Faculteit Wetenschappen
Vakgroep Toegepaste Wiskunde en Informatica

Academiejaar 2009–2010

Communicatienetwerk voor het GPS III systeem

Simon Crevals

Promotor: Prof. dr. G. Brinkmann
Begeleider: N. Van Cleemput

Woord vooraf

Deze thesis ter afsluiting van een universitaire studie is een ideale gelegenheid om enkele mensen te bedanken die belangrijk geweest zijn tijdens mijn studie en in het bijzonder bij het tot stand komen van deze thesis.

In de eerste plaats zou ik graag mijn promotor, Prof. Dr. Gunnar Brinkmann, en begeleider Nicolas Van Cleemput willen bedanken voor hun uitstekende begeleiding en alle tijd die ze voor me vrij gemaakt hebben. Dit zowel in de vorm van regelmatige afspraken tijdens het onderzoek, als het nalezen van dit werk gedurende de laatste weken.

Verder wil ik John Frye bedanken voor de communicatie met Lockheed Martin. Hij zorgde voor de data die nodig was voor deze thesis en beantwoorde mijn vragen zodat ik de thesis van interessante achtergrondinformatie kon voorzien.

Ook wil ik mijn ouders bedanken voor hun steun en vertrouwen (dat zeker in het begin wel eens op de proef gesteld werd). Zij gaven mij de kans om verder te studeren.

Tenslotte wil ik ook mijn vrienden bedanken, in het bijzonder Jasper en Saya, voor hun eeuwige steun en hun bijdrage aan dit werk.

Abstract

This master thesis is to answer a question of John Frye, employee of Lockheed Martin. For the upcoming GPS III system, Lockheed Martin needs to build an intersatellite communication network. This kind of network is not present in the current GPS system since there is no communication between the satellites whatsoever. The communication network can be represented by a connection graph, indicating which satellites are connected to each other. Each satellite has 4 antennas to make connections to other satellites, so a connection graph has to be a 4-regular subgraph of the graph with all possible connections (satellites that are too close to each other or have the earth between them at any point in time can't be connected). A connection graph is acceptable if it has a diameter of at most 4 and has an acceptable angle of connections around each satellite. The quality of the angle can be described by a number called the uredop value of a satellite with respect to its neighbours. Smaller uredop values represent a better quality and uredop values smaller than 3 are considered acceptable.

Nevertheless there is a big difference between an acceptable and a good uredop value, and an acceptable diameter and the best diameter possible. Lockheed Martin would like to know the best connection graph G. In addition to being a connection graph, we would like G to have diameter 3, which is the theoretical lower bound, and the maximum uredop value over all its satellites should be as small as possible. As a little extra, it would be nice if the diameter of G would be 4 when one of its connections fails, which we prove to be the smallest possible.

After implementing our first ideas and running some tests, we got a better grasp of the difficulty of the problem. We found that the problem can be translated to an independent set problem. For the general case, the independent set problem is one of the NP-complete problems. We prove that our specific independent set problem remains NP-complete. We built a specialized program for our independent set problem that generates all independent sets corresponding to connection graphs with a given maximum allowed uredop value M. This branch and bound program builds connection graphs vertex by vertex, trying all possible combinations of connections corresponding to uredop values smaller than M. We optimized the program by using efficient datastructures and an efficient implementation. In the end the program found the best graphs within the hour, which is very fast.

Inhoudsopgave

Hoofdstuk 1

Inleiding

1.1 Achtergrond van het probleem

1.1.1 Oorsprong van het probleem

Het huidige GPS-systeem bestaat uit 24 satellieten, met het GPS III systeem zal dit uitgebreid worden naar 27 tot 32 satellieten. Tussen satellieten van het huidige GPS-systeem is geen communicatie mogelijk. In het nieuwe GPS III systeem wil men deze communicatie echter opzetten. Deze communicatie zou er namelijk toe leiden dat er enerzijds continu telemetrie van alle satellieten kan ontvangen worden en anderzijds dat elke satelliet op elk moment bediend kan worden. Dit laatste zal bijvoorbeeld gebruikt worden om verbeterde baanvoorspellingen te uploaden naar de satellieten en dit vaker dan de huidige tweemaal per dag. Deze regelmatigere updates geven aanleiding tot een preciezer systeem. We mogen aannemen dat er nog redenen zijn waarom het nuttig is op elk moment elke satelliet te kunnen beheren, maar daar kunnen we niet verder op ingaan.
Gezien de onderlinge posities van de satellieten, is er niet altijd rechtstreekse communicatie mogelijk tussen elk paar satellieten. Als op een bepaald punt in hun baan twee satellieten te dicht bij elkaar komen, of de aarde zich tussen beide satellieten bevindt, is er geen communicatie tussen deze satellieten mogelijk. Als we enkel verbindingen toelaten tussen satellieten waar deze situaties zich nooit voordoen, dan heeft elke satelliet nog mogelijkheid tot communicatie met slechts een beperkt aantal andere satellieten. In het GPS III systeem zou elke satelliet 4 vaste verbindingen naar satellieten uit deze mogelijkheden hebben; met andere woorden: elke satelliet is constant met dezelfde 4 satellieten verbonden. Hoewel het nog niet duidelijk is hoeveel satellieten het GPS III systeem precies zal bevatten, hebben we slechts de nodige data gekregen om het probleem op te lossen voor de constellatie met 27 satellieten. We zullen mogelijke communicatienetwerken voorstellen door connectiegrafen. Voor de definities uit de grafentheorie hebben we ons gebaseerd op Diestel (2005) en Brinkmann (2010).

Definitie. *V_{GPS} is de verzameling van alle 27 satellieten uit de constellatie.*

Definitie. $G_{Pos} = (V_{GPS}, E_{Pos})$, *waarbij* $\forall v, w \in V_{GPS} : \{v, w\} \in E_{Pos} \Leftrightarrow$ *tussen satellieten* v *en* w *is communicatie in principe mogelijk.*

Definitie. *Gegeven 2 grafen,* $G = (V, E)$ *en* $G' = (V', E') : G'$ *heet een* **deelgraaf** *van* $G \Leftrightarrow V' \subseteq V$ *en* $E' \subseteq E$.

Definitie. *Een* **connectiegraaf** *is een opspannende, 4-reguliere deelgraaf van* G_{Pos}*, dus met dezelfde toppenverzameling als* G_{Pos}*, namelijk* V_{GPS}.

