Het Steiner boom probleem

Enkele gangbare algoritmes en heuristieken

Roy Visser, BSc.

Werkstuk Bedrijfswiskunde en Informatica

Vrije Universiteit, Amsterdam

Faculteit Exacte Wetenschappen

Bedrijfswiskunde en Informatica

De Boelelaan 1081a

1081 HV Amsterdam

Februari 2006

Voorwoord

Eén van de onderdelen van mijn studie Bedrijfswiskunde en Informatica is het schrijven van een zogenaamd BWI-werkstuk. De inhoud van dit werkstuk dient te bestaan uit een literatuuronderzoek over een onderwerp uit de Bedrijfswiskunde en Informatica.

Mijn BWI-werkstuk geeft een overzicht van een aantal algoritmes en heuristieken voor het Steiner boom probleem. Het Steiner boom probleem is een probleem afkomstig uit de grafen theorie en probeert een zo gunstig mogelijk netwerk te construeren. Hierbij mag gebruikt gemaakt worden van hulppunten. Door deze laatste toevoeging wordt het probleem dermate complex dat het tot de klasse van NP-complete problemen gerekend wordt.

Vanaf deze plaats wil ik de heer Wattel bedanken voor het aandragen van het onderwerp, het ter beschikking bestellen van literatuur en het nodige geduld bij het begeleiden van mijn werkstuk.

Roy Visser
Februari 2006

NB. In dit werkstuk is geen managementsamenvatting opgenomen, hiervoor wordt verwezen naar de samenvatting en conclusie in hoofdstuk 7 van dit werkstuk.

Inhoudsopgave

1 Inleiding

Het Steiner boom probleem is een algemenere versie van het meer bekende opspannende boom probleem. Bij het Steiner boom probleem gaat het erom een verzameling punten op een zo gunstig mogelijke manier te verbinden. Hierbij mag eventueel gebruik gemaakt worden van hulppunten, de zogenaamde Steiner punten.

Dit werkstuk behandelt de problematiek rond deze Steiner Bomen. Er wordt een algemene beschrijving van het probleem gegeven en de ontstaansgeschiedenis wordt behandeld. Vervolgens wordt er dieper ingegaan op de netwerkvariant van het Steiner boom probleem. Dit is de meest bekende variant van het probleem. Het belangrijkste doel van het werkstuk is om een overzicht te verschaffen van veel gebruikte algoritmes en heuristieken voor deze variant van het probleem. Gezien de omvang van het werkstuk is het ondoenlijk om alle bekende algoritmes en heuristieken te behandelen, dit werkstuk zal dan ook zeker niet volledig zijn. Wel geeft het een globaal beeld voor een ieder die iets over het Steiner boom probleem wil weten.

Dit werkstuk is in zeven verschillende hoofdstukken (inclusief deze inleiding) opgedeeld. In hoofdstuk twee wordt achtergrond informatie gegeven over het Steiner boom probleem in zijn algemeenheid. In hoofdstuk drie wordt dieper ingegaan op de geschiedenis van het Steiner boom probleem. In hoofdstuk vier wordt vervolgens de inleidende theorie gegeven over het Steiner boom probleem in netwerken. Hier wordt onder andere de complexiteit van het probleem uitgewerkt. In het volgende hoofdstuk wordt de behandelde theorie gebruikt om een aantal algoritmes te bespreken die een exacte en dus optimale oplossing geven voor het Steiner boom probleem. Omdat het Steiner boom probleem een zogenaamd NP-compleet probleem is, is het van belang om goede en snelle heuristieken te hebben. Deze heuristieken worden besproken in hoofdstuk zes. In hoofdstuk zeven tot slot is de samenvatting en conclusie opgenomen.

In dit werkstuk wordt er van uitgegaan dat de lezer enige basiskennis bezit over grafentheorie. De basisbegrippen uit de grafentheorie zullen in dit werkstuk dan ook veelal zonder verdere introductie gebruikt worden. Voor meer achtergrondinformatie over grafentheorie wordt de lezer verwezen naar standaard tekstboeken en collegedictaten op dit gebied. Het tweede deel van de dictaten geschreven door de Open Universiteit en gebruikt bij het vak Discrete Wiskunde aan de VU geeft een goede inleiding in de grafentheorie.

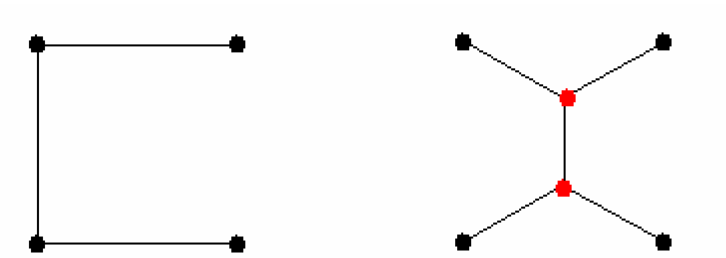
In de Engelstalige literatuur bestaan zowel een netwerk als graaf uit *vertices* en *edges*. In Nederlandstalige stukken worden deze begrippen op verschillende wijzen vertaald. In dit werkstuk zullen, wanneer het specifiek om een graaf gaat, de begrippen knoop *(vertex)* en zijde *(edge)* gebruikt worden. Echter wanneer het om een netwerk of om een positie in een vlak zullen ook de begrippen punt en verbindig gebruikt worden.

2 Het Steiner boom probleem

Het Steiner boom probleem is een verzamelnaam voor een aantal sterk aan elkaar verwante problemen. In de literatuur worden drie verschillende varianten van dit probleem onderscheiden, namelijk:

1. Het Euclidische Steiner boom probleem
2. Het Steiner boom probleem in netwerken
3. Het Recti-lineaire Steiner boom probleem

In al deze typen is het probleem een minimale opspannende boom T te creëren waarmee een gegeven set punten N verbonden wordt. Het verschil met het normale opspannende boom probleem is echter dat er nu extra punten (zogenaamde Steiner punten) gebruikt mogen worden om een kortere opspannende boom te creëren. Door deze toevoeging valt het probleem in de klasse van NP problemen, problemen die niet in een polynomiale tijd zijn op te lossen. In onderstaande figuur is een simpel voorbeeld te zien van hoe een opspannende boom verbeterd kan worden door het toevoegen van hulppunten. De linker graaf bevat een normale opspannende boom welke bijvoorbeeld met het algoritme van Prim bepaald kan worden. In de rechter graaf zijn zogenaamde Steiner punten toegevoegd (rode punten). Hierdoor kan een kortere opspannende boom bepaald worden. (de lengte van de rechter opspannende boom is 10 procent korter dan de linker opspannende boom)



figuur 2-1: Opspannende bomen (rechts met rode Steiner punten)

Van de hierboven genoemde problemen zal in dit werkstuk in het bijzonder ingegaan worden op het Steiner boom probleem in netwerken.

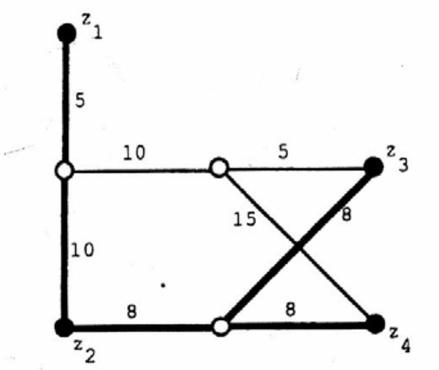
2.1 De verschillende probleemtypes

2.1.1 Het Euclidische Steiner boom probleem

Bij het Euclidische Steiner boom probleem (of kortweg het Euclidische probleem) is alleen de verzameling N van punten welke opgenomen dienen te worden in de opspannende boom gespecificeerd. De plaats en hoeveelheid van de Steiner punten is niet bepaald. Deze punten mogen dan ook vrij in het Euclidische vlak worden geplaatst. Aangezien er bepaalde restricties op plaats en graad van de knopen te leggen zijn is het mogelijk om een Euclidisch probleem om te schrijven naar een netwerk probleem. Echter er zijn exacte algoritmes en heuristieken beschikbaar die gebruik maken van enkele speciale eigenschappen van het euclidische probleem en daardoor efficiënter kunnen werken dan de algoritmes en heuristieken die beschikbaar zijn voor het netwerk probleem. Mede daarom wordt het euclidisch probleem dan ook echt als een afzonderlijk type probleem beschouwt. De boom getekend in figuur 2-1 is een voorbeeld van een Euclidische Steiner boom.

