{dp}^{PR} z_{dp} + \sum_{d \in D} \sum_{e \in D} \sum_{p \in P} c_{dep}^p y_{dep} + \sum_{d \in D} \sum_{q \in Q} c_{qd}^S x_{qd}$$

In het vervolg van dit hoofdstuk zullen we gebruik maken van deze formulering van het BTP.

3.2 Simulated Annealing

In de gekozen oplosmethode gebruiken we Simulated Annealing. We gaan nu eerst uitleggen wat Simulated Annealing is en lichten daarna toe waarom we hier gebruik van maken.

3.2.1 Local Search

Simulated Annealing is een local search methode. Local search methodes worden gebruikt voor het oplossen van optimaliseringsproblemen. De werking van een local search methode is als volgt.

1. Er wordt een startoplossing gemaakt.
2. Vanuit de huidige oplossing wordt een andere oplossing in de omgeving van de huidige oplossing (een buur) uitgezocht. Deze buur wordt de nieuwe huidige oplossing.
3. Indien de buur beter is dan de beste oplossing die tot zover gevonden was, wordt de beste oplossing vervangen door de nieuwe oplossing.
4. Indien niet voldaan is aan een stopcriterium, ga naar stap 2.

Local search methodes eindigen meestal in een lokaal optimum. Dat wil zeggen dat er geen oplossingen in de buuromgeving zijn die beter zijn.

Verschillende local search methodes onderscheiden zich vooral van elkaar in het selecteren van de nieuwe (buur) oplossing.

3.2.2 Het raamwerk van simulated annealing

Het simulated annealing algoritme heeft de volgende uitgangspunten[3]:

- Er is een verzameling S van toegelaten oplossingen.
- Er is een kostenfunctie $J: S \rightarrow \mathbb{R}$.
- Voor elke $i \in S$, bestaat er een verzameling $S(i) \subset S - \{i\}$. We noemen $S(i)$ de burens van i .
- Voor elke i is er een serie positieve kansen $q_{ij}, j \in S(i)$ z.d.d. $\sum_{j \in S(i)} q_{ij} = 1$.
- Er is een niet-stijgende functie $T: \mathbb{N} \rightarrow \langle 0, \infty \rangle$. $T(t)$ noemen we de temperatuur tijdens iteratie t .
- Er is een startoplossing $x(0) \in S$.

Simulated Annealing is een iteratief algoritme, waarin het volgende uitgevoerd wordt, waarbij $i = x(t)$, de oplossing aan het begin van iteratie t is:

Op een tijdstip t in toestand $x(t)$ wordt het volgende uitgevoerd:

We noemen $x(t)$ voor makkelijkere notatie i .

1. Er wordt een buur $j \in S(i)$ uitgekozen. De kans dat $j \in S(i)$ gekozen wordt is q_{ij} .
2. Als $J(j) \leq J(i)$ dan $x(t + 1) = j$
Als $J(j) > J(i)$ dan $x(t + 1) = j$ met kans $e^{(J(i)-J(j))/T(t)}$ en anders $x(t + 1) = i$.

De stappen 1 en 2 worden herhaald totdat een stoptemperatuur bereikt is.

Tussendoor wordt onthouden wat de beste oplossing is.

Simulated Annealing heeft als voordeel dat in een iteratie maar één buur bekeken wordt. Dit zorgt ervoor dat de rekentijd van een iteratie kort is. Indien er weinig geschikte burens zijn, kan het nog steeds zo zijn dat het veel iteraties duurt voordat een oplossing i vervangen wordt.

3.3 Raamwerk van het algoritme

We gaan gebruik maken van Simulated Annealing (SA). Dit betekent dat we een aantal functies van SA zullen moeten invullen.

1. Het maken van een startoplossing
2. Het creëren van een buur vanuit een oplossing
3. Een kostenfunctie
4. Een Temperatuurfunctie

Het creëren van een buur komt in sectie 3.4 aan bod. Sectie 3.5 behandelt de kostenfunctie. Het maken van de startoplossing en de temperatuurfunctie komen hieronder aan bod.

3.3.1 Startoplossing

Voor het maken van een startoplossing wordt voor iedere klant-productset het depot van de goedkoopste toewijzing gekozen. Iedere klant-productset is dan toegewezen aan een distributiedepot.

Naar andere voorwaarden wordt nog niet gekeken. De toewijzing van de klant-productsets zal slechts in zeldzame gevallen toegelaten zijn. Het primaire transport wordt bij de startoplossing dan ook niet berekend, omdat we in het algemeen niet met een toegelaten oplossing voor het secundair transport te maken hebben.

3.3.2 Temperatuurfunctie

Voor de temperatuurfunctie is gekozen voor exponentiële afname. Na een iteratie wordt de temperatuur vermenigvuldigd met een vaste factor $(1 - \varepsilon)$, $0 < \varepsilon \ll 1$. Afhankelijk van de instantie kunnen starttemperatuur en ε gekozen worden. Dit heeft als resultaat dat:

$$T(t) = T_{start} \cdot (1 - \varepsilon)^t$$

3.4 Het creëren van een buur

Bij het creëren van een buur wordt in de eerste plaats gekeken naar het secundair transport. Zoals inmiddels bekend is dit te modelleren als instantie van het MRGAP. In de literatuur wordt MRGAP door Yagiura [2], [3] opgelost door middel van een local search methode. Zij gebruiken drie soorten buurtomgevingen. Deze heten de verplaats-omgeving, de verwissel-omgeving en de keten-omgeving.

Waar Yagiura [2], [3] zich richt op MRGAP en de bijbehorende termen taak, agent e.d. gebruikt, zullen we hier blijven praten over klant-productset en depot. Zo is duidelijker hoe de buurtomgevingen er voor ons probleem uitzien. Een oplossing geven we aan met σ . De buurtomgeving of omgeving van σ geven we aan als $N(\sigma)$

Verplaats-omgeving en verwisselomgeving

$$N_{verplaats}(\sigma) = \{\sigma' \mid \sigma' \text{ wordt verkregen uit } \sigma \text{ door één klant} \\ \text{– productset aan een nieuw depot toe te wijzen}\}$$

$$N_{verwissel}(\sigma) = \{\sigma' \mid \sigma' \text{ wordt verkregen uit } \sigma \text{ door twee klant} \\ \text{– productsets onderling van depot te laten wisselen}\}$$

Als n het aantal klant-productsets in Q is en m het aantal depots in D , dan is de grootte van de verplaatsomgeving is $O(nm)$ en de grootte van de verwisselomgeving $O(n^2)$. Vaak geldt dat $n > m$, dus ook $|N_{verplaats}| < |N_{verwissel}|$.