1.1.2 DOP waarden

DOP staat voor Dilution Of Precision, vrij vertaald als vermindering van precisie. DOP waarden zijn getallen die de kwaliteit van de combinatie van de connecties rond 1 bepaalde satelliet beschrijven. Deze waarden zijn allemaal gebaseerd op de onderlinge positie van de satellieten. De posities van de satellieten ten opzichte van elkaar veranderen constant, maar omdat alle GPS-satellieten rond de aarde draaien in een periode van 12 uur en omdat de eerste helft van een periode symmetrisch is aan de tweede helft, moeten alle DOP waarden dus maar voor een halve periode berekend worden. De uredop waarden die wij zullen gebruiken zijn afgeleid van de verticale (VDOP) en positionele (PDOP) DOP waarden beschreven in (Langley, 1999), net zoals bij alle DOP waarden geldt dus dat een kleinere uredop waarde overeen komt met een betere kwaliteit. Uredop staat voor User Range Error Dilution Of Precision. De kwaliteit voorgesteld door de DOP waarden heeft betrekking tot de plaatsbepaling, een lagere DOP waarde komt overeen met een nauwkeurigere plaatsbepaling. In het communicatienetwerk van satellieten wil dit zeggen dat de uredop waarde van satelliet s de nauwkeurigheid voorstelt waarmee de buren van s de positie van s bepalen. Op deze manier kunnen heel betrouwbare plaatsbepalingen gedaan worden, die nodig zijn voor 'safety of life' toepassingen die bijvoorbeeld gebruikt worden voor het luchtverkeer en het landen van vliegtuigen.
De 95% uredop waarde slaat op het feit dat de uredop waarde van de verbindingscombinatie 95% van de tijd kleiner is dan deze 95% uredop. In plaats van 95% uredop werd ook voorgesteld met de maximum uredop waarde te werken, maar omdat het belangrijker is dat de verbindingscombinatie meestal goed is dan dat deze nooit slecht is, heeft Lockheed Martin gekozen voor 95% uredop om de kwaliteit van de verbindingscombinatie te representeren. We kunnen echter ook alles voor de maximum uredop waarde doen, of voor andere waarden die de kwaliteit van de verbindingscombinaties representeren. Afhankelijk van welke 4 satellieten met de satelliet verbonden zijn, hebben we dus een andere 95% uredop waarde. Voor elke satelliet en elke mogelijke combinatie van 4 satellieten waarmee deze satelliet verbonden is, hebben we de corresponderende 95% uredop waarde gekregen. Vanaf nu zullen we voor de eenvoud steeds uredop schrijven in plaats van 95% uredop. Met de uredop waarde van de graaf bedoelen we het maximum van alle 27 uredop waarden die in de graaf voorkomen.

Definitie. *Gegeven een graaf* $G = (V, E)$ *en een top* $v \in V$ *: de* **omgeving** *van* v *in* G *is* $N(v, G) = \{w \in V | \{v, w\} \in E\}$.

Definitie. *De functie uredop, van een top v met respect tot zijn buren, is gedefiniëerd als: Zij* $M_4 = \{M \subseteq V_{GPS} || M| = 4\}$.
$uredop : V_{GPS} \times M_4 \rightarrow \mathbb{R}^+ \bigcup \{\infty\}$, *waarbij geldt dat* $\forall v \in V_{GPS} : M \not\subset N(v, G_{Pos}) \Rightarrow uredop(v, M) = \infty$.

Definitie. *De uredop waarde van een top v in een connectiegraaf G is gedefiniëerd als:* $uredop(v, G) = uredop(v, N(v, G))$.

Definitie. *De uredop waarde van een connectiegraaf G is gedefiniëerd als:* $uredop(G) = max(\{v \in V_{GPS} | uredop(v, G)\})$.

1.2 Probleembeschrijving

De bedoeling van deze thesis is om de beste connectiegraaf tussen de 27 satellieten uit V_{GPS} te construeren. Om tot een aanvaardbare graaf te komen waren enkele restricties gegeven. Allereerst moet deze graaf 4-regulier zijn. Verder is het natuurlijk belangrijk dat we enkel verbindingen gebruiken tussen satellieten waar onderlinge communicatie altijd mogelijk is. De graaf moet dus een deelgraaf zijn van G_{Pos}. Merk nogmaals op dat als een graaf G aan deze eerste twee restricties voldoet, we G een connectiegraaf noemen. Ten derde mag de diameter ten hoogste 4 zijn. En als laatste restrictie moet de uredop waarde van de graaf kleiner zijn dan 3. Aan al deze restricties moet voldaan zijn. We zullen deze restricties nu iets meer in detail bekijken om te zien hoe we tot een beste graaf kunnen komen.

1.3 Restricties in detail

1.3.1 4-regulier

De graaf die we construeren moet 4-regulier zijn. Dit omdat elke satelliet 4 antennes zal hebben om met andere satellieten te communiceren en is ons dan ook opgelegd. De grafen die we construeren moeten hier dus aan voldoen, willen ze een mogelijke oplossing zijn voor het probleem. We kunnen deze restrictie niet strenger maken om zo tot een betere graaf te komen, onze beste graaf zal hier dus ook gewoon aan moeten voldoen.

1.3.2 Mogelijke verbindingen

Voor elke satelliet $s \in V_{GPS}$ geldt dat $9 \leq |N(s, G_{Pos})| \leq 12$. Er zijn dus minimum 9 en maximum 12 andere satellieten die steeds in een gunstige positie ten opzichte van s liggen. Het exacte aantal is afhankelijk van welke satelliet we precies beschouwen. Als we met al deze mogelijke verbindingen een graaf bouwen, krijgen we net de graaf die we eerder gedefinieerd hebben als G_{Pos}. We kunnen natuurlijk geen verbindingen toelaten die niet mogelijk zijn, dus

hebben we ook hier een eis waar we aan moeten voldoen, anders hebben we geen juiste oplossing voor het probleem. De grafen die we construeren moeten dus deelgrafen zijn van G_{Pos}. Deze eis strenger maken is niet zo moeilijk, we zouden namelijk enkele mogelijke verbindingen kunnen schrappen. We hebben hier echter geen reden om aan te nemen dat bepaalde verbindingen tot slechte grafen zullen leiden. Het is dus ook niet zinvol om verbindingen te schrappen. We zullen uit deze voorwaarde dan ook geen verdere beperkingen halen om tot een beste graaf te komen.

1.3.3 Uredop waarden

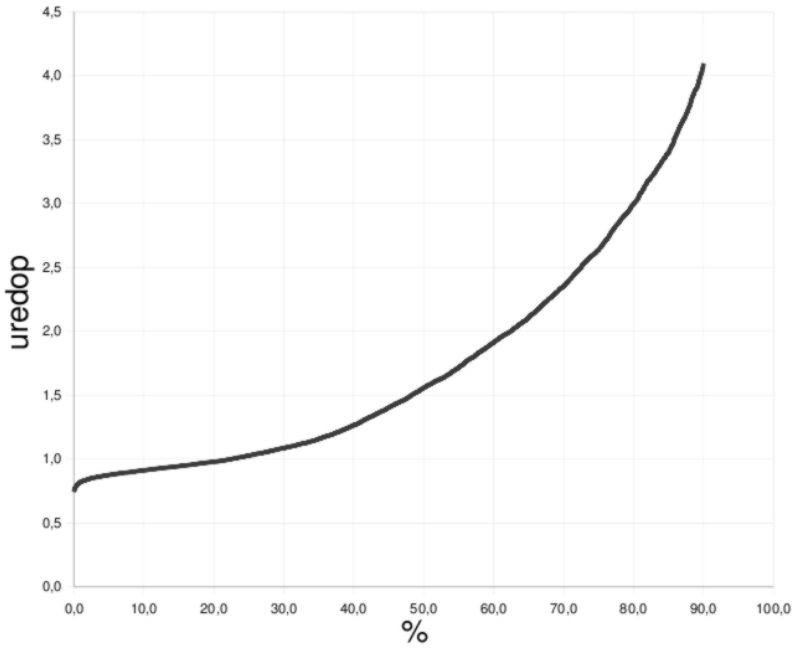
De maximale uredop waarde in een connectiegraaf mag ten hoogste 3 zijn om een aanvaardbare graaf te zijn, maar hoe kleiner de maximale uredop waarde hoe beter. Het is dus eerst en vooral belangrijk om te zien of we deze grens van 3 kunnen halen. Als we oplossingen vinden met uredop kleiner dan 3, geldt natuurlijk nog steeds dat een kleinere uredop waarde van de graaf overeenkomt met een betere oplossing, enkel gelet op uredop. De beste graaf zal dus de kleinste uredop waarde moeten hebben, gegeven bepaalde eigenschappen.
De verdeling van de uredop waarden is als volgt. Per satelliet s zijn er $126 \leq \frac{N(s)!}{4!(N(s)-4)!} \leq 495$ mogelijke verbindingscombinaties en in totaal zijn er 7842 verbindingscombinaties. Drie kwart van de connectiecombinaties heeft een uredop waarde kleiner dan 3, iets minder dan de helft heeft een uredop waarde kleiner dan 1.5 en iets minder dan een kwart heeft een uredop waarde kleiner dan 1.0 (zie Figuur 1.1). Let op: dit wil nog niet zeggen dat er grafen bestaan met uredop waarde kleiner dan 1 of zelfs 1.5.