2.1.2 Het Steiner boom probleem in netwerken

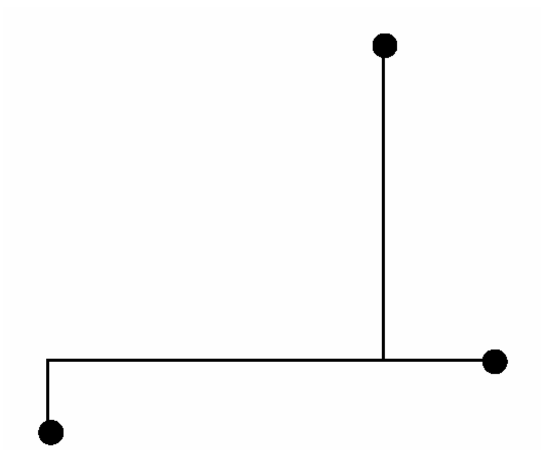
Dit is het bekendste type van het probleem. Bij het Steiner boom probleem in netwerken wordt een graaf (V,E) gegeven met daarbij een verzameling $N \subseteq V$ van knopen welke opgenomen dienen te worden. Deze knopen worden terminals genoemd. De overige knopen $S = N \setminus V$ kunnen gebruikt worden als Steiner knopen. Hiermee is de locatie en het maximum aantal van Steiner knopen vastgelegd. Over het algemeen hebben de zijden uit E allemaal een lengte c meegekregen. Zie onderstaande figuur voor een voorbeeld van het Steiner boom probleem in netwerken. De zwarte knopen zijn de terminals, de witte de non-terminals en dus mogelijke Steiner knopen. De dikgedrukte zijden geven de Steiner boom weer.



figuur 2-2: Netwerkvariant Steiner boom probleem

2.1.3 Het Recti-lineaire Steiner boom probleem

Het recti-lineaire probleem is wat betreft formulatie vrijwel gelijk aan het Euclidische Steiner boom probleem. Echter de verbindingen worden nu niet vrij in het Euclidische vlak gelegd maar mogen alleen horizontale of verticale lijnen zijn. Zie onderstaande figuur voor een voorbeeld van een recti-lineaire Steiner boom



figuur 2-3: Recti-lineaire Steiner boom probleem

2.2 Beschikbare literatuur

Er is een behoorlijke hoeveelheid literatuur beschikbaar waarin Steiner bomen behandeld worden. Echter de meeste van deze literatuur zijn boeken die over de grafentheorie als geheel gaan en waarin de problematiek van Steiner bomen kort aan bod komt. Zo behandeld Even (1979) het probleem kort wanneer de klasse van NP-complete problemen beschreven wordt. Ball, Magnanti, Monma en Nemhauser behandelen Steiner bomen als één van de problemen waarbij een optimale boom geconstrueerd dient te worden. Nadeel van beide boeken is echter dat zij wel het probleem of de verschillende types problemen benoemen en de complexiteit van het probleem behandelen, maar slechts zeer beperkt oplossingstechnieken of heuristieken benoemen.

Veruit de meeste informatie over Steiner bomen kan gevonden worden in *The Steiner Tree* Problem van Hwang, Richards en Winter. Dit boek is verschenen als onderdeel van de *Annals of discrete mathematics* en is voorzover bekend het enige boek wat volledig aan Steiner Tree problemen is gewijd. Bij veel literatuur op internet over beschikbare algoritmes en heuristieken wordt het dan ook als een soort kwaliteitskeurmerk vermeld wanneer de heuristiek of het algoritme genoemd wordt in dit boek. Mede daarom is dit boek ook de belangrijkste bron geweest voor het schrijven van dit werkstuk.

2.3 Praktische toepassingen van het probleem

Er zijn verschillende toepassingen bekend van het Steiner boom probleem. Zo kan het recti-lineaire probleem toegepast worden bij het aanleggen van het telefoonnetwerk in steden in de Verenigde Staten, juist omdat deze steden vaak (min of meer) in vierkante roosters van straten zijn opgebouwd.
Het Steiner Tree probleem in een netwerk kan uiteraard toegepast worden in communicatie-, distributie- en transportsystemen. Er zijn echter meer toepassingen te noemen van dit probleem. Zo dient het probleem ook opgelost te worden in het zogenaamde VLSI design, het plaatsen en verbinden van de verschillende componenten op chips. Het netwerk wordt in dat geval bepaald door de positie van de verschillende componenten.
Ook het Euclidische Steiner boom probleem kan toegepast worden in verschillende distributiesystemen. Nadeel hierbij is echter dat de Steiner punten op iedere plek in het vlak terecht kunnen komen. Er wordt totaal geen rekening gehouden met het wel of niet beschikbaar zijn van deze plaats en met het wel of niet de mogelijkheid hebben om een bepaalde verbinding te trekken. In de praktijk zijn dit toch zaken waar rekening mee gehouden dient te worden. Een Steiner boom probleem in netwerken kan dan uitkomst bieden.

3 Geschiedenis van het Steiner boom probleem

Het Steiner boom probleem is van origine geformuleerd als het Euclidische Steiner boom probleem. Vandaar dat in de literatuur de historie van het Steiner boom probleem meestal beschreven is vanuit het oogpunt van het Euclidische Steiner boom probleem. De historie van het dit probleem is onderzocht door Zacharis en Kuhn. Een samenvatting van deze resultaten is te vinden in het boek van Hwang, Richards en Winter (1992). Ter vergelijk, het Steiner boom probleem in netwerken wordt in dit boek niet vooraf gegaan door een uitgebreide beschrijving van de geschiedenis.

3.1 Probleem met 3 punten

Het originele Euclidische Steiner boom probleem is afkomstig van Fermat (1601-1665). Fermat beschreef het volgende probleem: Gegeven drie punten, v*ind een punt in een vlak, waarvan de som van de afstanden naar de drie gegeven punten minimaal is.*

3.1.1 Torricelli punt

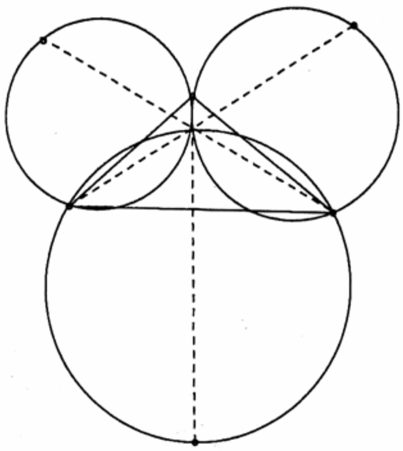
Torricelli(1608-1647) vond voor 1640 een geometrische oplossing voor dit probleem. Deze oplossing was als volgt:
Gegeven drie punten A, B en C

- Construeer op elke zijde van de driehoek ABC een gelijkzijdige driehoek
- Trek cirkels om deze drie gelijkzijdige driehoeken
- De drie cirkels snijden elkaar in het gezochte punt.

Dit punt wordt ook wel het Fermat-punt of Torricelli-punt genoemd. In het vervolg van het werkstuk wordt dit punt het Torricelli-punt genoemd.

In 1647 toonde Cavalieri in "Exercitationes Geometricae" aan dat de lijnen vanaf de drie gegeven punten naar het Torricelli-punt in het Torricelli-punt een hoek maken van 120 graden met elkaar. In 1750 toonde Simpson in zijn boek "Doctrine and Application of Fluxions" aan dat ook de drie lijnen vanuit een de gegeven punten naar de hoekpunten van de tegenoverliggende gelijkzijdige driehoek (geconstrueerd in de oplossing van Torricelli) snijden in het Torricelli-punt. Deze lijnen worden dan ook Simpson-lijnen genoemd. Heinen bewees in 1834 dat de lengte van een Simpson-lijn gelijk is aan de som van de lengtes van de lijnen vanuit de gegeven punten naar het Torricelli-punt. Onderstaande grafiek laat zowel het Torricelli punt als de Simpson-lijnen zien.

Verder liet Heinen zien dat, wanneer een van de hoeken van de driehoek ABC groter of gelijk aan 120 graden is, het Torricelli-punt buiten deze driehoek komt te liggen. Het Torricelli-punt is in dit geval niet langer het door Fermat bedoelde punt.



figuur 3-1: Torricelli-punt en Simpson-lijnen

3.1.2 Jakob Steiner

Simpson nam in zijn boek fluxions als oefening ook het algemene Fermat probleem op. Dit probleem luidt als volgt: zoek een punt in een vlak (of meer dimensionale ruimte) waarvan de afstand (of eventueel gewogen afstand) vanaf n verschillende punten minimaal is. Deze oefening trok de aandacht van een aantal bekende wiskundigen, waaronder Jakob Steiner. Echter, omdat hier slechts één punt toegevoegd mag worden heeft dit (behalve voor het geval n=3) niets te maken met het Steiner boom probleem zoals dat nu bekend is.

figuur 3-2: Jakob Steiner

Jarnik and Kössler schreven in 1934 een stuk in een Tsjechoslowaaks tijdschrift waarin zij een kortste netwerk probeerden te vinden om n willekeurige punten te verdelen. Zij onderzochten dit probleem voor het regelmatige n-gon en vonden een goede oplossing voor n = 3, 4 en 5. Verder toonden zij aan dat wanneer n ≥ 13 de oplossing gelijk was aan n-1 van de zijden van het regelmatige n-gon. Omdat dit probleem verschilt in van het door Fermat probleem refereerden Jarnik en Kössler niet naar Fermat.