De keten-omgeving bestaat uit oplossingen die verkregen worden door tegelijk een aantal verplaatsingen uit te voeren. Met $\sigma(q)$, $q \in Q$, geven we het depot $d \in D$ aan waar de klant-productset q aan toegewezen is bij oplossing σ .

$N_{keten}(\sigma)$ is de verzameling van oplossingen die ontstaan door klant-productsets q_1, q_2, \dots, q_l te verplaatsen zodanig dat

$$\sigma'(q_r) = \sigma'(q_{r-1}), r = 2, 3, \dots, l \\ \sigma'(q_1) = \sigma'(q_l)$$

Hierbij kan l de waarden $2, 3, 4, \dots, n$ aannemen. De l verplaatsingen zijn dus zo dat er een kring ontstaat. De keten-omgeving is erg groot, d.w.z. exponentieel in l .

3.5 De Kostenfunctie

Bij het opstellen van de kostenfunctie spelen twee aspecten die extra aandacht nodig hebben. Als eerste moet er worden bepaald hoe we kosten toekennen aan oplossingen die niet toegelaten zijn. Hier gaan we penaltykosten voor invoeren.

Als tweede gaan we de kosten van het primaire transport een rol laten spelen in het algoritme dat het secundaire transport bepaalt. Voor dit tweede punt gaan we het algoritme dat hiervoor gemaakt is, stap voor stap behandelen.

3.5.1 Penaltykosten

Bij de manier van local search die we gaan toepassen, zal de toestand (huidige oplossing) meer dan eens niet toegelaten zijn. Dit betekent dat één of meerdere van de throughputvoorwaarden wordt geschonden doordat één of meerdere depots te veel klant-productsets moeten toeleveren.

Het algoritme zal dus ook kosten moeten toekennen aan oplossingen die niet toegelaten zijn. Alleen de gewone doelfunctie gebruiken, zal er waarschijnlijk toe leiden dat de beste oplossing volgens het algoritme een oplossing is, die niet toegelaten is. We zullen dus moeten zorgen dat oplossingen die niet toegelaten zijn, duurder worden. Hiervoor gebruiken zogeheten penaltykosten, net zo als in [2].

De grootte van penaltykosten hangt af van de grootte van de overschrijding(en) van de throughputvoorwaarden. Naarmate het algoritme meer tussenoplossingen genereert die niet toegelaten zijn, zullen de kosten voor overschrijdingen ook nog eens duurder worden.

De formules voor de penaltykosten zien er als volgt uit:

$$\text{penaltykosten}(\sigma) = \sum_{\substack{d \in D \\ p \in P}} w_{dp} O_{dp}$$

waarbij:

$$O_{dp} = \max \left\{ 0, \sum_{q \in Q} v_{qp} x_{qd} - t_{dp}^{\max} \right\}, d \in D, p \in P.$$

De overschrijding van de throughputvoorwaarde van product $p \in P$ bij depot $d \in D$ is gelijk aan O_{dp} . Met gewicht w_{dp} wordt bepaald hoe zwaar een overschrijding van de throughputvoorwaarde voor product $p \in P$ bij depot $d \in D$ wordt meegewogen bij het berekenen van de totale penaltykosten. Gedurende het uitvoeren van het algoritme zullen de gewichten steeds aangepast worden.

Voor niet-toegelaten oplossingen worden de kosten berekend als de som van de kosten van de doelfunctie en de penaltykosten. Merk op dat voor toegelaten oplossingen de penaltykosten gelijk zijn aan nul.

3.5.2 Schatting van primaire transport

Na een toewijzing van het secundaire transport wordt met een LP-solver het Resultierend Probleem (RP) opgelost. Een toewijzing kan binnen het secundaire transport heel goedkoop zijn, maar bij het primaire transport uiteindelijk toch heel duur worden.

Om de kosten van het primaire transport niet uit het oog te verliezen tijdens het plannen van het secundaire transport gaan we SA niet gebruiken met de gewone kostenfunctie. De kosten van een toewijzing gaan we verhogen met de minimale kosten voor primair transport die erbij zullen komen, als die toewijzing wordt gekozen. Als we dit willen doen, zullen we eerst bij ieder distributiedepot voor ieder product moeten berekenen wat de minimale kosten zijn om één eenheid van dat product bij dat depot te krijgen.

Voor het berekenen van de verhoging van de kosten gebruiken we het algoritme kostenverhoging (KV). In dit algoritme moet geïdentificeerd worden of depots kunnen produceren. Dit gebeurt aan de hand van de maximum-productieparameters u_{dp}^{max} .

3.5.3 Het KV-algoritme

Het KV-algoritme berekent eerst voor iedere combinatie (d, p) , $d \in D_p^D$, $p \in P$ wat de minimale kosten k_{dp} zijn om één eenheid van product p bij depot d te krijgen.

Daarna worden er met behulp van de berekende parameters k_{dp} nieuwe kostenparameters voor het secundair transport berekend.

Voor het beschrijven van het eerste deel, laat $d^* \in D_p^D$ respectievelijk $p^* \in P$ het depot en het product zijn waarvoor we de minimale kosten berekenen. De verzameling van depots die product p^* kunnen produceren, duiden we aan met $D_{p^*}^P$.

Het KV algoritme lijkt veel op Dijkstra's algoritme. We gaan eerst het idee van het algoritme beschrijven en daarna volgt de pseudocode.