1.3.4 Diameter

Een diameter van 4 werd geëist, maar voor een algemene 4-reguliere graaf van 27 toppen is diameter 3 ook nog haalbaar. Een 4-reguliere graaf met diameter 2 heeft ten hoogste 17 toppen. De diameter van een communicatienetwerk is gelijk aan het aantal verbindingen dat maximaal nodig is om een boodschap tussen twee willekeurige satellieten te versturen, een kleinere diameter is dus een groot voordeel. Indien diameter 3 haalbaar is binnen de andere restricties zou dit al zeker een betere graaf opleveren. Voor een communicatienetwerk is niet alleen de diameter van belang, maar ook wat er met de diameter gebeurt als er één link zou wegvallen. Liefst zouden we hebben dat de graaf dan nog steeds een zo klein mogelijke diameter heeft.

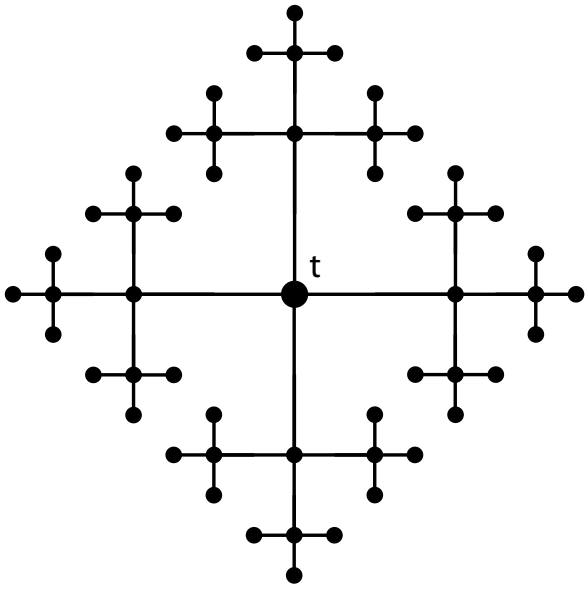
Definitie. *Een* **wandeling** *van lengte n is een opeenvolging $v_0, e_1, v_1, e_2, ..., v_n$, waarbij v_i toppen en e_i bogen zijn, zodat $e_i = \{v_{i-1}, v_i\}$ en $\forall i,\ 1 \leq i \leq n-1 : e_i \neq e_{i+1}$.*

Stelling. *Gegeven een 4-reguliere graaf $G = (V, E) : |V| \geq 26 \Rightarrow \exists e \in E : diameter((V, E \backslash \{e\})) \geq 4$.*



Figuur 1.1: Eerste 90 % van de connectiecombinaties met hun uredop waarde

Bewijs. Stel $G = (V, E)$, een 4-reguliere graaf: $|V| \geq 26$ en $\forall e \in E : diameter((V, E \backslash \{e\})) = 3$.
Stel bovendien dat we een willekeurig gekozen boog $e = \{v, w\}$ zouden verwijderen. Dan moet er nog een pad zijn tussen v en w met lengte kleiner of gelijk aan 3, anders is de diameter van $(V, E \backslash \{\})$ groter dan 3. Dus ligt e op minstens één cykel van lengte 3 of 4, en gezien e willekeurig was, geldt dit voor elke boog. G is 4-regulier, dus vertrekken vanuit elke top 4 bogen en ligt elke top dus ook op minstens twee cykels van lengte 3 of 4 . *
Kies nu een vaste top $t \in V$. Omdat G 4-regulier is, zijn er vanuit t 4 wandelingen van lengte 1, 12 van lengte 2 en 36 van lengte 3 (zie Figuur 1.2); dit zorgt voor een totaal van 52 wandelingen van lengte kleiner dan of gelijk aan 3. Deze wandelingen kunnen bogen gemeenschappelijk hebben, of zelfs volledig dezelfde bogen gebruiken, bijvoorbeeld als begin- en eindpunt gelijk zijn, maar de volgorde tegengesteld. Elke top s verschillend van t moet via minstens twee van deze wandelingen bereikt kunnen worden. Als deze twee wandelingen een boog gemeenschappelijk hebben, zouden we net die boog kunnen weglaten, dus dan moet s zelfs via minstens 3 van deze wandelingen bereikt kunnen worden. **
Als een boog incident met t op een cykel van lengte 3 ligt, dan bereiken de twee wandelingen van lengte 3 in beide richtingen over deze cykel enkel t. Als een boog incident met t op een cykel van lengte 4 ligt, dan kunnen vanuit de top op afstand 2 van t op de cykel nog twee andere toppen bereikt worden via een wandeling van lengte 3 (zie Figuur 1.3). Voor beide toppen bestaan echter twee wandelingen, namelijk langs elke kant van de cykel, die de



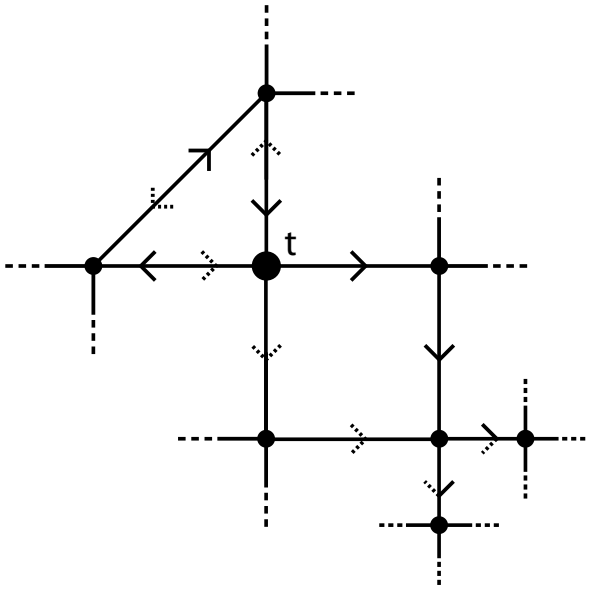
Figuur 1.2: De 52 mogelijke wandelingen op afstand 3 of minder van t. Sommige van de hier afgebeelde toppen kunnen dezelfde zijn.

laatste boog gemeenschappelijk hebben. Naar beide toppen is dus nog minstens één andere wandeling nodig (zie **).
Er zijn minstens twee cykels van lengte 3 of 4 met bogen incident met t (zie *), dus zijn er minstens vier wandelingen vanuit t, die naar t gaan of naar toppen waarnaar een extra wandeling, naast de twee gekende wandelingen, nodig is.
Er is gesteld dat $|V| \geq 26$, dus zonder t zijn er nog minstens 25 toppen, die via minstens twee wandelingen bereikt moeten worden. Dit zijn dus minstens 50 wandelingen. Met de vier wandelingen bij die terug naar t gaan of naar een top waarnaar een extra wandeling nodig is, hebben we dus minstens 54 verschillende wandelingen nodig met lengte kleiner dan of gelijk aan 3, maar er bestaan er maar 52. Er bestaat dus geen 4-reguliere graaf met 26 of meer toppen zodat de diameter 3 is na het verwijderen van eender welke boog. □

Indien mogelijk willen we een graaf met diameter 3, die na het verwijderen van één boog ten hoogste diameter 4 kan hebben.