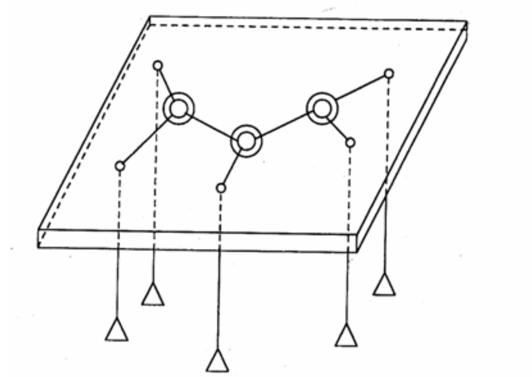
In 1941 schreven Courant en Robbins het boek *What is mathematics?*. In dit boek lieten zij de link zien tussen het Fermat problem en het door Jarnik en Kössler beschreven probleem. Echter bij het benoemen van deze problemen maakten zij een fout, zij noemden deze problemen namelijk 'Steiner Probleem' of 'Straten Netwerk probleem' in plaats van Fermat, Jarnik en Kössler te noemen. Omdat het boek behoorlijk populair is geworden werd niet alleen de aandacht op het probleem gevestigd, maar werd het probleem ook bekend onder de (verkeerde) naam 'Steiner probleem'

3.1.3 Melzak

Melzak beschreef vervolgens een groot aantal eigenschappen van netwerken en gaf een oplossing voor het Steiner probleem. Ook Gilbert en Pollak beschreven het probleem en gebruikten de begrippen 'Steiner Minimale Boom' en 'Steiner punten'. Ook breidden zij het probleem uit naar d-dimensionale ruimten. Verder kwamen ze met een probabilistische versie van het probleem.

3.1.4 Fysische modellen

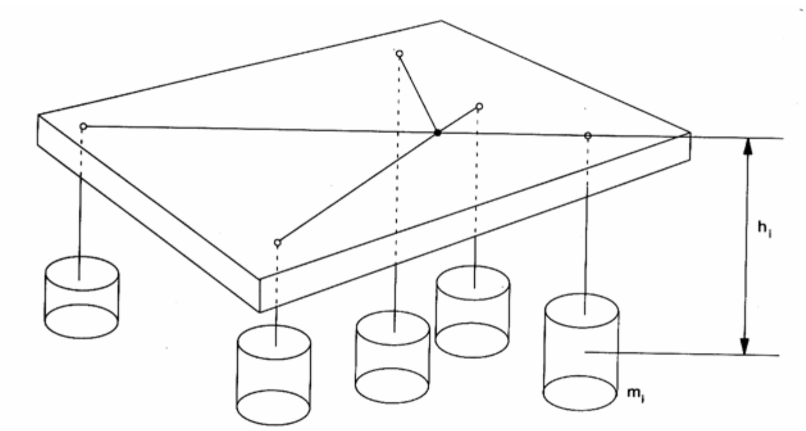
Er zijn een aantal fysische modellen bekend van, in het bijzonder, het Euclidische Steiner boom probleem. Eén daarvan is het zogenaamde *touw model*. Dit model bepaalt de exacte locatie van de Steiner punten, gegeven de wijze waarop de Steiner punten verbonden zijn met de terminals (dit wordt de topologie van het netwerk genoemd). Er wordt dus een minimale boom gevonden voor een bepaalde topologie. Zo'n boom wordt een relatieve minimale Steiner boom genoemd. Het model werkt als volgt. De locaties van de terminals worden door middel van gaten in een plank aangegeven. Vervolgens worden er touwen met gelijke lengte door de gaten getrokken en worden er gelijke gewichten aan ieder touw gebonden.



figuur 3-3: touw model

De Steiner punten worden vormgegeven door middel van metalen ringen. De verbindingen tussen de Steiner punten en de terminals worden gemaakt door de touwen door de gaten (terminals) heen te verbinden met de ringen (Steiner punten). Wanneer het geheel 'losgelaten' wordt (door de plank bijvoorbeeld een meter boven de grond te houden) schuiven de touwen en dus de ringen (Steiner punten) door de zwaartekracht naar de optimale positie.

Een variant op dit model wordt gegeven door Goor, Ploos van Amstel en Ploos van Amstel. In dit model wordt (zonder het te noemen) het eerder genoemde model van Jarnik and Kössler opgelost. Gegeven n verschillende punten in een vlak, zoek een punt in waarvan de afstand (of eventueel gewogen afstand) vanaf de n punten minimaal is. Door de gewichten die aan de touwen hangen van elkaar te laten verschillen kan de gewogen variant nagebootst worden. Zij gebruiken dit model om de optimale locatie voor een distributiecentrum te bepalen gegeven een aantal verschillende afzetcentra. Het gewicht staat dan symbool voor de omzet in het corresponderende afzetcentrum.



figuur 3-4: zwaartepunt model

4 Steiner boom probleem in netwerken

4.1 Probleembeschrijving

Het Steiner boom probleem in netwerken is een combinatorische versie van het Euclidische probleem. Het probleem kan als volgt geformuleerd worden.

> Gegeven een netwerk G met knopenverzameling V, zijden E met bijbehorende lengtes c(e) en een niet lege deelverzameling van V, genaamd N, vind een deelnetwerk $T_G(N)$ van G zodanig dat er een pad is tussen ieder paar knopen uit N en dat de totale lengte van $T_G(N)$ minimaal is.

De verzameling N wordt ook wel de verzameling van terminals genoemd. De verzameling V \ N wordt dan de verzameling van non-terminals genoemd. Iedere non-terminal die opgenomen wordt in het deelnetwerk $T_G(N)$ wordt een Steiner knoop genoemd. Groot verschil met het Euclidische variant van het Steiner probleem is dat de Steiner punten nu gekozen worden uit een eindige verzameling punten. Het deelnetwerk $T_G(N)$ zelf wordt een Steiner minimaal netwerk voor N in G genoemd.

4.1.1 Speciale gevallen

Er zijn een aantal uitzonderlijke gevallen van het hierboven beschreven probleem te noemen. Deze worden hier eenmaal genoemd en in het verdere verloop van dit hoofdstuk genegeerd.

Stel dat G uit minimaal twee (niet verbonden) componenten bestaat en dat de terminals in beide componenten voorkomen. In dat geval is er geen oplossing voor het probleem. In het andere geval, wanneer alle terminals in één component liggen kunnen de andere componenten genegeerd worden (tenzij er sprake zou zijn van zijden met een negatieve lengte in een van de andere componenten).

Iedere zijde met een negatieve lengte die wordt opgenomen in $T_G(N)$ verlaagt de totale lengte van $T_G(N)$. Daarom moeten alle zijden met een negatieve lengte opgenomen worden in $T_G(N)$.

In het vervolg zullen we er van uitgaan dat de zijden in netwerk G alleen positieve lengtes hebben. Verder zullen we aannemen dat G uit slechts een component bestaat. Gevolg hiervan is dat het minimale netwerk $T_G(N)$ noodzakelijkerwijs een boom is. Mede daardoor wordt het probleem in de literatuur vaak het Steiner boom probleem genoemd en wordt. Verder wordt $T_G(N)$ daardoor een Steiner minimale boom voor N in G genoemd.

4.1.2 Triviale gevallen

Naast de hierboven beschreven speciale gevallen zijn er ook enkele triviale gevallen te definiëren. In deze gevallen gaat het in principe om een ander type probleem welke beter met een voor dat probleem specifiek algoritme opgelost kan worden.

- Stel dat de verzameling N uit slechts 1 terminal bestaat. In dat geval bestaat de minimale Steiner boom uit exact deze ene terminal.
- Stel dat de verzameling N uit 2 terminals en een willekeurig (groter dan 0) aantal non-terminals bestaat. In dat geval wordt het probleem teruggebracht tot het wel bekende kortste pad probleem. Voor dit probleem zijn er diverse algoritmes bekend die in polynomiale tijd een oplossing berekenen. (bijvoorbeeld het Dijkstra algoritme)
- Stel dat de verzameling N gelijk is aan V. In dat geval is de minimale Steiner boom gelijk aan de minimale opspannende boom. Ook voor de constructie van een minimale opspannende boom zijn diverse algoritmes bekend. Ook deze algoritmes geven in polynomiale tijd een oplossing. (bijvoorbeeld het algoritme van Prim of het algoritme van Kruskal) Het algoritme van Prim wordt in hoofdstuk zes van dit werkstuk nog in detail besproken.

4.2 Complexiteit van het probleem

Het Steiner boom probleem is een NP-hard optimalisatie probleem. Karp heeft zonder bewijs aangetoond dat het probleem zelfs NP-hard is wanneer G een bipartite graaf is.

Een bipartite graaf is een graaf waarvan de knopen in twee disjuncte verzamelingen onderverdeeld kunnen worden zodanig dat er binnen één van deze verzamelingen geen paar van knopen is dat direct verbonden is.

Het Steiner boom probleem is zelfs NP-hard wanneer er geen enkele zijde is die een tweetal terminals of een tweetal non-terminals aan elkaar verbindt. Oftewel de twee disjuncte verzamelingen uit de bipartite graaf zijn exact de verzameling van terminals en de verzameling van non-terminals. Dit leidt tot het volgende Steiner boom beslissingsprobleem:

> Gegeven een bipartite graaf G met knopenverzameling V, zijden E met bijbehorende lengtes c en een niet lege deelverzameling van V, genaamd N en een positief geheel getal B, is er een deelnetwerk $T_G(N)$ van G zodanig dat er een pad is tussen ieder paar knopen uit N en dat het totale aantal zijden in $T_G(N)$ kleiner is dan B.

Om aan te tonen dat het Steiner boom beslissingsprobleem in een bipartite graaf niet alleen NP-hard is maar zelfs NP-compleet is dienen we het om te vormen naar een willekeurig probleem waarvan bekend is dat het NP compleet is. Het bewijs voor het NP-compleet zijn van het Steiner boom probleem in netwerken loopt via het zogenaamde *exact cover by 3-sets* probleem. Dit probleem wordt ook wel afgekort met

3XC. Van het 3XC probleem is algemeen bekend dat het tot de klasse van NP-complete problemen behoort. Dit kan eventueel, inclusief bewijs, worden na gelezen in het boek van Even.