We houden een verzameling $S \subseteq D$ bij. Deze start als $S = \{d^*\}$. Verder wordt voor ieder depot uit $d \in D \setminus S$ bijgehouden wat de waarde is van de goedkoopste manier van transport $tkost(d)$ voor product p^* van depot d naar depot d^* , waarbij alleen overslag mag plaatsvinden op depots uit S . In het begin van het algoritme geldt:

$$tkost(d) = c_{dd^*p^*}^P, d \in D.$$

Tijdens iedere iteratie wordt telkens het depot $d^\wedge \in D \setminus S$ dat de laagste $tkost(d^\wedge)$ heeft toegevoegd aan S . Na het toevoegen van depot d^\wedge aan S wordt voor alle depots $d \in D \setminus S$ de waarde van $tkost(d)$ bijgewerkt.

$$tkost(d) = \min (tkost(d), tkost(d^\wedge) + c_{dd^\wedge p^*}^P)$$

We stoppen met toevoegen van depots aan S , indien alle productiedepots voor product p^* in S zitten, dus op het moment dat $D_{p^*}^P \subseteq S$.

Nadat we gestopt zijn bepalen we de goedkoopste manier om product p^* bij depot d^* te krijgen als het minimum van de som van de productiekosten en de transportkosten, waarbij we alleen de productiedepots voor product p^* hoeven af te gaan.

$$k_{d^* p^*} = \min\{tkost(d) + c_{dp^*}^{PR} \mid d \in D_{p^*}^P\}$$

Als we voor ieder product $p \in P$ voor ieder distributiedepot $d \in D_p^D$ de waarde van k_{dp} bepaald hebben gaan we door naar deel 2 van het algoritme. Daarin gaan we de nieuwe kostenparameters voor het secundair transport opstellen. De kosten van iedere toewijzing worden gelijk aan de kosten van het secundair transport met daarbij opgeteld de kosten die het primair transport minimaal met zich meebrengt.

$$c_{qd}^S(alg) = c_{qd}^S + \sum_{p \in P} v_{qp} k_{dp}, q \in Q, d \in D$$

Dit gebeurt alleen met toewijzingen waarvan de kosten c_{qd}^S eindig zijn.

Het KV Algoritme in Pseudocode

```
For all  $p^* \in P$  do
  For all  $d^* \in D_p^D$  do
     $S = \{d^*\}$ 
     $tkost(d^*) = 0$ 
    Determine  $D_{p^*}^P$ 
    For all  $d \in D \setminus d^*$  do
       $tkost(d) = c_{dd^*p^*}^P$ 
    End for
    While  $D_{p^*}^P \not\subseteq S$  do
      Select  $d^\wedge$  such that  $tkost(d) \geq tkost(d^\wedge) \forall d \in D \setminus S$ 
       $S := S \cup \{d^\wedge\}$ 
      For all  $d \in D \setminus S$  do
         $tkost(d) = \min(tkost(d), tkost(d^\wedge) + c_{dd^\wedge p^*}^P)$ 
      End For
    End While
     $k_{d^*p^*} = \min\{tkost(d) + c_{dp^*}^{PR} \mid d \in D_{p^*}^P\}$ 
  End For
End For

For all  $q \in Q$  do
  For all  $d \in D$  do
    If  $c_{qd}^S \neq \infty$  then
       $c_{qd}^S(\text{alg}) := c_{qd}^S + \sum_{p \in P} v_{qp} k_{dp}$ ,  $q \in Q, d \in D$ 
    Else
       $c_{qd}^S(\text{alg}) := \infty$ 
    End If
  End for
End for
```

3.6 Het gehele algoritme

Met behulp van de methodes die nu behandeld zijn, kunnen we het algoritme als geheel gaan bekijken. Voordat kan worden begonnen met SA moeten we eerst een aantal procedures aangaan. Het hierboven behandelde KV algoritme is daar één van. Tijdens SA wordt ook het primair transport voor meerdere oplossingen van het secundair transport opgelost.

3.6.1 Vooraf

Voordat we starten met SA worden de volgende parameters ingevuld.

- Starttemperatuur van SA
- De waarde van de vermenigvuldigingsfactor na een iteratie van SA
- De stoptemperatuur
- De startgewichten van de throughputvoorwaarden

- De startoplossing
- De waarden van $c_{qd}^S(alg)$

3.6.3 Het oplossen van het resulterend probleem

Voor instanties waar het secundaire transport toegelaten is, wordt berekend wat het optimale primaire transport is. Dit wordt gedaan door het bijbehorende LP door CPLEX te laten doorrekenen. Als het primaire transport opgelost kan worden, dan is de oplossing toegelaten voor de hele instantie van het BTP. Van die oplossing worden dan de kosten berekend. Het berekenen van deze kosten gebeurt met de originele kostenparameters. Alleen oplossingen waarvan de echte kosten berekend worden, worden vergeleken met de beste oplossing.

Een iteratie

Een iteratie van het aangepaste SA algoritme ziet er zo uit:

Bij de volgende beschrijving van een iteratie van het aangepaste SA algoritme bedoelen we met *kosten* bedoelen we de kosten zoals die in de doelfunctie van het BTP geformuleerd zijn en met *algkosten* bedoelen we de kosten van secundair transport berekend met de kostenparameters $c_{qd}^S(alg)$ met daarbij opgeteld de penaltykosten.

Iteratie t met de huidige oplossing σ en beste oplossing σ_{best} ziet er als volgt uit. Bij vervanging van de huidige oplossing start de *vervangingsprocedure*. Deze staat toegelicht onder de iteratie.

1. Creëer een buur σ'
De buur bestaat alleen uit een toewijzing van het secundair transport. In de huidige versie van de implementatie zijn alleen de verplaatsomgeving en de verwisselomgeving gerealiseerd. Er wordt random bepaald welke omgeving gebruikt wordt in een iteratie. Beide omgevingen hebben een kans van 0,5 om gebruikt te worden.
2. Vergelijk de kosten van de buur met die van de huidige oplossing. Gebruik bij deze berekening de kostenparameters $c_{qd}^S(alg)$
 - Als de buur goedkoper is, vervang de huidige oplossing door de buur. Start de *vervangingsprocedure*.
 - Als de buur duurder is, vervang de huidige oplossing door de buur met kans $e^{((algkosten(\sigma') - alkosten(\sigma))/T(t))}$
Bij vervanging: start de *vervangingsprocedure*.
3. Verlaag de temperatuur.
4. Indien $T(t + 1) > T_{stop}$ ga naar stap 1.