1.4 Beste graaf

Om tot een beste graaf te komen moeten vele dingen tegen elkaar afgewogen worden. Eerst en vooral willen we de graaf vinden met de kleinste uredop waarde, die ook aan alle basisrestricties voldoet, dit zal in ieder geval een mogelijke beste graaf zijn. Als deze al diameter 3 heeft, is dat prachtig, maar anders moeten we deze vergelijken met de graaf met diameter



Figuur 1.3: Cykel met lengte 3 en cykel met lengte 4

3 met de kleinste uredop waarde. We moeten dan beslissen of de verbetering van diameter 4 naar diameter 3 opweegt tegen de stijging in uredop. Binnen de grafen met diameter 3 en deze met diameter 4 moeten we eenzelfde vergelijking maken tussen deze met maximum diameter 4 na het verwijderen van één boog en de anderen. Voor elke verzameling van eigenschappen zullen we dus een beste graaf krijgen, hieruit moet dan besloten worden welke de beste graaf is. Het is echter niet voor de hand liggend om te bepalen of de uredop waarde belangrijker is dan de diameter (na het wegvallen van een boog), gezien de achterliggende functies totaal verschillend zijn. Een kleinere uredop waarde zorgt voor een betere prestatie van het GPS systeem en een kleinere diameter zorgt voor minder vertraging bij het bedienen van de satellieten en het ontvangen van de telemetrie. Het is echter niet aan ons om deze beslissing te maken en we zullen dan ook proberen voor elke verzameling van eigenschappen een beste graaf te vinden.

Hoofdstuk 2

Eerste test

2.1 Introductie

Eerst en vooral is het belangrijk een goed zicht te krijgen op het probleem. We willen bijvoorbeeld liefst dat onze beste graaf diameter 3 heeft, maar bestaan er wel 4-reguliere grafen met diameter 3 die ook aan de andere restricties voldoen? Op het moment dat we met dit onderzoek begonnen, hadden we nog niet de uredop waarden waarmee we uiteindelijk werken. Deze eerste test is uitgevoerd met de VDOP waarden als kwaliteitsmaat voor de verbindingscombinaties. Deze VDOP waarden kunnen natuurlijk volledig analoog aan de uredop waarden gedefinieerd worden als een functie van een top en zijn omgeving.

2.2 Idee

Om te weten te komen of er 4-reguliere grafen zijn met diameter 3 die ook deelgraaf zijn van G_{Pos} hebben we met genreg (Meringer, 1999) 4-reguliere grafen met 27 toppen isomorf-vrij gegenereerd en grafen met diameter 3 eruit gefilterd, tot we enkele honderden grafen hadden. Dan hebben we een afbeelding $f : V \to V_{GPS}$ gezocht die aantoont dat deze grafen deelgraaf zijn van G_{Pos} onder f.

Definitie. *Als $G' = (V', E')$ en $G = (V, E)$ grafen zijn, f een bijectieve afbeelding van V naar V'' en bovendien $V'' \subseteq V'$ en $E'' \subseteq E'$, met $E'' = \{\{f(v), f(w)\} | \{v, w\} \in E\}$, dan heet G een deelgraaf van G' onder het isomorfisme f. We noteren de graaf (V'', E'') als $f(G)$.*

Als we een isomorfisme f vinden waarvoor de connectiegraaf G een deelgraaf is van G_{Pos} onder f, dan kunnen we makkelijk de VDOP waarde van $f(G)$ bepalen. Een graaf G kan echter deelgraaf zijn van G_{Pos} onder verschillende isomorfismen en dus verschillende bijhorende VDOP waarden hebben. Omdat we ook een idee willen krijgen van de kleinst mogelijke VDOP waarde, stoppen we niet met zoeken voordat we zeker zijn dat we het isomorfisme b gevonden hebben zodat $VDOP(b(G))$ de kleinste waarde oplevert onder alle isomorfismen f

waarvoor $f(G)$ deelgraaf is van G_{Pos}. Nadat we voor G het beste isomorfisme b gevonden hebben, hoeven we voor de volgende grafen G' met diameter 3 die we behandelen enkel nog te zoeken naar isomorfismen b' zodat $VDOP(b'(G')) < VDOP(b(G))$. Eenmaal we weten dat er een isomorfisme f bestaat zodat een connectiegraaf G deelgraaf is van G_{Pos} onder f, zijn we namelijk enkel nog geïnteresseerd in een isomorfisme b en een connectiegraaf G, zodat $VDOP(b(G))$ zo klein mogelijk is.

2.3 Deelgraaf algoritme

2.3.1 Isomorfismen opbouwen

Het deelgraafalgoritme dat bepaalt of $G = (V, E)$ een deelgraaf is van G_{Pos}, is een branch and bound algoritme dat isomorfismen van V naar V_{GPS} bouwt. We starten met een afbeelding van de lege verzameling naar V_{GPS}. Een recursiestap van het algoritme ziet er als volgt uit: uit de vorige recursiestap hebben we een afbeelding $f' : V' \to V_{GPS}$, waarbij $V' \subset V$. Voor elke $p \in V \backslash V'$ zullen we, voor we naar de volgende recursiestap gaan, f' uitbreiden tot een afbeelding $f'' : V' \bigcup \{p\} \to V_{GPS}$; $f''(v) = f'(v)$ als $v \in V'$ en p wordt afgebeeld op de top $t \in V_{GPS} \backslash F$, met $F = \{s | \exists v \in V' : f'(v) = s\}$ en waarvoor $|N(t, G_{Pos}) \bigcap F|$ zo klein mogelijk is. Als er meerdere mogelijkheden voor t zijn, nemen we deze met minimale graad. Gezien G_{Pos} niet verandert, kunnen we de volgorde waarin de toppen van V_{GPS} als beeld gekozen worden op voorhand bepalen.

2.3.2 Snoeicriteria

Deelgraaf van G_{Pos}

Als we een top s van G afbeelden op $p \in V_{GPS}$ en er in G een boog is van s naar t, waarvoor $f(t) = q \in V_{GPS}$ reeds bepaald is, dan moet gelden dat $\{p, q\} \in E_{Pos}$. Anders kan G nooit een deelgraaf zijn van G_{Pos} (zie definitie deelgraaf) en kunnen we backtracken. Om op deze manier te snoeien is het dus een vereiste dat $\{p, q\} \notin E_{Pos}$. We zullen bijgevolg rapper dergelijke bogen vinden bij toppen van G_{Pos} met weinig buren bij de reeds vastgelegde toppen en weinig buren in het algemeen. De vertakking van het branch and bound algoritme zo klein mogelijk houden, is een algemene techniek voor het ontwikkelen van efficiënte branch and bound algoritmen. De vertakking bepaalt een groot deel van de efficiëntie van het algoritme, vandaar dat de volgorde ook erg belangrijk is.

4-regulariteit

Stel dat we willen bepalen of $G = (V, E)$ een deelgraaf is van G_{Pos}. Voor elke top $p \in V$ is gekend dat $|N(p, G)| = 4$. We weten in elke recursiestap welke toppen van V reeds een beeld hebben onder de afbeelding f', deze zitten in de verzameling V'. We kennen dus ook

de buren die element zijn van V' en noemen deze verzameling $B = \{b|b \in N(p,G) \bigcup V'\}$. Door het vorige snoeicriterium moet $f''(p)$, met f'' de afbeelding die we in deze stap bouwen, verbonden zijn met het beeld van alle buren uit B. Voor alle buren $c \in N(p,G)\backslash B$ moet nu nog een boog mogelijk zijn naar een top $t \in V_{GPS}\backslash F$, met $F = \{s|\exists v \in V' : f'(v) = s\}$, dit wil zeggen dat $f''(p) = t \Rightarrow |N(p,G)\backslash B| \leq |N(t, G_{Pos}\backslash F|$ moet gelden. Als dit niet zo is, zullen we verder in het algoritme sowieso aan het vorige snoeicriterium voldoen. Door vroeger te snoeien, wordt minder vertakt. Dit is dus zeker ook een nuttig snoeicriterium.