Het *exact cover by 3-sets* probleem kan als volgt beschreven worden: Gegeven een collectie C van 3-sets, allen deelverzameling van een gemeenschappelijke set X, bestaat er een subcollectie van C zodanig dat het een exacte cover is van X?

Met een exacte cover wordt bedoeld dat ieder element uit X in de subcollectie C zit en dat geen enkel element uit X twee keer in de subcollectie C zit. Oftewel:

$$\bigcup_{i \in I} C_i = X$$

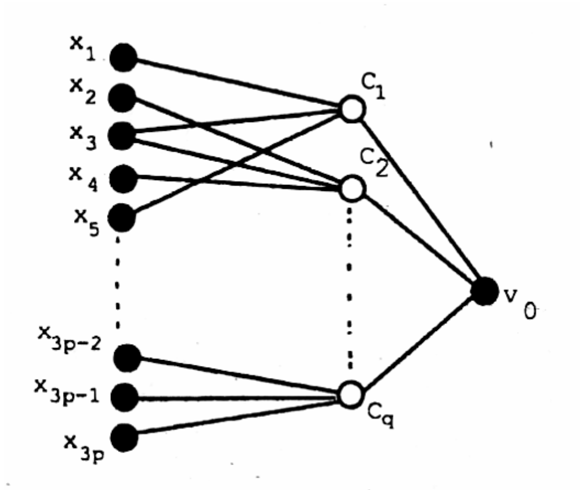
en voor iedere i, $j \in I$ waarvoor geldt $i \neq j$ moet gelden dat deze disjunct zijn:

$$C_i \cap C_j = \emptyset$$

Om aan te tonen dat het Steiner boom beslissingsprobleem NP-compleet is dient nu aangetoond te worden hoe een instantie van het 3XC probleem omgevormd kan worden in een instantie van het Steiner boom beslissingsprobleem. Dit wordt aangetoond in het bewijs van de volgende stelling. Ook dit bewijs is oorspronkelijk gegeven door Karp.

Stelling: Het Steiner boom beslissingsprobleem is NP-compleet
Bewijs: Laat de input voor 3XC bestaan uit een gemeenschappelijke set $X = \{x_1, x_2, \ldots, x_{3p}\}$ en een collectie $C = \{C_1, C_2, \ldots, C_q\}$ zodanig dat voor $C_i \subseteq X$ en $|C_i|=3$ iedere i geldt. De input f(I) voor het Steiner boom probleem is als volgt gedefinieerd (zie ook onderstaande figuur).

- $V = v_0 \cup C \cup X$:
  De verzamelingknopen bestaat uit een wortel v_0, een set knopen corresponderend met ieder element uit de collectie
- $E = \{(v_0,C_i) \mid 1 \leq i \leq q\} \cup \{(C_i,x_j) \mid x_j \in C_i, 1 \leq i \leq q,\ 1 \leq j \leq 3p\} \cup$
  De verzameling zijden bestaat uit een verbinding van v_0 naar ieder knoop afkomstig uit C. Vervolgens gaat er vanuit ieder knoop C_i een zijde naar ieder van de drie knopen x_j afkomstig uit C_i en dus uit de gemeenschappelijke set X (aangezien $C_i \subseteq X$)
- $N = v_0 \cup X$
  De knopen uit X verenigd met de knoop v_0 worden als terminals beschouwd. Als gevolg hiervan worden de knopen C_i als non-terminals beschouwd.
- $B = 4p$



figuur 4-1: Naar een 3XC probleem omgezette instantie

Gegeven deze mapping is de vraag of de oplossingen van 3XC inderdaad ook een oplossing voor het Steiner boom probleem zijn. Het mag duidelijk zijn dat wanneer C een exacte cover van X bevat, er dan een totaal aan knopen is van 3p (afkomstig uit X) + p (afkomstig uit C, exact p omdat er een cover is) + 1 (v_0) = 4p+1. Het totale aantal zijden is dan gelijk aan 4p (omdat het aantal zijden in een boom altijd één lager is dan het aantal knopen), dit is weer gelijk aan de waarde van B. Wanneer er dus een exact cover bestaat, bestaat er ook steiner boom met lengte B voor de verzameling N in G=(V,E) bestaat. Neem nu aan dat C geen exacte cover van X bevat. In dat geval moeten er tenminste p+1 non-terminals zijn. Het aantal knopen is dan minimaal 3p+(p+1)+1 = 4p+2 en het aantal zijden is dan minimaal 4p+1. Daarmee is bewezen dat er een Steiner boom met maximaal B zijden aanwezig is in de bipartite graaf G(V,E) dan en slechts dan als er in het corresponderende 3XC probleem een exacte cover is.

5 Exacte Algoritmes

5.1 Opspannende boom enumeratie algoritme

Het meest simpele exacte algoritme voor de constructie van een minimale Steiner boom is gegeven door Hakimi. Hij gaf het volgende algoritme

> *Een minimale Steiner boom $T_G(N)$ kan gevonden worden door simpelweg alle minimale opspannende bomen van de subnetwerken van G af te lopen. Deze subnetwerken worden gevormd door de supersets van de terminals.*

Iedere superset van de terminals is een verzameling van alle terminals plus nul of meer non-terminals. Wanneer we **alle** mogelijke supersets aflopen, en van ieder van deze supersets de opspannende boom in het corresponderende subnetwerk van G construeren weten we zeker dat de kleinste van deze opspannende bomen de gezochte Steiner minimale Boom is.

5.1.1 Aangepaste versie van het algoritme

Een aangepaste versie van dit algoritme is beschreven door Lawler. Deze aanpassing is gebaseerd op het gegeven dat het vinden van een minimale Steiner boom $T_G(N)$ in een voor de set N van terminals in netwerk G equivalent is aan het vinden van een Steiner minimale boom in voor dezelfde set N van terminals in het afstandsnetwerk $D = D_G(V)$

Het afstandsnetwerk $D_G(V)$ kan als volgt geconstrueerd worden.

1. Bepaal voor iedere twee punten in het oorspronkelijke netwerk het kortste pad tussen twee deze punten. (indien het mogelijk is een verbinding te trekken tussen deze twee punten)
2. Creëer een netwerk $D_G(V)$ met de punten verzameling V van het oorspronkelijke netwerk G. Dit netwerk is volledig verbonden, de lengtes (gewichten) van de verbindingen zijn de lengtes van de kortste paden geconstrueerd in stap 1. Wanneer er geen verbinding mogelijk is dient er een gewicht van plus oneindig toegekend te worden

Merk op dat wanneer het oorspronkelijke netwerk uit één component bestaat er in $D_G(V)$ nooit verbindingen met een waarde van plus oneindig zullen zijn. Verder hadden we al opgemerkt dat we netwerken met twee of meer componenten buiten beschouwing konden laten. Ook hadden we al aangenomen dat G alleen verbindingen met positieve lengtes zou bevatten. Daarom kunnen we vanaf nu aannemen dat het afstandsnetwerk $D_G(V)$ alleen uit positieve en eindige afstanden bestaat.

Zoals reeds gezegd is de aangepaste versie van het opspannende boom enumeratie algoritme gebaseerd op het feit dat het vinden van een minimale Steiner boom $T_G(N)$ in een voor de set N van terminals in netwerk G equivalent is aan het vinden van een Steiner minimale boom in voor dezelfde set N van terminals in het afstandsnetwerk $D = D_G(V)$. Dit kans als volgt aangetoond worden.

Lemma: $|T_G(N)| = |T_D(N)|$

Bewijs: Om te beginnen moeten we het netwerk G omvormen naar een volledig netwerk genaamd G*. Dit doen we door voor iedere twee niet rechtstreeks verbonden knopen in G met een lengte groter dan $|G|$ toe te voegen. Hierdoor krijgen deze zijden de lengte die equivalent lijkt aan plus oneindig. Het mag duidelijk zijn dat het oplossen van het Steiner boom probleem voor een set terminals N in G volledig equivalent is aan het oplossen van het Steiner boom probleem voor dezelfde set terminals N in G*.

Na het construeren van G* weten we dat iedere zijde in het afstandsnetwerk D een corresponderende zijde heeft in het volledige netwerk G*.

Neem nu aan dat een Steiner minimale boom $T_G(N)$ is gegeven. Oftewel $|T_G(N)| = |T_{G^*}(N)|$. Verder weten we dat iedere zijde in D niet langer is dan de corresponderende zijde in G*. Hieruit volgt dat $|T_G(N)| = |T_{G^*}(N)| \geq |T_D(N)|$
Neem nu aan dat een Steiner minimale boom $T_D(N)$ is gegeven. Wanneer we iedere zijde in $T_D(N)$ vervangen door het corresponderende kortste pad leidt dit tot een verbonden subnetwerk van G. Dit subnetwerk verbindt alle terminals. Dus geldt ook $|T_D(N)| \geq |T_G(N)|$. Wanneer we dit combineren komen we uit op $|T_G(N)| = |T_D(N)|$

Het algoritme beschreven door Lawler is op de volgende Lemma's gebaseerd.

Lemma: Er bestaat een Minimale Steiner boom $T_D(N)$ voor een set terminals N in het afstandsnetwerk D waarvoor geldt dat iedere Steiner knoop een graad van tenminste 3 heeft.