De vervangingsprocedure, die wordt aangeroepen bij wijziging van de huidige oplossing ziet er als volgt uit:

1. Controleer of oplossing σ' toegelaten is, wat betreft het secundair transport.
Zo nee: ga naar stap 6.
2. Stel het RP op en los het op met behulp van CPLEX.
Als CPLEX geen toegelaten oplossing vindt, ga naar stap 7.
3. Vul de y -variabelen en z -variabelen van oplossing σ' met de door CPLEX gevonden waarden.
4. Indien $kosten(\sigma') < kosten(\sigma_{best})$
 $\sigma_{best} := \sigma'$
5. Verlaag alle gewichten:
 $\forall d \in D, \forall p \in P : w_{dp} := \frac{w_{dp}}{1.5}$
Ga naar stap 7.
6. Verhoog alle gewichten:
 $\forall d \in D, \forall p \in P : w_{dp} := 2 \cdot w_{dp}$
7. EINDE

We zien dat het accepteren van een buur, waarvan het secundair transport niet toegelaten is, zorgt dat de gewichten voor de overschrijdingen van de throughputvoorwaarden worden verhoogd. Bij het accepteren van een buur waarbij het secundair transport wel toegelaten is, worden deze gewichten echter verlaagd. Om te zorgen dat er voldoende toegelaten oplossingen worden gevonden, gaat het verhogen met een grotere factor dan het verlagen.

Het berekenen van het primair transport gebeurt voor iedere oplossing, waarvan het secundaire transport toegelaten is. Zo wordt niet onnodig vaak een LP opgelost, iets dat veel extra rekentijd zou vragen.

4 Resultaten

In dit hoofdstuk gaan we het ontwikkelde algoritme testen. De testinstanties worden verkregen met behulp van een zelf ontwikkelde generator. Hier is voor gekozen, omdat het voor ons niet mogelijk was om aan geschikte instanties uit de werkelijkheid te komen. Het eerste deel van dit hoofdstuk behandelt hoe het genereren van de testinstanties in zijn werk gaat. In het tweede deel volgen de resultaten.

4.1 Instanties genereren

Bij het maken van een instantie beginnen bij het model van hoofdstuk 2. Er zijn producten, klanten, klant-productsets, depots en depot-productsets.

Het algoritme dat we hebben ontworpen gaat ervan uit dat voorwaarden voor depot-productsets al zijn weggewerkt met behulp van de methode uit sectie 2.2.5. Die methode gaan we dus toepassen doen binnen het ontwerpen van een instantie. Dit zal dus extra producten opleveren, die geen extra kosten met zich meebrengen.

Verder hebben we gezien dat ons algoritme geen gebruik maakt van locaties, maar alleen van de resulterende kosten. Het is dus niet van groot belang dat er klant-productsets zijn met dezelfde locatie. Van de structuur van klanten en klant-productsets zal worden afgeweken. De klant-productsets zullen rechtstreeks worden gegenereerd.

We zullen laten zien hoe een testinstantie wordt opgebouwd en welke parameters een rol spelen.

4.1.1 Random getallen

Bij het genereren van een instantie wordt op veel plaatsen gebruik gemaakt van random getallen. Soms is dat een geheel getal, soms ook een decimaal getal.

Het maken van een willekeurig getal gebeurt overal uniform verdeeld.

Hieronder bedoelen we met $\text{randInt}(a; b)$, $a < b$ en $a, b \in \mathbb{N}$ een random geheel getal uit de verzameling $\{a, a + 1, \dots, b\}$ en met $\text{rand}(a; b)$, $a < b$ en $a, b \in \mathbb{R}$ een random decimaal getal uit het interval $[a, b]$.

4.1.2 Gebied

Voor het kostenmodel spelen locaties en afstanden een rol. Locaties bestaan uit een x-coördinaat en een y-coördinaat. De minimale waarde van zo'n coördinaat is altijd 0.

De maximale waarden die de coördinaten kunnen aannemen, $MaxX$ en $MaxY$, kunnen worden ingevoerd.

Voor afstanden in het gebied gebruiken we de Euclidische afstand. Dus de afstand $afs(a, b)$ tussen punt $a(x_a, y_a)$ en $b(x_b, y_b)$ is gelijk aan $\sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}$.

4.1.3 Producten

Het aantal verschillende producten wordt aangegeven door de parameter AP .

Voor een product $p \in P$ worden twee soorten kosten willekeurig bepaald:

1. De productiekosten $KP(p) \in \{2, 3, 4, \dots, 10\}$
2. De transportkosten $KT(p) \in \{2, 3, 4, \dots, 10\}$

Die dienen later om de kostenparameters voor de instantie mee te berekenen.

4.1.4 Klant-productsets

Het aantal klant-productsets wordt aangegeven door de parameter AK .

Bij klant-productsets wordt opgegeven wat de kans $ProbV$ is dat een willekeurige klant-productset $q \in Q$ een strict positieve vraag heeft voor een willekeurig product $p \in P$. Verder worden voor een klant-productset q worden de volgende eigenschappen met een uniforme verdeling gegenereerd:

1. De x-coördinaat $XK \in \{2, 3, 4, \dots, MaxX\}$
2. De y-coördinaat $YK \in \{2, 3, 4, \dots, MaxY\}$
3. Voor iedere product $p \in P$ is er met kans $ProbV$ een positieve vraag v_{qp} . Indien de vraag positief is, dan is $v_{qp} = randInt(10,100)$.

4.1.5 Depots

Bij de depots worden de meeste eigenschappen gegenereerd. Behalve de locatie, moeten de ook de functies, de maximale producties en maximale throughputs gegenereerd worden. Hierbij moet er ook op gelet worden dat de instantie straks waarden van vraag en maximale throughput heeft die op elkaar zijn afgestemd.

Locatie

Het aantal depots wordt aangegeven door de parameter AD .

Bij depots worden de volgende eigenschappen met een uniforme verdeling gegenereerd:

1. De x-coördinaat $XD = randInt(0, MaxX)$
2. De y-coördinaat $YD = randInt(0, MaxY)$

Funcities

Voor de functies van depots moet worden ingevoerd welke fractie van de depots deze functie heeft. De fractie van de depots die kunnen distribueren, overslaan en produceren geven we aan met respectievelijk $ProbD$, $ProbO$, $ProbP$. Ieder van deze parameters moet groter dan nul en kleiner of gelijk aan één zijn. Ook moet gelden dat ze samen groter of gelijk aan één zijn, anders zouden er depots zonder functie gegenereerd worden. Als de drie parameters samen groter zijn dan één, dan zullen er depots ontstaan met meer dan één functie. Dat is toegestaan bij instanties.