2.4 Resultaten

Tegen de verwachtingen in bleek dat we voor bijna elke 4-reguliere graaf G met diameter 3 een isomorfisme konden vinden waarvoor G een deelgraaf was van G_{Pos}. Met deze verschillende grafen en hun isomorfismen corresponderen wel héél uiteenlopende VDOP waarden. Vele VDOP waarden waren echter niet slecht. De beste VDOP waarde die we zo gevonden hebben voor een connectiegraaf met diameter 3, is 1.38. Eerder onderzoek naar dit soort grafen was reeds binnen Lockheed Martin zelf gedaan en hun beste resultaat was een connectiegraaf met diameter 4 en een VDOP waarde van 1.75.

2.5 Conclusie

Hoewel het resultaat dat op deze manier gevonden is niet slecht leek –onze enige vergelijking was het resultaat dat eerder behaald werd door Lockheed Martin–, verwachtten we niet dat dit de beste graaf is. Er zijn namelijk miljarden 4-reguliere grafen met 27 toppen en diameter 3 en we hebben slechts op een klein deel van deze grafen het subgraph algoritme uitgevoerd. Beslissen of er een isomorfisme f bestaat zodat $f(G)$ deelgraaf is van G', is namelijk een NP-compleet probleem (Garey & Johnson, 1979), en ook voor ons specifiek geval bleek het controleren op deelgraaf zijn niet echt efficiënt. De tijd die het programma nodig heeft per graaf in combinatie met het grote aantal grafen dat getest moet worden, zorgt er natuurlijk voor dat we op deze manier nooit binnen een redelijke tijd de beste graaf zullen vinden.
Voor bijna elke graaf G die we controleerden, vonden we een isomorfisme f, zodat G een deelgraaf is van G_{Pos} onder f. Gezien dit volledig onafhankelijk is van het feit dat we nog de VDOP waarden gebruikten, weten we in ieder geval dat er veel kans is dat we goede grafen zullen vinden met diameter 3. Het feit dat we een connectiegraaf moeten hebben is dus niet echt beperkend. Ook met diameter 3 hebben we nog enorm veel mogelijkheden. De grootste beperking zal dus van de gewichten moeten komen, zij het VDOP waarden of uredop waarden, en we zullen een manier moeten vinden om deze connectiegrafen snel te construeren.

2.6 Slot

Nadat we een connectiegraaf met diameter 3 en VDOP waarde 1.38 gevonden hebben, hebben we deze ook doorgegeven aan Lockheed Martin. Nu bleek echter dat hoewel de VDOP waarde heel goed was, de PDOP waarde van deze graaf heel slecht was. Tot dan toe was er echter nog niet gesproken over de PDOP waarden. We wisten dus niet dat dit een eis was voor de connectiegrafen en konden dit dan ook niet voorzien. Bij Lockheed Martin besloot men toen om ons een nieuwe waardemeter te geven, de uredop waarden, die afgeleid zijn van zowel de VDOP als de PDOP waarden. Gezien de conclusies die we in dit hoofdstuk al hebben kunnen trekken, leek het niet nuttig al deze grafen te testen met de nieuwe waarden, maar in het verder onderzoek gebruiken we natuurlijk wel de uredop waarden als nieuwe waardemeter.

Hoofdstuk 3

Uiteindelijk programma

3.1 Algemeen idee

Uit het vorige hoofdstuk volgt dat als we in redelijke tijd de beste oplossing willen vinden, een grote beperking van de uredop waarden moet komen. We zullen dus connectiegrafen genereren met uredop waarde kleiner dan een gegeven bovengrens. We houden voor elke top bij welke verzamelingen van buren overeenkomen met toegelaten uredop waarden. Op deze manier worden al heel veel mogelijke combinaties geschrapt bij het kiezen van een goede bovengrens. Aan de hand van de toegelaten combinaties van buren voegen we dan bogen toe. Telkens we een boog tussen twee toppen s en t toevoegen, zullen we de mogelijke combinaties van s en t moeten aanpassen, gezien voor s enkel nog combinaties mogelijk zijn waarbij t een buur is van s en analoog voor de combinaties van t. Als aan een top s vier buren toegekend zijn, dan moet de mogelijkheid geschrapt worden om s toe te voegen als buur aan een top u, die niet één van de 4 buren van s is. Alle combinaties van u waar s in de burenverzameling voorkomt, worden dus geschrapt uit de mogelijke combinaties van u. Om een van deze redenen kan het zijn dat een top geen mogelijke combinaties meer heeft. Dan is het duidelijk tijd voor een bound en een stap terug in de recursie.
We kunnen deze methode op twee manieren toepassen: enerzijds door één boog per keer toe te voegen, anderzijds door voor een top een van de mogelijke combinaties te selecteren en dan al deze bogen tegelijk toe te voegen.

3.2 Boog-per-boog methode

In deze methode wordt elke recursiestap in twee delen opgesplitst, één deel met een geselecteerde boog en één deel waarin deze geselecteerde boog niet meer mag voorkomen. Dit zorgt ervoor dat onze recursiediepte niet op voorhand gekend is, maar wel beperkt is door het aantal bogen van G_{Pos}. Een pad van een blad naar de wortel zal steeds $2|V_{GPS}|$ stappen bevatten waarin een boog aanvaard wordt, maar kan daarnaast ook nog heel wat stappen

bevatten waarin bogen geweigerd worden. We zullen steeds de boog selecteren tussen de top s, waarvoor nog niet alle buren vastliggen en die het minste mogelijke combinaties over heeft, en de buur van s die in de meeste van deze combinaties voorkomt. Dit houdt de boom zo klein mogelijk, omdat er bij het grootste aantal combinaties een boog vastgelegd wordt, en eens er vier bogen aan een top vastgelegd zijn, is deze top afgewerkt. De graaf wordt dus steeds voor zoveel mogelijk combinaties tegelijk opgebouwd, en ter zelfder tijd beperken we de vertakking in het begin van het algoritme door steeds de top met het kleinste aantal mogelijke combinaties te nemen. Het voordeel van deze methode is dat we soms meerdere combinaties rond s tegelijk kunnen schrappen. Als we een boog $\{s,t\}$ toevoegen die ervoor zorgt dat aan een snoeiciterium voldaan is, kunnen we alle combinaties van s waar t in de burenverzameling voorkomt, schrappen.

3.3 Independent set methode

3.3.1 Independent set

Definitie. *Gegeven een graaf $G = (V, E)$. Een toppenverzameling $IS \subseteq V$ is een* **independent set** *in $G \Leftrightarrow \forall v, w \in IS : \{v, w\} \notin E$.*

Definitie. *Gegeven een graaf $G = (V, E)$. en toppenverzameling $C \subseteq V$ is een* **kliek** *in $G \Leftrightarrow \forall v, w \in C : \{v, w\} \in E$.*
Als er geen kliek C' bestaat waarvoor $|C'| > |C|$, dan heet C een maximum kliek in G.

Het is duidelijk dat een independent set in $G = (V, E)$ ook een kliek is in het complement $G^c = (V, \binom{V}{2} \backslash E)$ van G en omgekeerd. Bepalen of een graaf G een independent set of kliek heeft van minstens k toppen, is een gekend NP-compleet probleem (Garey & Johnson, 1979). We kunnen het vinden van connectiegrafen met uredop kleiner dan u vertalen naar het vinden van klieken van 27 toppen in een graaf $G_C(u)$.

