

University of Groningen

Advanced analysis of branch and bound algorithms

Turkensteen, Marcel

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version

Publisher's PDF, also known as Version of record

Publication date:

2007

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Turkensteen, M. (2007). *Advanced analysis of branch and bound algorithms*. [Thesis fully internal (DIV), University of Groningen]. s.n.

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Samenvatting (Summary in Dutch)

Dit proefschrift gaat over moeilijk oplosbare wiskundige problemen die veelvuldig in de praktijk opduiken. Ze behoren tot de klasse der *combinatorische optimalisatieproblemen* (COPs). Een kenmerk van dergelijke problemen is dat er een eindig aantal mogelijkheden is waaruit een optimale keuze bepaald dient te worden. Met ‘optimaal’ bedoelen we gewoonlijk dat de kosten zo laag mogelijk dienen te zijn, of de winst zo hoog mogelijk. Een voorbeeld van een COP is het kortste pad-probleem, dat in routeplanners opgelost wordt: gegeven een aantal verbindingen tussen locaties wordt getracht een kortste pad tussen de vertrekplaats en de bestemming te bepalen. De keuzemogelijkheden zijn hier de mogelijke paden tussen het vertrek- en aankomstpunt en dat zijn er heel erg veel!

De verzameling mogelijkheden heet de *zoekruimte* van het probleem. Ook al bestaat de zoekruimte uit een eindig aantal mogelijkheden, dat betekent nog niet dat het probleem opgelost kan worden door successievelijk alle mogelijkheden te evalueren. Vaak zijn er voor COPs buitensporig veel keuzemogelijkheden. Voor veel praktische problemen bestaan evenwel snelle oplossingsmethodes, zoals voor het kortste pad-probleem. Dergelijke problemen worden normaal gesproken binnen enkele secondes opgelost, ook al zijn er honderden miljoenen mogelijkheden. Dit komt doordat slimme oplossingsmethodes slechts een fractie van alle mogelijkheden in beschouwing nemen.

Voor een grote klasse problemen bestaan zulke slimme methodes echter niet en zullen ze, naar velen vermoeden, ook nooit gevonden worden. In het slechtste geval dienen bijna alle mogelijkheden in de zoekruimte bekeken te worden, voordat een optimale oplossing vastgesteld kan worden. Problemen in deze klasse worden \mathcal{NP} -lastig genoemd (\mathcal{NP} -hard in het Engels).

Het vinden van een optimale oplossing voor een \mathcal{NP} -lastig probleem kan dus zeer veel tijd vergen. Indien die tijd niet beschikbaar is, neemt men in de praktijk vaak de toevlucht tot *heuristieken*: snelle methodes die niet noodzakelijk tot optimale oplossingen leiden. Hoewel met zogenaamde *meta-heuristieken* oplossingen bepaald kunnen worden die hooguit slechts een paar procent slechter zijn dan de optimale, kan een extra verbetering van een paar procent tot besparingen leiden die zeer de moeite waard zijn. Dit feit motiveert in belangrijke mate de zoektocht naar verbeterde oplossingstechnieken.

Een belangrijke klasse van oplossingsmethodes voor \mathcal{NP} -lastige problemen is *Branch and Bound (BnB)*. Veel \mathcal{NP} -lastige problemen worden gemakkelijk oplosbaar wanneer er extra restricties opgelegd worden. De ontstane gemakkelijk oplosbare problemen heten *relaxaties*. BnB methodes beginnen met het oplossen van een relaxatie. Als de gevonden oplossing niet *toelaatbaar* is, dat wil zeggen, als het niet tot de zoekruimte van het oorspronkelijke probleem behoort, dan wordt de zoekruimte opgedeeld in *subproblemen*. Dit proces gaat door totdat alle subproblemen ofwel opgelost of uitgesloten zijn. Een subprobleem is *opgelost* als de oplossing van de relaxatie toelaatbaar is voor het oorspronkelijke probleem; een subprobleem heet *uitgesloten* als het geen verdere bijdrage levert aan het oplossingsproces.