Dus voor de parameters over de functies van depots moet gelden:

$$0 < ProbD \leq 1$$

$$0 < ProbO \leq 1$$

$$0 < ProbP \leq 1$$

$$ProbD + ProbO + ProbP \geq 1$$

Het bepalen van functies van depots gaat nu als volgt in zijn werk:

Het aantal depots kan distribueren, overslaan en produceren, noemen we respectievelijk ADD , ADO , ADP .

Er geldt:

$$ADD = ProbD \cdot AD$$

$$ADO = ProbO \cdot AD$$

$$ADP = ProbP \cdot AD$$

De waarden van ADD , ADO en ADP worden daarna afgerond op helen. In het model is overslag de functie die het vaakst met een andere functie gecombineerd wordt.

Daar gaan we bij de verdeling als volgt mee om:

1. Nummer de depots van 1, 2, ..., AD .
2. De eerste ADD depots krijgen de functie distributie.
3. De laatste ADP depots krijgen de functie productie.
4. De depots $ADD + 1$, $ADD + 2$, ..., $\min(AD, ADD + ADO)$ krijgen de functie overslag. Als dat nog niet voldoende overslagdepots oplevert ($ADD + ADO > AD$), dan krijgen ook een aantal distributiedepots de functie overslag erbij.

Totale vraag en speling

Voordat we bij depots waarden gaan genereren voor maximale throughput en maximale productie, gaan we eerst per product $p \in P$ berekenen wat de totale vraag v_p is.

$$v_p = \sum_{q \in Q} v_{qp}$$

Omdat de klant-productsets ieder geheel van één depot moeten komen, zal voor oplosbaarheid over het algemeen een overschot aan maximale throughput aanwezig moeten zijn bij de distributiedepots. Bij een situatie met throughputvoorwaarden voor depot/productsets of voor depots, zal ook bij productiedepots nodig zijn. Omdat overslag geen noodzakelijke stap is voor ieder product, heeft de totale capaciteit geen invloed op de oplosbaarheid van een instantie.

Voor het hierboven omschreven overschot is de parameter *Speling* bedoeld. Hoe groter de speling, des te makkelijker in het algemeen een toegelaten oplossing te vinden zal zijn.

$$Speling \in \langle 1, \infty]$$

Voor ieder product $p \in P$ wordt de totale productiecapaciteit $MaxP(p)$ gelijk gesteld aan de vraag naar het product vermenigvuldigd met de *Speling*.

$$MaxP(p) = Speling \cdot v_p$$

Voor de totale distributiecapaciteit $MaxD(p)$ gebeurt hetzelfde.

$$MaxD(p) = Speling \cdot v_p$$

Voor de totale overslag $MaxO(p)$ wordt extra ruimte gegeven, zodat dubbele overslag op grotere schaal mogelijk wordt. Het verhogen van de overslagcapaciteit zal geen invloed hebben op de oplosbaarheid van het probleem. De manier waarop de capaciteit verdeeld wordt, zal daar voor zorgen.

$$MaxO(p) = (Speling)^2 \cdot v_p$$

Productie

De verdeling van de productiecapaciteit wordt per product $p \in P$ afgehandeld.

We gaan de productiedepots één voor één af. We houden bij hoeveel capaciteit er nog verdeeld moet worden en hoeveel productiedepots we gehad hebben. We noemen dit *rest* respectievelijk *geweest*.

Bij het eerste depot geldt dus $rest=MaxP(p)$ en $geweest = 0$.

Bij ieder productiedepot d is de maximale productie van product $p \in P$

$$u_{dp}^{max} = randInt(0; 2 \cdot \lceil rest / (ADP - geweest) \rceil]$$

Op het moment dat het laatste productiedepot aan de beurt is, krijgt deze alles dat over is, zodat de *Speling* zo wordt, zoals ingesteld is.

Mocht het gebeuren dat het getrokken getal groter is dan *rest* dan wordt dit verminderd tot de waarde van *rest*. Depots die daarna nog aan bod komen krijgen dan een waarde van 0.

De keuze van deze verdeling leidt ertoe dat de verwachtingswaarde van de productiecapaciteit voor alle productiedepots ongeveer gelijk is.

De verdeling van de maximale throughput $MaxT(p)$ van de productiedepots wordt gelijk gesteld met de maximale productie.

$$t_{dp}^{max} = u_{dp}^{max}, d \in D_p^P, p \in P$$

Distributie

De verdeling van de maximale throughput van de distributiedepots gaat bijna analoog aan de verdeling van de maximale productie over de productiedepots.

We doen dit weer per product. We gebruiken weer de variabelen *rest* en *geweest*. Waarbij *rest* start als $MaxT(p)$ en *geweest* weer als 0.

Eerst behandelen we de depots die zowel kunnen distribueren als produceren. Bij die depots blijft de waarde van $MaxT(p)$ gelijk, maar bij die depots worden de waarden van *rest* en *geweest* al aangepast.

Na de depots met distributie- en productiefunctie gaan we de depots af die distributie en juist geen productie hebben. Hier wordt weer uniform verdeeld getrokken uit.

$$t_{dp}^{max} = randInt(0, 2 \cdot \lceil rest / (ADD - geweest) \rceil]$$

De afhandelingen om alle throughput precies op te maken zijn dezelfde als eerder.

Overslag

Bij maximale throughput ligt niet vast, waarvoor deze gebruikt mag worden. Een depot met meerdere functies kan deze allemaal uitvoeren, maar heeft feitelijk maar met één verzameling throughputvoorwaarden te maken. Voor het verdelen van de capaciteit van de overslag wordt eerst gekeken naar depots die naast overslag ook nog kunnen distribueren en/of produceren. Een depot dat al capaciteit toegewezen heeft gekregen vanwege zijn distributiefunctie nu ook capaciteit voor overslag geven, zorgt onder andere, dat zijn capaciteit voor distributie ook wordt verhoogd. Dit zou de instantie makkelijker oplosbaar maken, dan bij *Speling* is ingevoerd. Het verlagen

van de capaciteit zou tot onoplosbaarheid kunnen leiden. Dit staan we niet toe. Bij distributiedepots laten we de capaciteit daarom precies zoals deze was.