Definitie. *$G_C(u) = (V_C(u), E_C(u))$, waarbij $V_C(u) = \{(s, N) | s \in V_{GPS}, N \subset N(s, G_{Pos}), |N| = 4$ en $uredop(s, N) < u\}$ en $E_C(u) = \{\{v, w\} | v = (s, N_s) \in V_C(u), w = (t, N_t) \in V_C(u), s \neq t$ en $s \in N_t \Leftrightarrow t \in N_s\}$.*

Stelling. *De grootste klieken in $G_C(u)$ hebben ten hoogste 27 toppen.*

Bewijs. We hebben $|V_{GPS}| = 27$. Stel $|C| > 27$, met C een kliek in $G_C(u) \Rightarrow \exists s \in V_{GPS}, N_1, N_2 \in N(s, G_{Pos}) : (s, N_1), (s, N_2) \in C$, maar aangezien $s = s$, volgt uit de definitie van $G_C(u)$ dat $\{(s, N_1), (s, N_2)\} \notin E_C(u)$ en dus kan C geen kliek zijn in $G_C(u)$. □

Stelling. *Elke connectiegraaf $G = (V_{GPS}, E)$ met uredop kleiner dan u, komt overeen met een maximum kliek in $G_C(u)$.*

Bewijs. G is 4-regulier $\Rightarrow \forall v \in V_{GPS} : |N(v, G)| = 4$.
Neem $C = \{(v, N(v, G))|v \in V_{GPS}\}$, dus $|C| = 27$.
$uredop(G) < u \Rightarrow \forall v \in V_{GPS} : uredop(v, N(v, G)) < u \Rightarrow \forall v \in V_{GPS} : (v, N(v, G)) \in V_C(u)$.
$\forall v, w \in V_{GPS} : \{v, w\} \in E \Leftrightarrow v \in N(w, G)$ (en equivalent $w \in N(v, G)$) $\Rightarrow \{(v, N(v, G)), (w, N(w, G))\} \in E_C(u)$.
$\forall v, w \in V_{GPS} : \{v, w\} \notin E \Leftrightarrow v \notin N(w, G)$ (en equivalent $w \notin N(v, G)$) $\Rightarrow \{(v, N(v, G)), (w, N(w, G))\} \in E_C(u)$.
$\Rightarrow C$ is een maximum kliek in $G_C(u)$. □

Stelling. *Elke maximum kliek C in $G_C(u)$, met $|C| = 27$, komt overeen met een connectiegraaf met uredop kleiner dan u.*

Bewijs. Bouw $G = (V_{GPS}, E)$, met $E = \{\{v, w\}|\exists N_1 \subset N(v, G_{Pos}) : (v, N_1) \in C$ en $w \in N_1$ en $\exists N_2 \subset N(w, G_{Pos}) : (w, N_2) \in C$ en $v \in N_2\}$.
$\forall v \in V_{GPS} : \exists N \subset N(v, G_{Pos}) : (v, N) \in C$, want $|C| = 27$. Anders kunnen we namelijk een top vinden die in 2 verschillende elementen van C als eerste top voorkomt, wat niet kan.
Merk op dat uit definitie van $V_C(u)$ en kliek volgt dat $|N| = 4$.
Uit de definities van $E_C(u)$ en E volgt dat we v met alle elementen van N verbinden en dus geldt dat $\forall v \in V_{GPS} : uredop(v, N) < u$ en bijgevolg ook dat $uredop(G) < u$.
Uit de definitie van $V_C(u)$ volgt dat we een deelgraaf van G_{Pos} gebouwd hebben. G is dus een connectiegraaf met uredop kleiner dan u. □

We hebben nu aangetoond dat ons probleem een deelprobleem is van het independent set probleem. Het heeft dus de smaak van een NP-compleet probleem, maar om aan te tonen dat ons probleem nog steeds NP-compleet is, moeten we een gekend NP-compleet probleem reduceren op ons probleem. We zullen het Hamiltoniaanse cykel probleem, een ander gekend NP-compleet probleem (Garey *et al.*, 1976), reduceren op ons probleem omdat dit een gemakkelijker bewijs toelaat.

Definitie. *Een* **Hamiltoniaanse cykel** *H in een graaf G, is een cykel in G die elke top van G juist één keer bezoekt. Met andere woorden, H is een 2-reguliere, samenhangende, opspannende deelgraaf van G.*

Definitie. *De taal* **HC** *is de verzameling van alle strings s die een graaf G coderen, waarbij G een Hamiltoniaanse cykel bevat.*

Definitie. *De taal* **Uredop** *is de verzameling van alle strings van de vorm $g'\#c\#u$, waarbij g' een graaf G' codeert, c voor elke top de mogelijke verzamelingen van 4 buren met een bijhorende waarde (uit $\mathbb{N}$) codeert en u de binaire voorstelling is van een natuurlijk getal U en waarvoor geldt dat G een 4-reguliere, opspannende, samenhangende deelgraaf G bevat, zodat geldt dat $C(G) < U$. Waarbij de functie C analoog aan de functie uredop gedefiniëerd wordt, gebaseerd op de verzamelingen van buren en hun bijhorende waarde gecodeerd door c.*

Merk op dat de beelden van de functie *uredop* elementen zijn van $\mathbb{R}^+$, maar omdat we ervan uitgaan dat we deze getallen tot op een vast aantal cijfers na de komma kennen, kunnen we deze vervangen door natuurlijke getallen. Als we de uredop waarden kennen tot op 8 cijfers na de komma, wordt een uredop waarde w vervangen door het natuurlijk getal $w \times 10^8$, analoog zal de bovengrens m vervangen worden door $m \times 10^8$.

Stelling. *De taal Uredop is NP-compleet.*

Bewijs. Het is gemakkelijk om aan te tonen dat Uredop $\in$ NP is:

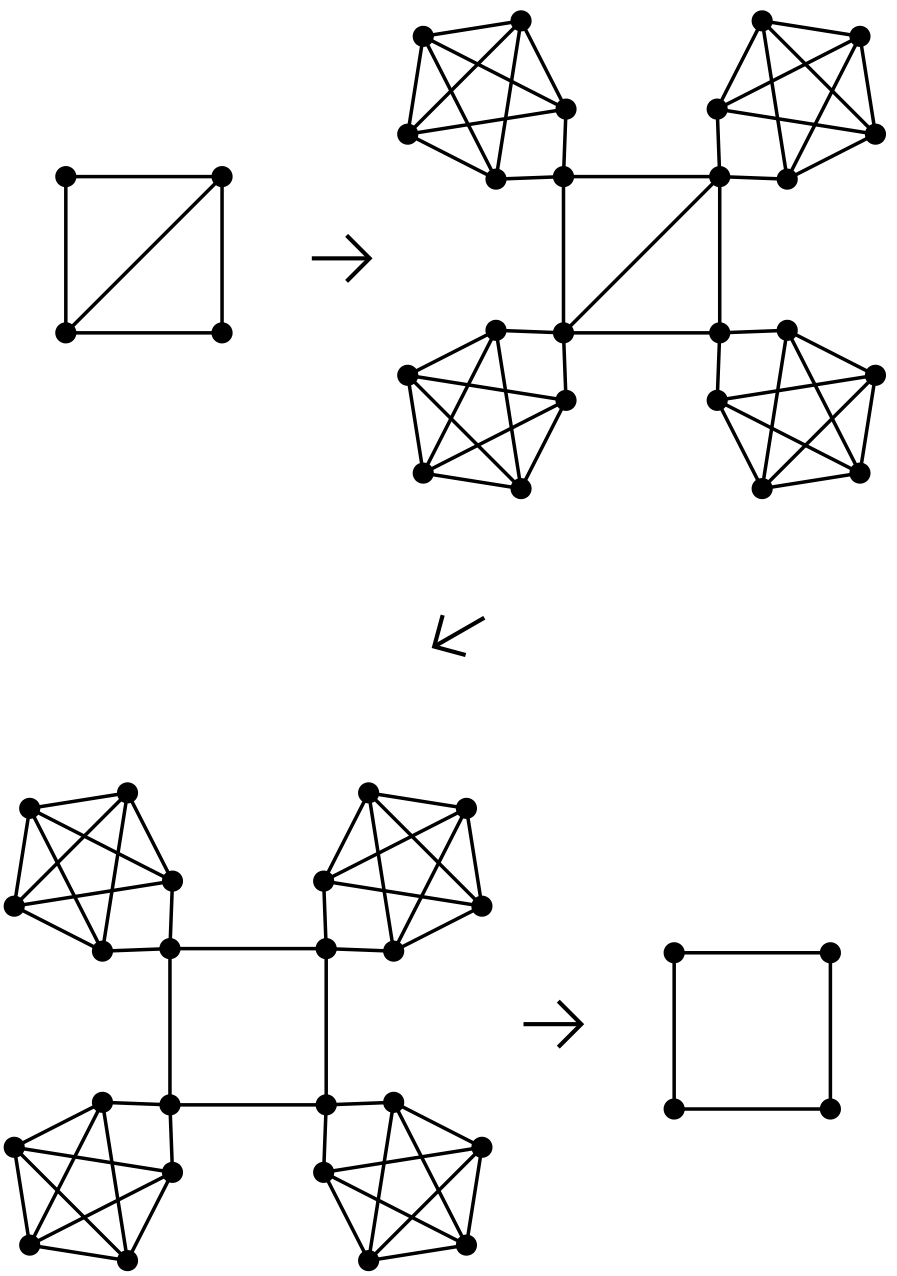
1. test of de input een graaf $G' = (V', E')$, een getal U, en een lijst L van zestallen codeert. Indien niet: zeg nee en stop.