Bewijs: Merk eerst op dat $T_D(N)$ logischerwijs geen Steiner knopen met graad 0 of 1 zal bevatten, omdat deze altijd uit de boom verwijderd kunnen worden. Stel dat $T_D(N)$ een Steiner knoop v_k heeft met graad 2 en neem aan dat (v_k, v_i) en (v_k, v_j) in v_k eindigen. Dan kunnen deze zijden vervangen worden door de verbinding (v_i, v_j) en kan v_k uit de boom verwijderd worden. We hebben dan nog steeds een boom die alle terminals verbindt. Omdat de driehoeksongelijkheid geldt in D heeft deze boom een kleinere of gelijke lengte als de oorspronkelijke boom. Met deze methode kunnen de Steiner knopen met graad twee stuk voor stuk vervangen worden. Daarom bestaat er een minimale Steiner boom voor de verzameling terminals N in D waarvan alle Steiner knopen tenminste graad 3 hebben.

Lemma Er bestaat een minimale Steiner boom voor N in D met hoogstens n-2 steiner knopen

Bewijs Laat s het aantal Steiner knopen in $T_D(N)$, waarin alle Steiner knopen minstens graad 3 hebben (zie ook het bovenstaande lemma). Neem verder d_n als gemiddelde graad voor de terminals en d_s als gemiddelde graad voor de Steiner knopen. Het aantal zijden in $T_D(N)$ is dan $n+s-1 = (d_n n + d_s s)/2$. Verder weten we dat $d_n \geq 1$ en $d_s \geq 3$ en dus volgt daaruit: $n+s-1 = (d_n n + d_s s)/2 \geq (n+3s)/2$. Waaruit dan weer volgt $n-2 \geq s$

Op basis van dit tweede lemma kan geconcludeerd worden dat niet alle mogelijke supersets van terminals afgelopen hoeven te worden om een minimale Steiner boom te vinden. Het is voldoende om alle supersets van terminals bestaande uit maximaal 2n-2 knopen af te lopen en voor deze supersets een minimaal opspannende boom te construeren. Wanneer $2n-2 < v$ is het minder werk om $T_D(N)$ te vinden dan om $T_G(N)$ te vinden. Dit is helemaal het geval wanneer G een zogenaamd dense netwerk is. Aan de andere kant, wanneer G sparse is, raken bij het construeren van D de specifieke eigenschappen van het sparse netwerk kwijt en kan het construeren van $T_D(N)$ meer werk opleveren dan het construeren van $T_G(N)$ in het oorspronkelijke netwerk G.

Het kan berekend worden dat het aantal minimum opspannende bomen dat bepaald moet worden om $T_D(N)$ te vinden gelijk is aan.

$$\sum_{i=0}^{n-2} \binom{v-n}{i} \leq 2^{v-n}$$

De complexiteit van het algoritme van Lawler is dan dus $O(n^2 2^{v-n} + v^3)$ waarbij de laatste term afkomstig is van de kortste pad berekening. Verder valt op dat het algoritme weliswaar polynomiaal is in het aantal terminals (n), maar exponentieel is in het aantal non-terminals (v-n) in het netwerk.

5.2 Branch-and-Bound algoritme

In de volgende paragraaf wordt een branch-and-bound algoritme beschreven dat gebaseerd is op de zijden van de graaf. Dit algoritme is gevonden door Shore, Foulds en Gibbons. Een soort gelijk algoritme, maar iets minder efficiënt is gevonden door Yang en Wing.

Het belangrijkste idee achter dit branch-and-bound algoritme (en eigenlijk achter ieder branch-and-bound algoritme) is dat de set F_0 van alle mogelijke oplossingen (hier opspannende bomen voor de terminal set N) systematisch wordt onderverdeeld in steeds kleinere subsets. Voor iedere subset dient vervolgens bepaald te worden of deze wel of niet een minimale Steiner boom kan bevatten. Dit wordt gedaan door per subset een boven- en ondergrens te bepalen. De ondergrens geeft aan wat de minimale lengte is die een opspannende boom in die subset moet hebben (het is echter niet gezegd dat er ook daadwerkelijk een boom met die lengte aanwezig is in de subset). De bovengrens geeft aan met welke lengte er in ieder geval een opspannende boom te construeren is binnen die subset. Stel dat de ondergrens van een subset i hoger is dan de tot dan toe laagste gevonden bovengrens (en dus de kortste opspannende boom die met zekerheid te construeren is). In dat geval kan er met zekerheid gezegd worden dat subset i niet de minimale Steiner boom bevat. Subset i zal in het verdere verloop van het algoritme dan ook verder buiten beschouwing worden gelaten.

Iedere subset F_i van opspannende bomen in het branch-and-bound algoritme wordt beschreven door een set IN_i van zijden die verplicht in iedere boom in die subset opgenomen dienen te worden en een set OUT_i van zijden die in geen enkele opspannende boom binnen die subset opgenomen mogen worden. Uiteraard geldt in F_0 dat zowel IN_0 als OUT_0 uit een lege verzameling bestaat, immers voor F_0 geldt dat iedere mogelijke opspannende boom deel uitmaakt van deze set.

Gegeven een subset F_i zijn er drie zaken van belang om te weten

1. Wat is (wanneer deze bestaat) de optimale boom binnen F_i rekening houdend met de sets van verplichte en uitgesloten zijden, IN_i en OUTi. (Voor het algoritme is het niet noodzakelijk om in F_i een optimale boom te kunnen construeren, echter de methode om dit te doen is de basis van de methode van de bepaling van zowel de onder- als bovengrens)
2. Wat is de ondergrens van F_i, de minimale lengte die iedere opspannende boom in F_i tenminste heeft.
3. Wat is de bovengrens van F_i, een lengte waarbij in ieder geval een opspannende boom geconstrueerd kan worden.

Bij 1:
De minimale optimale boom binnen een subset F_i kan op de volgende wijze bepaald worden. Stel dat G_i het netwerk is dat verkregen wordt uit G door 1) het verwijderen van alle zijden die in OUT_i zitten en 2) door het samentrekken van knopen langs de zijden (*contraction along the edges*) van de set IN_i. Laat de set N_i bestaan uit de terminals in G_i, waarbij ook de knopen incident met de zijden uit IN_i als terminal worden beschouwd. Het bepalen van de optimale oplossing in F_i is equivalent met bepalen van een minimale Steiner boom $T_{Gi}(N_i)$ voor N_i in G_i. Wanneer $T_{Gi}(N_i)$ is gevonden dienen hier te zijden uit N_i aan toegevoegd te worden om de optimale oplossing voor de subset F_i te verkrijgen.

Bij 2:
Om een ondergrens b_i voor F_i te bepalen kan het volgende proces gevolgd worden:

Bepaal de volgende waardes voor een terminal z_k in G_i

$\bar{c}_{z_k}$: De lengte van de kortste zijde incident met z_k. Indien er geen enkele zijde incident is met z_k dan plus oneindig.

$\tilde{c}_{z_k}$: De lengte van de kortste zijde tussen z_k en een andere terminal in N_i. Als geen enkele zijde verbonden is met een andere terminal uit N_i dan wordt de waarde plus oneindig.

Neem aan dat $T_{Gi}(N_i)$ tenminste één Steiner knoop bevat. Gevolg hiervan is dat $T_{Gi}(N_i)$ tenminste $|N_i|$ zijden bevat. Iedere terminal is dan dus incident met tenminste één zijde uit $T_{Gi}(N_i)$. Verder kunnen deze zijden zo gekozen worden dat ze allemaal verschillend zijn. (Dit omdat er tenminste evenveel met terminals incidente zijden zijn als terminals zelf). De lengte van $T_{Gi}(N_i)$ is minimaal gelijk aan de kortste zijden incident met iedere terminal in N_i, oftewel:

$$\bar{t}_i = \sum_{z_j \in N_i} \bar{c}_{z_j} \leq \left| T_{G_i}(N_i) \right|$$

Neem nu aan dat $T_{Gi}(N_i)$ alleen uit terminals bestaat. Dan heeft $T_{Gi}(N_i)$ exact $|N_i|$-1 zijden. In dit geval is de lengte van $T_{Gi}(N_i)$ minimaal de som van de kortst mogelijk verbinding van iedere terminal naar (uiteraard) een andere terminal. Nu is echter exact één verbinding dubbel geteld. Om een ondergrens te vinden dient de kortste van deze verbindingen weer van de som afgetrokken te worden. Oftewel:

$$\tilde{t}_i = \sum_{z_j \in N_i} \tilde{c}_{z_j} - \min\left\{\tilde{c}_{z_j} \middle| z_j \in N_i\right\} \leq \left| T_{G_i}(N_i) \right|$$

Om een ondergrens te vinden dienen we de kleinste van deze twee waarden te nemen (we weten immers niet of er Steiner knopen in $T_{Gi}(N_i)$ aanwezig zijn). Uiteraard dient hier de lengte van de zijden in IN_i nog bij opgeteld te worden. In formule vorm levert dit:

$$u_i = \min\left\{\bar{t}_i, \tilde{t}_i\right\} + \sum_{e_m \in IN_i} c(e_m)$$

Bij 3:
Het bepalen van een bovengrens u_i voor F_i is vrijwel equivalent aan het bepalen van de optimale oplossing voor F_i zoals zojuist beschreven bij punt 1. Verschil is dat de minimale Steiner boom voor N_i in G_i nu niet 'echt' bepaald wordt maar vervangen wordt door een mogelijk andere opspannende boom voor N_i in G_i. Deze opspannende boom wordt verkregen met één van de heuristieken die later in dit werkstuk beschreven zullen worden.