De capaciteit van de productie van veel kleiner belang voor de oplosbaarheid. Daarom laten we een verhoging van de capaciteit daar wel toe. Verlagen zouden voor onoplosbaarheid kunnen zorgen. Deze laten we niet toe. Deze depots krijgen het maximum toegewezen van wat ze al hadden en een willekeurig getal uit een soortgelijke uniforme verdeling zoals al twee keer eerder is opgebouwd.

De rest van de overslagcapaciteit wordt verdeeld zoals bij de rest van de capaciteitsverdelingen ook gebeurd is.

4.1.6 Kosten

We gaan nu bespreken hoe alle kostenparameters gegenereerd worden. Er moeten parameters worden gegenereerd voor de productie, voor het primaire transport en voor het secundaire transport.

Productie

Als model voor de productiekosten c_{dp}^{PR} is gekozen om bij ieder productiedepot $d \in D_p^P$ voor ieder product $p \in P$ de productiekosten $KP(p)$ te vermenigvuldigen met een random getal.

$$c_{dp}^{PR} = KP(p) \cdot rand(0,8; 1,3), d \in D_p^P, p \in P$$

Secundair transport

Als model voor de kosten voor secundair transport c_{dq}^S is gekozen om de kosten afhankelijk te maken van de afstand van het depot $d \in D$ tot de klant-productset $q \in Q$, de vraag v_{qp} van de klant-productset naar product $p \in P$ en de transportkosten $KT(p)$ van het product. Ook hier wordt weer vermenigvuldigd met een random getal. Voor de kosten van de toewijzing moet daarna nog de som worden genomen over alle producten.

$$c_{dq}^S = \sum_{p \in P} afs(d, q) \cdot v_{qp} \cdot KT(p) \cdot rand(0,8; 1,3)$$

Primair transport

Als model voor de kosten van primair transport c_{dep}^P spelen de afstand tussen de depots $d, e \in D$ en de transportkosten $KT(p)$ van product $p \in P$ een rol. Er wordt ook hier vermenigvuldigd met een random getal.

$$c_{dep}^P = afs(d, e) \cdot KT(p) \cdot rand(0,8; 1,30)$$

Voor de overslagdepots is het primair transport goedkoper. Bij vervoer vanaf een overslagdepot wordt de kostenparameter nog vermenigvuldigd met de parameter *OverslagKorting*.

$$OverslagKorting \in (0,1]$$

4.1.7 Depot-productsets

Als laatste worden de depot-productsets nog toegevoegd. Het aantal wordt aangegeven met *ADP*. Iedere depot-productset $r \in R$ een levert extra product p^r op, met overal kosten gelijk aan nul en bij sommige depots een extra voorwaarde.

Bij het toevoegen van een depot-productset wordt de verzameling producten uitgebreid met één extra product. Alle mogelijke kosten die met dit nieuwe product te maken hebben, worden op nul gezet.

Er moet nog bepaald worden welke echte producten tot de depot-productset behoren. Hiervoor wordt de parameter *PSetProb* gebruikt. Ieder product komt met een kans van *PSetProb* in de depot-productset $r \in R$. Het wordt niet toegelaten dat een dummy-product dat is toegevoegd voor een depot-productset bij een andere depot-productset in komt te zitten.

Bij iedere klant-productset $q \in Q$ wordt de vraag naar product p^r berekend als de som van de vraag over de producten in depot-productset r .

$$v_{p^r q} = \sum_{p \in r} v_{qp}, q \in Q, r \in R$$

De depot-productset $r \in R$ levert geen extra voorwaarden op voor de productie. Bij alle productiedepots $d \in D^P$ wordt de maximale productie $u_{dp^r}^{max}$ van product p^r gelijk aan de som van de maximale productie u_{dp}^{max} van de producten $p \in r$.

$$u_{dp^r}^{max} = \sum_{p \in r} u_{dp}^{max}, d \in D^P$$

De depot-productset $r \in R$ kan een extra voorwaarde opleveren voor de throughput bij depot $d \in D$. Hiervoor gebruiken de parameter *ExtraProb*. De kans dat een depot-productset een extra voorwaarde oplevert is gelijk aan *ExtraProb*.

Als er geen extra voorwaarde komt, dan is de maximale throughput $t_{dp^r}^{max}$ van product p^r gelijk aan de som van de maximale throughput t_{dp}^{max} van de producten $p \in r$.

$$t_{dp^r}^{max} = \sum_{p \in r} t_{dp}^{max}, d \in D$$

Als er wel een extra voorwaarde komt, dan wordt de formule aangepast met een factor $1/(\frac{1}{2} + \frac{1}{2} \cdot \text{speling})$.

$$t_{dp}^{max} = \sum_{p \in r} \frac{t_{dp}^{max}}{\frac{1}{2} + \frac{1}{2} \cdot \text{speling}}, d \in D$$

Omdat *Speling* groter dan één is, komt er inderdaad een extra voorwaarde, maar het toevoegen van depot-productsets kunnen niet zorgen dat de totale capaciteit te klein wordt; de totale maximale throughput blijft groter dan de totale vraag.

4.1.8 Depot-productsets in het primair transport

In sectie 2.2.5 wordt een methode beschreven hoe in een instantie van een BTP een depot-productset weggewerkt kan worden. Bij deze methode wordt uitgegaan van onderstaande aanname:

Ieder depot kan ieder product produceren en de kosten van de productie zijn overal gelijk.

Onder deze aanname hebben we eigenlijk alleen te maken met het oplossen van het secundair transport. Binnen het secundair transport werkt de methode om iedere throughputvoorwaarde voor een depot-productset of voor een depot te vervangen door een extra product.

In het algemene geval van het BTP hebben we ook te maken met primair transport. Voor het primair transport werkt de methode uit 2.2.5 niet zoals gewenst is. Dit komt doordat bij leveringen in het primair transport producten los van elkaar worden behandeld, terwijl in het secundaire transport een levering uit hoeveelheden van verschillende producten bestaat.