2. kies een verzameling E van $2|V'|$ bogen en test of de graaf $G = (V', E)$ 4-regulier en samenhangend is. Indien niet: zeg nee en stop.

3. test voor elke top v uit V' of L een zestal bevat met v als eerste getal, de elementen van $N(v, G)$ als volgende vier getallen en met laatste getal kleiner dan U. Indien niet: zeg nee en stop.

4. zeg ja en stop.

We moeten nu nog aantonen dat elk probleem in NP polynomiaal op Uredop gereduceerd kan worden. Om dit te bewijzen reduceren we de taal HC naar Uredop. We moeten dus een polynomiaal algoritme beschrijven dat een input g omzet naar een output o zodat $g \in$ HC als en slechts als $o \in$ Uredop.
Stel dat we willen bepalen of g een element is van HC.
Test of g een graaf codeert. Indien dit zo is, noemen we deze graaf $G = (V, E)$ en gaan we verder met het algoritme. Anders outputten we bijvoorbeeld een codering voor een graaf met één top, gevolgd door een lege lijst en 1.
We voegen voor alle $v \in V$ 5 nieuwe toppen toe en verbinden al deze nieuwe toppen met elkaar. Vervolgens verwijderen we één van de toegevoegde bogen en verbinden de twee toppen die incident waren met deze boog met v. Merk op dat alle toegevoegde toppen graad 4 hebben. De graaf die we zo construeren (Figuur 3.1), noemen we $G' = (V', E')$.
Voor elke top $v \in V'$ en elke mogelijke verzameling van 4 buren van v, maken we een zestal aan. Dit zestal bestaat uit v als eerste getal, gevolgd door de 4 gekozen buren van v en 0 als laatste getal. We voegen al deze zestallen toe aan een lijst L. Elke top heeft ten hoogste $\frac{(n-1)!}{(n-5)!4!} < n^4$ verzamelingen van 4 buren, met n het aantal toppen in de graaf G'. Het opstellen van de lijst L kan dus in polynomiale tijd gebeuren. De output o is nu gelijk aan $g'\#l\#1$, met g' een codering voor G' en l een codering voor L.
We moeten nu nog aantonen dat geldt dat $g \in HC \Leftrightarrow o \in Uredop$.
Als we in G' een opspannende, 4-reguliere, samenhangende deelgraaf S vinden, dan bevat



Figuur 3.1: Voorbeeld met stappen om een Hamiltoniaanse cykel te vinden via onze vertaling.

deze alle bogen uit $E' \backslash E$, want alle toegevoegde bogen zijn incident met een toegevoegde top en alle toegevoegde toppen hebben graad 4.
Elke top $v \in V$ is incident met twee bogen uit $E' \backslash E$ en v is dus nog met twee bogen uit E incident in S.
Gezien de toegevoegde toppen en bogen in G' nooit twee toppen uit V verbinden en S een opspannende, samenhangende, 4-reguliere deelgraaf is, is S zonder de toegevoegde bogen en toppen een opspannende, samenhangende, 2-reguliere deelgraaf van G, en dus een Hamiltoniaanse cykel in G. Omgekeerd leidt een Hamiltoniaanse cykel ook tot een opspannende, samenhangende, 4-reguliere deelgraaf in G', als we de unie van alle toegevoegde bogen en toppen en de Hamiltoniaanse cykel beschouwen. □

Het bewijs is ook duidelijk veralgemeenbaar naar k-reguliere, in plaats van 4-reguliere, deelgrafen. We hebben het hier specifiek voor 4-reguliere deelgrafen bewezen omdat dit de grafen zijn waar we mee werken.

3.3.2 Methode

In deze methode (Pseudocode 3.1) gaan we voor de top s, waarvoor we zijn verzameling van buren nog niet bepaald hebben en die het minste mogelijke combinaties over heeft, de verzameling van buren bepalen. Elke nog mogelijke combinatie met s wordt afzonderlijk gekozen om verder te gaan in de branch and bound. De top met het kleinste aantal mogelijke combinaties kiezen, zorgt duidelijk voor een kleinere vertakking in het begin van onze branch and bound boom. Deze methode is een stuk eenvoudiger dan de boog-per-boog methode en heeft een vaste recursiediepte, namelijk het aantal toppen $|V_{GPS}|$. We doen minder werk wanneer we op deze manier de buren rond een top vastleggen, maar er is soms minder snel aan een snoeicriterium voldaan.

3.4 Gemeenschappelijke kenmerken

Door de manier waarop we de connectiegrafen genereren, voegen we alleen bogen toe waarvoor combinaties bestaan met toegelaten uredop waarden. De uiteindelijke grafen zullen dus een uredop waarde hebben kleiner dan de gegeven bovengrens. Aangezien deze combinaties voor elke top bestaan uit een verzameling van 4 buren waarnaar een verbinding mogelijk is, zullen grafen die we op deze manier opbouwen sowieso een 4-reguliere deelgraaf zijn van G_{Pos}.
Zoals eerder vermeld, houden we met een bovengrens van bijvoorbeeld 1.0 minder dan een kwart van de mogelijke verzamelingen van buren over. Dit zijn een kleine 1800 waarden en verdeeld over 27 toppen geeft dit lijsten met lengtes tussen 27 en 119 per top. Als we bij een top een boog vastleggen, is er onmiddellijk ook een andere lijst waaruit al veel combinaties geschrapt kunnen worden. Als alle bogen rond een top t vastliggen, mag deze top niet meer voorkomen in lijsten van andere toppen die geen buur zijn van t en kunnen we ook de lijsten

```
# Tedoen: lijst met toppen waarvoor de buren nog niet bepaald zijn
# Gedaan: lijst met toppen waarvoor de buren bepaald zijn
# Combinaties: lijst van lijsten met toppenverzamelingen van buren (voor elke top)
FUNCTIE recursie (toppenlijst Tedoen, toppenlijst Gedaan, combinatielijst Combinaties)
  # top met het kleinste aantal mogelijke combinaties kiezen
  top Doe = Topx: |Combinaties[Topx]| = min{|Combinaties[Topy]| Topy in Tedoen}
  # als er geen mogelijke verzamelingen van buren is voor Doe, moeten we backtracken
  IF |Combinaties[Doe]| = 0
    RETURN
  END IF
  # Gedaan en Tedoen aanpassen
  Tedoen = Tedoen\{Doe}
  Gedaan = Gedaan U {Doe}
  FOR toppenverzameling Buren in Combinaties[top]
    # Combinaties aanpassen voor alle toppen uit Tedoen
    Combinaties[top] = {Buren}
    FOR top Topy in Tedoen
      IF Topy in Buren
        Combinaties[Topy] = {toppenverzameling Bureny | Bureny in Combinaties[Topy]
                                                        en Doe in Bureny}
      ELSE
        Combinaties[Topy] = {toppenverzameling Bureny| Bureny in Combinaties[Topy]
                                                       en Doe niet in Bureny}
      END IF
    END FOR
    IF |Gedaan| = 0
      # een 4-reguliere graaf gevonden: uitprinten
      print(Combinaties)
    ELSE
      # verder gaan in de recursie
      recursie(Tedoen, Gedaan, Combinaties)
    END IF
  END FOR
  # Gedaan en Tedoen herstellen
  Tedoen = Tedoen U {Doe}
  Gedaan = Gedaan\{Doe}
  RETURN
```