5.2.1 Verloop van het algoritme

Laat b_0 de kortst gevonden bovengrens tot nu toe zijn. (Uiteraard starten we met b_0 gelijk aan plus oneindig)

Per set F_i wordt de bovengrens b_i en ondergrens u_i bepaald. Wanneer $b_0<u_i$ dan weten we zeker dat geen enkele boom in F_i kleiner zal zijn dan de best gevonden boom tot nu toe. Het heeft dan ook geen zin meer om F_i verder op te splitsen of uit te rekenen. Aan de andere kant als $b_0>u_i$ maar $u_i=b_i$ dan is de gevonden oplossing bij de bepaling van b_i ook meteen de beste oplossing voor F_i. In dit geval wordt b_0 gelijk aan b_i. Ook dan heeft het geen zin meer om F_i verder op te splitsen of door te rekenen.

In alle andere gevallen kan F_i niet genegeerd worden (het is niet zeker of er een minimale Steiner boom in F_i zit). F_i dient dan gepartitioneerd te worden in twee subsets. Om dit te doen wordt een geschikte zijde e_m gekozen. De twee subsets worden dan als volgt:

$$F_j : IN_j = IN_i \cup e_m ; OUT_j = OUT_i$$
$$F_k : IN_k = IN_i ; OUT_k = OUT_i \cup e_m$$

Uiteraard dient te gelden $e_m \in E \setminus \left(IN_i \cup OUT_i \right)$, oftewel de te kiezen zijde dient uit de verzameling van alle zijden te komen en mag niet al in IN_i of OUT_i zitten.

De keuze van e_m is voor een groot deel bepalend voor de performance van het branch-and-bound algoritme. Shore, Foulds en Gibbons hebben e_m gekozen als de zijde met minimale lengte incident met een terminal z_k uit N_i waarvoor $\hat{c}_{z_k}$ - $\bar{c}_{z_k}$ maximaal is. $\hat{c}_{z_k}$ is gedefinieerd als de lengte van de één na kortste zijde incident met z_k. Indien deze waarde niet bestaat dan wordt de waarde plus oneindig genomen.

Wanneer F_j en F_k bepaald zijn, dan wordt eerst F_j doorgerekend. Wanneer de optimale oplossing voor F_j is gevonden of wanneer is vastgesteld dat F_j geen optimale oplossing voor F_0 kan bevatten wordt F_k doorgerekend. Hierna stapt het algoritme terug naar F_i. Het algoritme eindigt wanneer het op deze wijze terug komt in F_0. Op dat moment bevat b_0 de lengte van de kortste opspannende boom, oftewel $|T_G(N)|$.

Er zijn ook branch-and-bound algoritmes bekend, welke gebaseerd zijn op de knopen van de graaf in plaats van op de zijden. Eén van deze algoritmes is ontwikkeld door Beasly. In dit algoritme wordt iedere subset gekenmerkt door een set IN_i van non-terminals welke opgenomen moeten worden een set OUT_i van non-terminals die uitgesloten dienen te worden in de opspannende boom. Dit algoritme wordt verder niet in detail besproken in dit werkstuk.

5.3 Overige algoritmes

Naast deze gedetailleerd besproken algoritmes zijn er nog een aantal overige algoritmes bekend. Deze worden niet tot in detail besproken in dit werkstuk. Deze algoritmes zijn gebaseerd op de volgende ideeën

- Aflopen van topologieën.
- Dynamisch programmeren
- Mathematisch programmeren

Uiteraard geldt voor al deze algoritmes dat zij in niet polynomiale tijd de optimale oplossing voor het Steiner boom probleem vinden. Voor algoritmes gebaseerd op mathematisch programmeren technieken geldt dat zij dit doen via het zogenaamde Steiner vertakkings Probleem. Dit probleem is een generalisatie van het hier besproken Steiner boom probleem. Oftewel, ieder Steiner boom probleem kan via een instantie van het Steiner Vertakkings Problem worden opgelost.

6 Heuristieken

Zoals al eerder genoemd is de beslissingsvariant van het Steiner boom probleem NP-compleet. Het is daarom ook erg onwaarschijnlijk dat er een exact algoritme bestaat dat het Steiner boom probleem in polynomiale tijd op kan lossen. Zoals genoemd bij de exacte algoritmes is de berekentijd van de nu bekende algoritmes of exponentieel in het aantal terminals, of exponentieel in het aantal non-terminals.

Vanwege deze exponentiele berekentijd is het van praktisch belang om goede heuristieken te hebben die snel een goede oplossing kunnen vinden voor het probleem. In dit hoofdstuk worden enkele van deze heuristieken besproken. Het gaat dan met name om:

- Pad heuristieken
- Boom heuristieken
- Knoop heuristieken

6.1 Boom en Pad heuristieken

Twee veel gebruikte typen heuristieken zijn boom en pad heuristieken. Pad heuristieken zijn gebaseerd op de volgende globale werkwijze. Startend met een willekeurig gekozen terminal (of eventueel een ander subnetwerk van het oorspronkelijke netwerk G) wordt de boom geleidelijk aan uitgebreid totdat alle terminals in de boom zijn opgenomen. Het uitbreiden van de boom is bij pad heuristieken over het algemeen gebaseerd op het toevoegen van kortste paden tussen terminals die nog niet, en knopen die al wel in de boom zijn opgenomen.

Bij boom heuristieken wordt juist in één keer een complete beginboom geconstrueerd. Vervolgens worden er technieken toegepast om deze beginboom te verbeteren.

6.2 Minimaal opspannende boom heuristiek

Takahashi en Matsuyama hebben de minimaal opspannende boom heuristiek beschreven. Bij deze heuristiek wordt eigenlijk het verkeerde algoritme voor het verkeerde probleem gebruikt. Er wordt dan ook geen gebruik gemaakt van de specifieke eigenschappen van het Steiner boom probleem.

Wanneer een minimaal opspannende boom voor alle knopen uit V wordt bepaald, wordt er ook een boom geconstrueerd die alle terminals verbindt gebruik makend van (in dit geval alle) non-terminals. Na constructie van deze minimale opspannende boom voor G(V,E) kunnen de non-terminals met graad één nog één voor één verwijderd worden om een iets betere oplossing te verkrijgen.

Hoewel de gevonden oplossing veel slechter is dan die van later beschreven heuristieken, is het toch interessant om hier het algoritme voor de constructie van een minimaal opspannende boom te bespreken. Vooral omdat sommige andere (kortste pad) heuristieken gebaseerd zijn op dit algoritme. Het gaat hierbij om het algoritme van Prim.

Het algoritme van Prim is bedoeld om een optimale opspannende boom te construeren tussen *alle* knopen van een graaf. Door geen onderscheid te maken tussen terminals en non-terminals, wordt echter ook een Steiner boom gecreëerd, waarbij *alle* non-terminals als Steiner knoop gebruikt worden. In dit geval gaat het uiteraard niet om de optimale oplossing voor het Steiner boom probleem.

Het algoritme van Prim werkt als volgt:

Gegeven een graaf G met knopenverzameling V en zijden E

1. kies een willekeurige knoop uit V en beschouw dit als subboom T
2. Herhaal zolang niet alle knopen uit V in T zijn opgenomen
    a. Zoek een zijde e_i met minimale lengte tussen een knoop uit T (knopen al opgenomen in de boom) en een knoop v_i uit $V \setminus T$ (knopen nog niet opgenomen in de boom)
    b. Voeg zowel v_i als e_i toe aan T

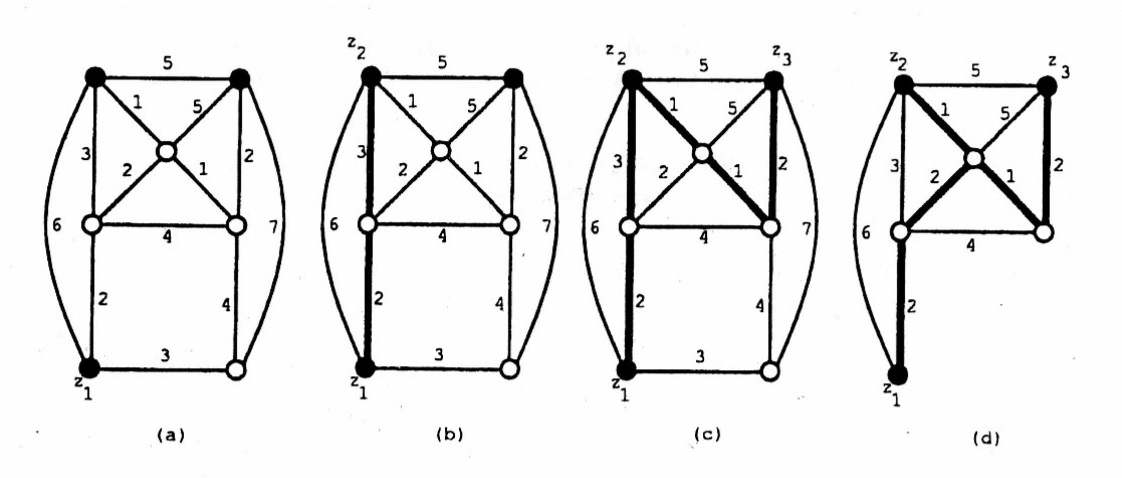
Wanneer alle knopen uit G opgenomen zijn in T vormt T de minimale opspannende boom.