Indien ook in het primaire transport met depot-productsets gewerkt wordt, dan zullen deze en hun bijbehorende voorwaarden meegenomen moeten worden naar het resulterend probleem. Dummy-producten brengen voor het LP geen extra kosten of voorwaarden. Een omzetting van een depot-productset naar een extra product binnen het secundair transport heeft geen gevolgen voor de kosten of de oplosbaarheid van het RP.

In de implementatie is niet speciaal rekening gehouden met voorwaarden van depot-productsets in het primaire transport. Voor de complexiteit van een instantie van het BTP maakt het niet veel uit, omdat het RP in beide gevallen een LP blijft.

4.2 Testresultaten

In deze sectie bespreken we de uitgevoerde tests met ons algoritme. Voor kleine instanties vergelijken we de oplossingen met, door CPLEX berekende, optimale oplossingen. Voor grote instanties blijkt dat de huidige implementatie qua snelheid nog tekort schiet.

Testomstandigheden

Het ontwikkelde algoritme is geïmplementeerd in Delphi. Het Resultierend Probleem (RP) werd steeds opgelost met behulp van CPLEX 12.2.0.0. De tests zijn uitgevoerd op een Pentium Dual Core T4400 (2 keer 2,20 Ghz) met 4,00 GB geheugen.

Kwaliteit van de oplossingen

Om de kwaliteit van de gevonden oplossingen van het ontwikkelde algoritme te onderzoeken, zijn tien instanties met gelijke parameters opgesteld.

Bij deze tien instanties heeft CPLEX de optimale oplossingen bepaald, zodat de oplossingen van het ontwikkelde algoritme vergeleken kunnen worden met de optimale oplossingen.

We beschrijven eerst hoe de instanties eruit zien.

De instanties

- De parameters die de grootte van het gebied bepalen is 100.
- De kans op een positieve vraag van een klant-productset naar een product, *ProbV*, is 0,7.
- De fractie distributiedepots, *ProbD*, is 0,6.
- De fractie overslagdepots, *ProbO*, is 0,2.
- De fractie productiedepots, *ProbP*, is 0,2.
- De korting voor primair transport vanaf overslagdepots, *Overslagkorting*, is 0,5.
- De kans dat een depot-productset bij een depot een extra voorwaarde oplevert is 0,3.
- Het aantal producten, *AP*, is 3.
- Het aantal klant-productsets, *AK*, is 100.
- Het aantal depots, *AD*, is 10.
- Het aantal depot-productsets, *ADP*, is 0.
- De waarde van *Speling* is 1,5.

Bovenstaande waarden van de parameters zullen verder worden aangeduid als de standaardinstellingen.

We gaan de effectiviteit van een enkele run van het ontwikkelde algoritme onderzoeken bij de standaardinstellingen. Een enkele run van de aangepaste versie

van het simulated annealing algoritme (SA-A) had een starttemperatuur van 10000, een factor van 0,99995 en eindtemperatuur van 1. In de tabel 5 staan de resultaten.

Na het nummer in de eerste kolom, staat in de tweede kolom de waarde van de doelfunctie in het optimum, in de derde kolom het resultaat van een enkele run van het ontwikkelde algoritme (SA-A) en in de vierde kolom de tijd die het algoritme daarvoor nodig had. In de vijfde kolom staat het absolute verschil tussen de gevonden oplossing en het optimum en in de laatste kolom staat de procentuele afwijking van het optimum.

Nr	Optimum (CPLEX)	SA-A	Tijd(s)	Vershil	Procentuele afwijking (%)
1	660077,0	860750,9	161	200674	30,4
2	2755898,8	2861342,8	138	105444	3,8
3	2737306,8	2816704,8	128	79398	2,9
4	2470811,9	2507477,2	142	36665	1,5
5	1346032,9	1377821,6	69	31789	2,4
6	2737306,8	2798594,6	148	61288	2,2
7	3270919,1	3329204,8	85	58286	1,8
8	843858,4	853851,9	140	9994	1,2
9	1528863,2	1596691,5	111	67828	4,4
10	3270919,1	3317386,5	110	46467	1,4

Tabel 5: meetresultaten van tien instanties met standaardinstellingen

We zien dat op eerste instantie na, SA-A oplossingen oplevert die een klein percentage van het optimum afwijken.

Verschillen in rekentijd kunnen veroorzaakt worden door verschillen in het aantal burens dat geaccepteerd wordt. In het bijzonder zullen burens, waarbij er een toegelaten oplossing bestaat extra tijd kosten. Voor ieder van die burens wordt namelijk het LP opgesteld en opgelost.

Het ontwikkelde algoritme is net als SA een algoritme, waarin toeval een rol speelt. Het zal daarom altijd raadzaam zijn om instanties vaker door te rekenen om betere oplossingen te vinden.

Bij de eerste vijf testinstanties leverden een paar extra runs betere oplossingen op. Het vinden van optimale oplossingen is, zelfs na veel runs, niet te verwachten. In tabel 6 staan de resultaten van extra runs.

Nr	Optimum (CPLEX)	SA-A	Vershil	Procentuele afwijking (%)
1	660077,0	816090,7	156014	23,6
2	2755898,8	2858030,7	102132	3,7
3	2737306,8	2803307,7	66001	2,4
4	2470811,9	2506744,9	35933	1,5

5	1346032,9	1365880,5	19848	1,5
---	-----------	-----------	-------	-----

Tabel 6: meetresultaten van eerste vijf testinstanties na meerdere runs van SA-A.

Grote instanties

De huidige wijze waarop SA-A geïmplementeerd is niet efficiënt. Veel berekeningen van kosten, overschrijdingen worden iedere iteratie opnieuw gedaan, terwijl ze (grotendeels) weer hetzelfde zijn. Verdere ontwikkeling op dit gebied was binnen de tijd niet haalbaar. Grote instanties testen duurt extra lang. Er is één test gedaan.

Afwijkingen van de standaardinstellingen:

- Het aantal producten, AP , is 6.
- Het aantal klant-productsets, AK , is 4000.
- Het aantal depots, AD , is 30.
- De waarde van *Speling* is 2.