Hoodstuk 2 gaat over een aanpassing van de BnB techniek voor het zogenaamde *asymmetrische handelsreizigersprobleem*. Dit probleem duiden we aan met de aan het Engels ontleende afkorting *ATSP*. Gegeven n ($n \geq 3$) locaties en de afstanden tussen elk paar locaties, dient er een kortste rondrit, ofwel een kortste complete *tour*, gevonden te worden die elke locatie precies één keer aandoet. De afstand tussen locatie i en locatie j is hierbij niet noodzakelijkerwijs gelijk aan de afstand tussen j en i . Het ATSP is \mathcal{NP} -lastig en BnB technieken zijn veel gebruikte methoden om een optimale oplossing van het ATSP te vinden. De meest voor de hand liggende relaxatie is het toewijzingsprobleem, afgekort met AP. In het AP dient elke locatie in een tour te liggen, maar hiervoor mogen meerdere gescheiden tours, de zogenaamde *subtours*, gebruikt worden. De gangbare techniek om een ATSP-oplossing te maken uit een AP oplossing met meer dan één subtour is *patching*. De subtours worden hierbij samengevoegd tot één complete tour door alle locaties. De kosten die gepaard gaan met patching dienen als bovengrens voor de optimale ATSP-tour wanneer deze lager zijn dan de waarde van de beste oplossing tot dan toe in het BnB proces. Als we kunnen vaststel-

len dat alle toelaatbare oplossingen van een subprobleem hogere kosten hebben dan de huidige beste oplossing, dat wil zeggen, als de benedengrens van het subprobleem hoger is dan de huidige bovengrens, dan kunnen we dit subprobleem uitsluiten. Hoe beter de bovengrens, des te kleiner is het aantal subproblemen dat de BnB methode in beschouwing hoeft te nemen.

In Hoofdstuk 2 wordt *iterative patching* geïntroduceerd; een methode waarin patching toegepast wordt in elk subprobleem in de BnB zoekprocedure. In bestaande BnB methodes voor het ATSP, zoals in Carpaneto *et al.* (1995), wordt patching alleen toegepast bij initiële subproblemen. Deze methodes maken gebruik van een *Best First Search (BFS)* strategie, een zoekstrategie waarin het meest veelbelovende subprobleem eerst opgelost wordt. BFS heeft het nadeel dat er veel geheugen nodig is om moeilijke probleeminstanties met veel subproblemen op te lossen. Daarom gebruiken wij *Depth First Search (DFS)*, een zoekstrategie waarbij het meest recentelijk gegenereerde subprobleem als eerste opgelost wordt. Het blijkt dat, in geval van *DFS*, *iterative patching* zinvol is. Het uitsluiten van subproblemen leidt tot een zodanige tijdwinst dat de tijd die nodig is om de patching uit te voeren ruimschoots vergoed wordt. De *KSP-procedure* van Karp en Steele (1991) en de *Contract-or-Patch* heuristiek van Gutin *et al.* (2001) blijken de beste patching-procedures om in *iterative patching* te gebruiken.

In Hoofdstuk 3 worden een aantal andere onderdelen van BnB methodes bekeken, wederom voor het ATSP en de AP-relaxatie. Indien een BnB methode als zoekstrategie *DFS* gebruikt, dan betekent dit dat de methode eerst een subprobleem geheel uitpluist, voordat een ander subprobleem in beschouwing genomen wordt. In het geval van het ATSP en de AP-relaxatie wordt een verbinding, zeg e , verboden. Daarna worden eerst subproblemen opgelost waar de verbinding e niet in voorkomt. Als e in elke optimale ATSP-oplossing voorkomt, dan heeft het BnB algoritme tevergeefs veel tijd besteed aan het zoeken in dit deel van de zoekruimte. Het is dus belangrijk om goed te voorspellen welke verbindingen gehandhaafd dienen te blijven, de *overlevers*, en welke verwijderd moeten worden, de *uitstervers*.

BnB methodes met de *DFS*-zoekstrategie bepalen de verbinding die verwijderd moet worden als volgt. Zoek de kleinste subtour (met het kleinste aantal verbindingen) en verwijder de verbinding met de hoogste kosten. Wij stellen voor om, in plaats van kosten, *boventoleranties* te gebruiken. De boventoleranties van

de verbinding e met betrekking tot de AP-oplossing is de maximale toename in de kosten van de verbinding waarbij deze zich nog steeds in deze optimale AP-oplossing bevindt. Onze experimenten tonen aan dat met boventoleranties accurater bepaald kan worden wat de overlevers en de uitstervers zijn, veel beter dan met kosten. We tonen ook aan dat de *benedengrens* van een subprobleem kunnen verhogen door boventoleranties te gebruiken. De benedengrens van een subprobleem is ten hoogste de waarde van de goedkoopste toegelaten opklossing van een subprobleem. Als deze waarde verhoogd wordt, hoeven minder subproblemen in beschouwing genomen te worden. Beide verbeteringen leiden ertoe dat het aantal te bestuderen subproblemen aanzienlijk teruggebracht kan worden. Wanneer beide verbeteringen samen geïmplementeerd worden, zijn de reducties groter dan de totale reductie wanneer de verbeteringen apart geïmplementeerd worden: het *synergie-effect*.