6.3 Kortste pad heuristiek (SPH)

Takahashi en Matsuyama hebben een kortste pad heuristiek ontwikkeld voor het Steiner boom probleem. De werking van deze heuristiek lijkt sterk op het hierboven beschreven algoritme van Prim. Echter nu wordt er bij het toevoegen van een knoop wel degelijk onderscheid gemaakt tussen terminals en non-terminals. In iedere stap wordt een nieuwe terminal, door middel van zijn kortste pad naar de tot dan toe geconstrueerde deelboom, toegevoegd.

De kortste pad heuristiek werkt als volgt

1. Begin met een subboom T_{SPH}. Deze subboom bevat één willekeurig gekozen terminal z_1. Laat k het aantal terminals in de subboom zijn en n het totaal aantal terminals in het netwerk. Uiteraard geldt nu k=1
   Figuur a, gekozen terminal is z_1
2. Als k=n → STOP
3. Bepaal een terminal z_{k+1} die het dichtst bij T_{SPH} ligt. Voeg vervolgens aan T_{SPH} het kortste pad van de subboom naar z_{k+1} toe. Nu geldt k=k+1. Ga vervolgens terug naar stap 2.
   Figuur b, terminal z_2 gekozen en verbonden volgens het kortste pad
   Figuur c, terminal z_3 gekozen en verbonden volgens het kortste pad



figuur 6-1: werking van de kortste pad heuristiek

Rayward-Smith en Clare hebben aangetoond dat de op deze wijze geconstrueerde boom T_{SPH} op eenvoudige wijze nog verder verbeterd kan worden. Zij hebben hiervoor twee extra stappen gedefinieerd, welke uitgevoerd dienen te worden zodra het algoritme gestopt is.

4. Bepaal een minimaal opspannende boom voor de knopen uit het subnetwerk van G dat gevormd wordt door alle knopen (en alle zijden incident met deze knopen) die niet in T_{SPH} zitten weg te halen. NB. Er kunnen dus wel zijden uit G die niet in T_{SPH} zitten opgenomen worden in het subnetwerk, zolang zij maar incident zijn met twee knopen uit T_{SPH}
   Figuur d, boom verbeterd door het bepalen van de minimaal opspannende boom voor het deelnetwerk
5. Verwijder uit deze opspannende boom één voor één alle non-terminals met graad één. De boom die overblijft is de (suboptimale) oplossing

De kortste pad heuristiek kan geïmplementeerd worden door middel van een kleine wijziging van het eerder beschreven algoritme van Prim. Echter om het algoritme uit te voeren zijn wel alle kortste paden van alle terminals naar alle knopen (andere terminals en non-terminals) nodig. Dit is dan ook meteen de beperkende factor in de rekentijd nodig voor de heuristiek. De kortste paden kunnen berekend worden met behulp van het algoritme van Dijkstra. Aangezien dit algoritme $O(v^2)$ tijd nodig heeft om het kortste pad van één knoop naar alle v knopen te bepalen en dit voor alle n terminals noodzakelijk is, heeft de kortste pad heuristiek $O(nv^2)$ aan tijd nodig.

Een ander belangrijk aspect van iedere heuristiek is de mate waarin de gevonden (suboptimale) oplossing kan afwijken van de optimale oplossing. *The Steiner Tree* Problem van Hwang, Richards en Winter geeft een uitgebreid bewijs voor de volgende stelling. Hier zal de stelling zonder verder bewijs gegeven worden.

Stelling: $|T_{SPH}| \;/\; |T_G(N)| \leq 2 - 2/n$ voor ieder netwerk G en terminal set N.

Oftewel alleen in het slechtste geval is de gevonden boom meer dan 2-2/n keer zo lang als de optimale boom. Deze waarde wordt dan ook de worst-case foutratio genoemd.

Zoals eerder aangegeven is de beschreven heuristiek sterk gerelateerd aan het algoritme van Prim voor de constructie van een minimale opspannende boom. Het is ook mogelijk om de heuristiek te baseren op een andere algoritme voor de constructie van een minimale opspannende boom, namelijk het algoritme van Kruskal. Wang, Wildmayer en Plesnik hebben in verschillende publicaties aangetoond dat de benodigde rekentijd en afwijking ten opzichte van $T_G(N)$ gelijk zijn aan die van de op Prim gebaseerde heuristiek.

6.3.1 Repeterende versie van het algoritme

Er zijn verschillende repeterende versies van de kortste pad heuristiek bekend. De noodzaak voor repeterende versies van de heuristiek is gebaseerd op het feit dat de lengte van T_{SPH} afhankelijk is van de (willekeurig) gekozen startterminal. Daarom kan het mogelijk zijn om een betere oplossing te vinden door de heuristiek meerdere keren te herhalen, steeds met een andere start terminal. Winter en Smith hebben onderzoek gedaan naar deze repeterende versies. Zij hebben de volgende versies onderzocht.

1. SPH-N: bepaal T_{SPH} n keer. Iedere keer beginnend met een andere startterminal
2. SPH-V: bepaal T_{SPH} v keer. Iedere keer beginnend met een andere start knoop (dus zowel terminals als non-terminals)
3. SPH-zN: bepaal T_{SPH} n-1 keer. Iedere keer beginnend met een kortste pad tussen één vast gekozen startterminal naar een andere terminal.
4. SPH-NN: bepaal T_{SPH} n(n-1)/2 keer. Iedere keer beginnend met het kortste pad tussen een ander paar startterminals

Al deze repeterende varianten leiden tot betere resultaten. Met name SPH-V en SPH-NN leiden tot goede resultaten. Uiteraard brengen deze betere resultaten wel een langere rekentijd met zich mee.

6.4 Kortste pad met oorsprong heuristiek

Een erg simpele heuristiek is bestudeerd door Takahashi en Matsuyama. Dit is de zogenaamde kortste pad met oorsprong heuristiek. Deze heuristiek werkt als volgt:

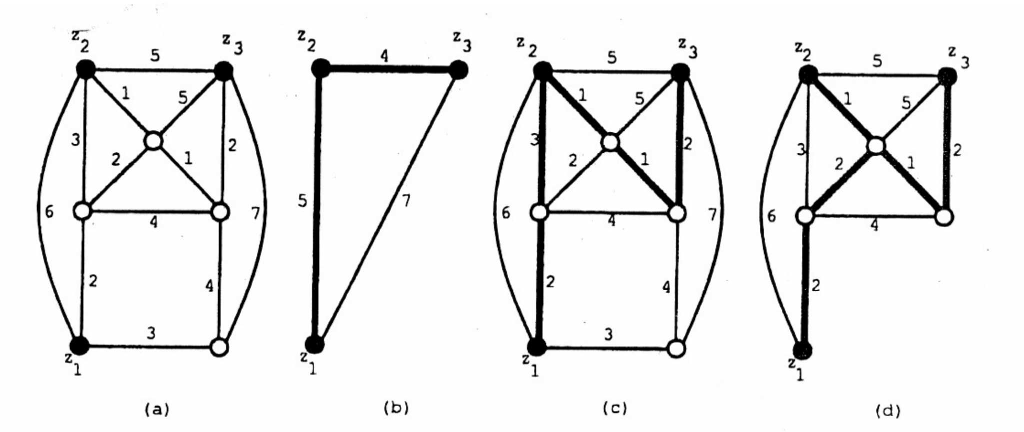
1. Kies een willekeurige terminal z_i als oorsprong.
2. Construeer vanuit z_i kortste paden naar alle andere terminals.

Deze verzameling van kortste paden vormen samen de oplossing volgens de kortste pad met oorsprong heuristiek. Deze boom wordt T_{SPOH} genoemd. Echter de zogenaamde worst-case fout ratio $|T_{SPOH}| / |T_G(N)|$ is gelijk aan n-1. Oftewel, een stuk slechter dan de normale kortste pad heuristiek. Ook de gemiddelde performance is een stuk slechter.

6.5 Afstandsnetwerk heuristiek

Een andere boom heuristiek is onafhankelijk beschreven door Choukhmane; Kou, Markowsky en Berman; Plesnik enIwainsky, Canuto, Taraszow en Villa. Het gaat om de zogenaamde afstandsnetwerk heuristiek. De stappen in deze heuristiek zijn als volgt:

1. Construeer een afstandsnetwerk $D_G(N)$ voor N in G.
   Zie figuur a voor de oorspronkelijke graaf en b voor het afstandsnetwerk $D_G(N)$ voor N in G
2. Bepaal een minimale opspannende boom voor $D_G(N)$
   Zie de dikgedrukte zijden in figuur b.
3. Vervang iedere zijde in de minimale opspannende boom door een kortste pad in G. Deze boom wordt T_D genoemd.
   Zie figuur c, de dikgedrukte zijden behoren tot T_D.
4. Bepaal een minimale opspannende boom T_{DNH} in het subnetwerk van G bepaald door de knopen uit T_D.
   Zie figuur d, merk op dat de boom verbeterd is ten opzichte van figuur c
5. Verwijder uit T_{DNH} één voor één de non-terminals met een graad van één.



figuur 6-2: Werking van de afstandsnetwerk heuristiek

In het slechtste geval is de rekentijd van deze heuristiek $O(nv^2)$. Ook hier geldt weer (net als bij de kortste pad heuristiek) dat dit veroorzaakt wordt door de kortste pad berekeningen. Deze zijn in dit geval nodig om het afstandsnetwerk te bepalen. Ook geldt weer (net als bij de kortste pad heuristiek) dat de worst-case foutratio gelijk is 2 – n/2. Oftewel $|T_{DNH}| / |T_G(N)| \leq 2 - n/2$

6.6 Overige heuristieken

Naast de hierboven beschreven boom en pad heuristieken zijn er nog talloze andere heuristieken bekend. In deze paragraaf zullen de belangrijkste andere typen heuristieken kort genoemd worden. Deze zullen echter niet zo gedetailleerd besproken worden als de hierboven beschreven heuristieken.