Het lukte SA-A om een toegelaten oplossing te vinden. Een run van het algoritme op deze instantie met dezelfde parameters als voor tabel 5 gebruikt werden duurt bij een dergelijke instantie een aantal uren. Omdat de kwaliteit van de oplossing moeilijk meetbaar is, is de test niet afgemaakt.

5 Conclusies en aanbevelingen

5.1 Conclusies

Het BTP kan opgedeeld worden in twee deelproblemen. Dit kan gebruikt worden bij het oplossen van het BTP. Zeker bij instanties waar het vinden van toegelaten oplossingen moeilijk is, is het verstandig om eerst het secundair transport op te lossen.

De deelproblemen afzonderlijk optimaal oplossen zal niet tot goede oplossingen leiden. Het is nodig om bij het oplossen van de deelproblemen gegevens uit te wisselen, om oplossingen te genereren die dichterbij de optimale oplossing komen van het algehele probleem.

Om de kwaliteit van het ontwikkelde algoritme goed te kunnen beoordelen is extra ontwikkeling van de snelheid nodig. Het vinden van de juiste parameters vraagt meer testen.

5.2 Aanbevelingen

Er zijn nog veel mogelijkheden voor extra onderzoek. In willekeurige volgorde noemen we er een aantal.

5.2.1 Kosten van primair transport meenemen bij SA

Er is nu voor gekozen om de minimale extra kosten voor primair transport mee te nemen bij het kiezen van het secundair transport. Dat is zeker niet de enige mogelijkheid. Er zijn nog tal van andere mogelijkheden waarmee geëxperimenteerd kan worden. Er kan gedacht worden aan:

- Shadowprices uit de laatste oplossing van het LP
- Gemiddelde kosten om een product bij een depot te krijgen
- Een lopend gemiddelde van één of meer van bovenstaande waarden

5.2.2 Andere startoplossing

Er kan nog onderzocht worden wat het effect is van de startoplossing op de eindoplossing. Een eenvoudige manier om dat te doen is door eindoplossingen van het algoritme als startoplossing te nemen en te onderzoeken of er verbeteringen gevonden worden. Een startoplossing, gebaseerd op de oplossing van de LP-relaxatie van het probleem, zou goed kunnen werken.

5.2.3 Speed-up van het algoritme

Er zijn nog zeker veel mogelijkheden om het algoritme sneller te laten verlopen. Bij kostenberekeningen worden overschrijdingen van de voorwaarden vaak dubbel berekend. Een iteratie zal sneller verlopen als de overschrijdingen van de voorwaarden van de huidige oplossing bewaard worden.

Extra analyse van het algoritme biedt vermoedelijk meer mogelijkheden om dubbele berekeningen te voorkomen.

De eerste keer het RP oplossen zou ook uitgesteld kunnen worden totdat een zeker aantal iteraties geweest is, zonder aan kwaliteit van de eindoplossing in te leveren.

5.2.4 Gewichten los van elkaar aanpassen

In de huidige implementatie zijn de gewichten van de penaltykosten op ieder moment allemaal gelijk aan elkaar. In [2] wordt ook met gewichten van penaltykosten gewerkt, maar daar zit het aanpassen van de gewichten ingewikkelder in elkaar. De mate van verhoging van een gewicht is, onder andere, afhankelijk van de hoeveelheid waarmee een voorwaarde overschreden wordt.

5.2.5 Andere buurtomgevingen

In de gebruikte implementatie zijn alleen de verplaatsomgeving en de verwisselomgeving verwerkt. Het toevoegen van andere omgevingen kan zeker de kwaliteit van de oplossingen verhogen.

5.2.6 Penalty's bij onoplosbaar RP

We hebben laten zien dat er instanties denkbaar zijn, waarbij een toewijzing van het secundair transport tot onoplosbaarheid van het RP kan leiden. Indien dit gebeurt stuurt ons algoritme de oplossing niet in een richting die dit voorkomt. Bij instanties die hier wel sterk last hebben kan het toewijzen van klant-productsets aan depots die ook kun produceren een extra penalty krijgen, zodat deze depots meer productiemogelijkheden overhouden.

6 Literatuurlijst

- [1] Günter Schmidt, Wilbert E. Wilhelm
"Strategic, Tactical and Operational Decisions in Multi-national Logistics Networks: A Review and Discussion of Modeling Issues"
International Journal of Technology Management Volume 28, Number 2 (2004), 151 - 171
- [2] Mutsunori Yagiura, Shinji Iwasaki, Toshihide Ibaraki, Fred Glover
"A Very Large-Scale Neighborhood Search Algorithm for the Multi-Resource Generalized Assignment Problem"
Discrete Optimization Volume 1, Issue 1, (2004), 87 - 98
- [3] Mutsunori Yagiura, Akira Komiya, Kenya Kojima, Koji Non
A Path Relinking Approach for the Multi-Resource Generalized Quadratic Assignment Problem
Lecture Notes in Computer Science, Volume 4638 (2007), 121-135
- [4] P.A.W. Goudvis
"Door de bomen het BOSS weer zien; Analyse van een distributieketen van producent tot consument"
Augustus 2001, Katholieke Universiteit Tilburg
- [5] H. W. Kuhn
"The Hungarian method for the assignment problem"
Naval Research Logistics Quarterly Volume 2, Issue 1-2, (1955), 83-97
- [6] Dimitris Bertsimas, John Tsitsiklis
"Simulated Annealing"
Statistical Science Vol 8, No 1 (1993), 10 - 15
- [7] Bezalel Gavish, Hasan Pirkul
"Algorithms for the Multi-Resource Generalized Assignment Problem"
Management Science Vol 37 No. 6 (1991), 695 -713
- [8] Joseph B. Mazzola, Stephen P. Wilcox
"Heuristics for the Multi-Resource Generalized Assignment Problem"
Naval Research Logistics, Vol 48 (2001), 468-483
- [9] Günter Schmidt, Wilbert E. Wilhelm
"Strategic, Tactical and Operational Decisions in Multi-national Logistics Networks: A Review and Discussion of Modeling Issues"
International Journal of Production Research, Edition: September (1999)
- [10] www.ortec.com
- [11] <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>