Toleranties hebben het nadeel dat het tijd vergt om ze te berekenen: om één boventolerantiewaarde te bepalen, dient een geheel AP opgelost te worden (al kan dit wel versneld uitgevoerd worden). De experimenten tonen desondanks aan dat voor veel instanties, inclusief instanties uit de praktijk, onze op toleranties gebaseerde methodes sneller werken dan de op kosten gebaseerde methodes. Ook blijkt dat deze DFS-methodes zich kunnen meten met de BFS-methode van Carpaneto *et al.* (1995). Met name voor moeilijke instanties kan de laatstgenoemde methode vaak geen oplossing vinden binnen de gestelde tijdslimiet van één uur, terwijl onze DFS-methodes daartoe wel in staat zijn.

In Hoofdstuk 4 wordt aanvullend onderzoek gedaan naar op toleranties gebaseerde BnB algoritmes. Ten eerste beschouwen we met behulp van toevalsgetallen gegenereerde instanties voor het ATSP. Op toleranties gebaseerde BnB methodes zijn met name snel voor problemen met veel verschillende afstanden. Deze problemen zijn volgens Zhang (1993) de moeilijkste toevalsinstanties. Ten tweede trachten we onze BnB-methodes te verbeteren door toleranties alleen te gebruiken ‘boven’ in de zoekboom; we verwachten dat ze hier het meest effectief zijn. Experimenten tonen echter aan dat dit niet altijd waar is. Ten derde beschouwen we de mogelijkheid om niet één, maar meerdere tolerantiewaarden bij de benedengrens op te tellen. Voor het AP blijkt dat niet mogelijk te zijn, maar voor het zogenaamde *Minimum Spanning Tree Probleem* wel; dit tonen we aan in Sectie 4.4

Hoofdstuk 5 tenslotte past de kennis van de hierboven besproken metho-

dieken toe op een concreet probleem uit de praktijk: het *segmentatieprobleem*. Hierin worden groepen consumenten, de zogenaamde *segmenten*, onderscheiden die min of meer op dezelfde wijze reageren op marketinginspanningen van een bedrijf. Bijvoorbeeld, consumenten in één segment kopen meer van het product tijdens een promotie-actie, terwijl andere consumenten hun koopgedrag nauwelijks veranderen. Wij concentreren ons hier op segmentatieproblemen waarbij de onderneming verschillende segmenten identificeert en vervolgens aan ieder segment een marketingmix toewijst.

Met name in internationale markten en bij bederfelijke goederen kunnen de logistieke kosten zo hoog worden dat bedrijven niet in staat zijn om segmentatiestrategieën, die het resultaat zijn van gebruikelijke methodes, uit te voeren. Daarom vallen ze terug op een strategie waarin landen of regio's als aparte segmenten bediend worden; zie Steenkamp en Ter Hofstede (2002). Een dergelijke strategie heeft als nadeel dat verschillen binnen landen en overeenkomsten tussen consumenten in naburige landen niet benut kunnen worden. In Hoofdstuk 5 geven we een methode die een afweging maakt tussen logistieke kosten enerzijds en het maken van zo homogeen mogelijke segmenten, de *fit*, anderzijds. Toepassing op een specifieke segmentatiestudie toont aan dat deze benadering in staat is internationale segmenten met relatief lage logistieke kosten te construeren. De segmentaties worden bepaald met behulp van *simulated annealing*, een effectieve en moderne metaheuristiek, die voor deze problemen goede oplossingen levert.

Tenslotte bestuderen we de mogelijkheid de BnB-methode uit Diehr (1985) toe te passen op de segmentatieproblemen uit onze cases. Het blijkt dat deze BnB methode niet snel genoeg is om segmentatieproblemen met 123 elementen op te lossen. Het blijkt zelfs meestal niet mogelijk de oplossing verkregen met *simulated annealing* te verbeteren.