Knoop heuristieken: Knoop heuristieken zijn gebaseerd op het identificeren van een set 'goede' non-terminals, welke als Steiner knopen gebruikt kunnen worden. Het idee om te zoeken naar 'goede' non-terminals is gebaseerd op het feit dat het, bij het bepalen van een minimale Steiner boom, het grootste probleem is om de Steiner knopen te selecteren uit de verzameling van non-terminals. Immers, zodra de juiste Steiner knopen gevonden zijn, kan de gezochte boom met ieder opspannende boom algoritme geconstrueerd worden. Voorbeelden van knoopheuristieken zijn de zogenaamde 'gemiddelde afstands heuristiek' en enkele varianten daarop. Een andere knoop heuristiek is de zogenaamde upgrading heuristiek. Deze is voornamelijk van theoretisch belang aangezien het de enige heuristiek is waarbij de worst-case fout ratio onder de 2 ligt (namelijk 11/6 en zelfs 16/9 voor de aangepaste variant). Verder kan de heuristiek gegeneraliseerd worden zodat deze toegepast kan worden in iedere metrische ruimte.

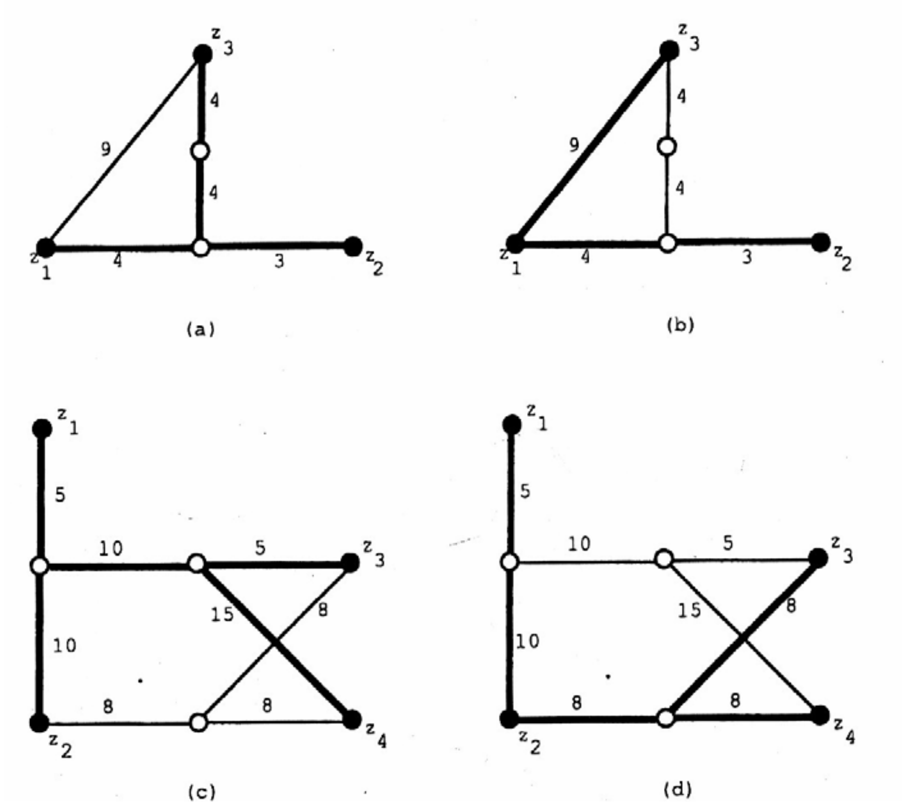
Samentrekkings heuristiek: Deze heuristiek is gebaseerd op een reductie, waarmee het netwerk G vereenvoudigd kan worden. Zelfs bij het bepalen van een optimale oplossing kan de volgende reductie namelijk worden toegepast. Wanneer de kortste zijde incident met een terminal z_i een verbinding is met een andere terminal z_j dan weten we zeker dat de zijde (z_i,z_j) opgenomen dient te worden in de optimale boom. Het is dan voldoende om het netwerk verkregen door samentrekking langs (z_i,z_j) te gebruiken voor de bepaling van de minimale Steiner boom. De samentrekkings heuristiek is gebaseerd op een nog algemenere vorm van samentrekking.

6.7 Vergelijk

Het is moeilijk een goede vergelijking tussen de verschillende heuristieken te maken of om een aanbeveling voor een bepaalde heuristiek te doen. Zowel Plesnik als Rayward-Smith hebben aangetoond dat wanneer er een vergelijking wordt gemaakt tussen twee heuristieken er (vrijwel) altijd probleeminstanties zijn te vinden waarop heuristiek A beter presteert dan heuristiek B en probleeminstanties waarop heuristiek B beter presteert dan heuristiek A.

Verder is aangetoond door Kucera, Marchetti-Spaccamela, Protasi en Talamo dat voor een random gekozen netwerk de afstandsnetwerk heuristiek Steiner bomen oplevert die vrijwel minimaal zijn. Echter zelfs de meest triviale methode, de kortste pad met oorsprong heuristiek levert vrijwel optimale oplossingen.

Over het algemeen levert de kortste pad heuristiek zelfs betere oplossingen dan de afstandsnetwerk heuristiek. Echter zoals hierboven al opgemerkt zijn er ook probleeminstanties te construeren waarin dit precies omgekeerd is. Dit wordt geïllustreerd door onderstaande figuur. A en B laten een situatie zien waarin T_{SPH} (a) korter is dan T_{DNH} (b) namelijk 15 tegen 16. Echter C en D laten een situatie zien waarin T_{SPH} (c) langer is dan T_{DNH} (d) namelijk 45 tegen 39.



figuur 6-3: Kortste pad vs. Afstandsnetwerk heuristiek

7 Samenvatting

Het Steiner boom probleem is een algemenere versie van het opspannende boom algoritme. Er zijn drie algemene varianten van het Steiner boom probleem, te weten:

1. Het Euclidische Steiner boom probleem
2. Het Steiner boom problemen in netwerken
3. Het Recti-lineaire Steiner boom probleem

In dit werkstuk zijn algoritmes en heuristieken voor de tweede genoemde variant, het Steiner boom probleem in netwerken, beschreven. Deze variant kent vele praktische toepassingen, voornamelijk in communicatie-, transport- en distributiesystemen.

De geschiedenis van het Steiner boom probleem loopt via Fermat, Torricelli en Simpson. Het probleem is door Courant en Robbins ten onrechte aan Jakob Steiner toegeschreven. Melzak heeft voor het eerst een algoritme ontwikkeld om het Euclidische probleem op te lossen.

Het Steiner boom probleem in netwerken is een NP-compleet probleem. Het bewijs van het NP-compleet zijn loopt via het *exact cover by 3-sets* probleem. Dit bewijs is oorspronkelijk gegeven door Karp en spitst zich toe op bipartite graven.

Vervolgens worden twee exacte algoritmes in detail behandeld, het zogenaamde opspannende boom enumeratie algoritme en een branch-and-bound algoritme. De tijd nodig om tot een optimale oplossing is voor beide algoritmes niet met een polynoom van het aantal knopen of het aantal terminals te beschrijven.

Vanwege deze niet polynomiale rekentijd is er behoefte aan geschikte heuristieken. In dit werkstuk worden een aantal heuristieken beschreven. De belangrijkste zijn de kortste pad heuristiek en de afstandsnetwerk heuristiek. Rayward en Smith hebben aangetoond dat er bij een vergelijk tussen heuristiek A en heuristiek B (vrijwel) altijd probleeminstanties zijn waarop A beter kan presteren dan B en omgekeerd. Een voorbeeld voor de kortste pad heuristiek en de afstandsnetwerk heuristiek is in dit werkstuk opgenomen

Literatuurlijst

Ball, M.O., Magnanti T.L., Monma, C.L. en Nemhauser, G.L. (1995) Network Models, *Handbooks in Operations research and management Science, volume 7, Elsevier, North-Holland*

Combinatoriek, grafentheorie en getaltheorie (deel 2), *Dictaat Open Universiteit*

Even, S (1979) Graph Algorithms, *Pitman publishing*

Goor, A.R. van, Ploos van Amstel, M.J. en Ploos van Amstel, W. (1999) Fysieke distributie: denken in toegevoegde waarde (vierde druk), *Educatieve Partners Nederland*

Hwang, F.K., Richards, D.S. en Winter, P. (1992) The Steiner Tree Problem, *Annals of discrete Mathematics, volume 53, Elsevier, North-Holland*