Pseudocode 3.1: Independent set methode

van deze toppen verkleinen. Door op deze manier te genereren worden de lijsten snel zeer klein en zal er minder vertakt worden naarmate we dieper in de branch and bound boom komen.

3.5 Diameter als snoeicriterium

Naast het voor de hand liggende snoeicriterium dat in het algemene idee besproken werd, kunnen we de diameter als snoeicriterium invoeren. Daarmee zijn dan alle restricties gebruikt als snoeicriterium, of in de bouw van de connectiegrafen verwerkt. Door de manier waarop we onze grafen bouwen, weten we op elk moment welke bogen nog allemaal mogelijk zijn, we noemen de graaf met deze bogen en dezelfde toppenverzameling G'_{Pos}. Als we de diameter van G'_{Pos} berekenen, hebben we een benedengrens voor de diameter van de uiteindelijke graaf, de diameter zal namelijk nooit dalen door bogen weg te nemen. Als de diameter van G'_{Pos} groter is dan 4, of groter dan 3 als we enkel diameter 3 grafen zoeken, dan kunnen we backtracken. In elke stap de diameter van G'_{Pos} berekenen, neemt echter te veel tijd in beslag, ook al kunnen we op deze manier het vaakst backtracken. Als we echter een heuristiek gebruiken die een benedengrens voor de diameter sneller berekent, maar niet noodzakelijk de beste benedengrens geeft, dan kunnen we nog steeds vaak backtracken op basis van de diameter van G'_{Pos}. Als we bijvoorbeeld vanuit 1 top het kortste pad in G'_{Pos} naar alle andere toppen berekenen, wat in lineaire tijd kan, dan besparen we veel tijd ten opzichte van het berekenen van de diameter van G'_{Pos} en hebben we een goede benedengrens voor de diameter, waardoor we dikwijls kunnen backtracken. In ons programma gebruiken we de laatste top waarvoor alle buren bepaald zijn, als top waarvoor we alle kortste paden berekenen.

3.6 Implementatie details

We hebben nu algoritmes beschreven om connectiegrafen te genereren met uredop waarde kleiner dan een gegeven bovengrens, maar deze moeten natuurlijk nog op een efficiënte manier geïmplementeerd worden. In deze sectie worden enkele details van de implementatie van de independent set methode besproken. De boog-per-boog methode verloopt grotendeels analoog.

3.6.1 Bitvectoren

Het eerste dat we in detail bespreken is het gebruik van bitvectoren, in plaats van verzamelingen van buren als lijsten van gehele getallen bij te houden. Een bitvector wordt voorgesteld door een geheel getal, maar in plaats van de waarde van het getal te beschouwen, kijken we specifiek naar welke bits in het getal op 1 staan en welke op 0. Als in een bitvector de i-de bit op 1 staat, wil dat zeggen dat i deel is van de verzameling voorgesteld door de bitvector. Zo wordt de verzameling $\{0, 2, 3, 6\}$ bijvoorbeeld voorgesteld door het getal

$2^0 + 2^2 + 2^3 + 2^6 = 1 + 4 + 8 + 64 = 77$, als we de bit met de kleinste waarde als eerste bit beschouwen. Binair is dit dus het getal 1001101. Twee belangrijke binaire operatoren op bitvectoren zijn AND en OR, u AND v heeft 1 op alle posities waar zowel u als v een 1 hebben staan en komt dus overeen met de snede van de verzamelingen voorgesteld door u en v en u OR v heeft 1 op alle posities waar u of v een 1 heeft staan en komt dus overeen met de unie van de verzamelingen voorgesteld door u en v. Zo is 01001101 AND 00001111 gelijk aan 00001101 en 01001101 OR 00001111 gelijk aan 01001111. Elke verzameling kan dus voorgesteld worden door één geheel getal en G_{Pos} en G'_{Pos} door een lijst van 27 gehele getallen. Door deze voorstelling kunnen de snede en de unie van twee verzamelingen dus veel sneller berekend worden. Zo bepaalt bv AND 2^i, wat het binair getal is met enkel i-de positie op 1, of i in de verzameling voorgesteld door bv zit, zonder dat een iteratie over alle elementen van de verzameling nodig is, wat wel het geval zou zijn als we de verzamelingen als lijsten van gehele getallen zouden bijhouden. Omdat we in ons algoritme heel vaak de snede van twee verzamelingen moeten berekenen, zorgt deze voorstelling met bitvectoren voor een aanzienlijke versnelling van het programma.

3.6.2 Beperken iteratielengte

Wat het algoritme ook behoorlijk versnelt, is het bijhouden van een lijst L met toppen waarvoor de buren nog niet vastgelegd zijn. In elke recursiestap worden de buren voor een top t bepaald. We verwijderen t dan uit deze lijst L, en net voor we terugkeren in de recursie voegen we t terug toe aan L. Zoals je in Pseudocode 3.1 ook kan zien itereren we regelmatig over deze lijst. Als we L niet bijhouden, dan moeten we telkens over alle toppen itereren. Soms verandert dan tijdens zo'n iteratie niets voor de toppen waarvoor de buren reeds bepaald zijn, maar anders moeten we in de iteratie voor elke top s nog eens expliciet controleren of de buren van s niet in een eerdere recursiestap vastgelegd zijn. Dit vraagt natuurlijk allemaal extra werk. Daar bovenop komt nog dat er door de vertakking meer stappen dieper in de recursieboom zijn en hoe dieper we in de recursieboom zijn, hoe kleiner de lijst L is en dus hoe meer tijd we sparen. Dit maakt het bijhouden en gebruiken van zo'n lijst L extra efficiënt.

3.6.3 Heuristiek diameter

De heuristiek die we hier gaan bespreken, berekent enkel of het mogelijk is dat de afstand tussen top t en alle andere toppen uiteindelijk kleiner kan zijn dan 4 en is dus specifiek om een graaf te bekomen met diameter 3. We zijn vooral geïnteresseerd in grafen met diameter 3 en dit laat toe om het gebruik van bitvectoren nog eens te illustreren.
Stel dat d de bitvector is die de buren van t voorstelt. Gezien we deze heuristiek toepassen op de laatste top waarvoor de buren vastliggen, zal d steeds een verzameling van 4 buren voorstellen, maar dat is voor het algoritme niet noodzakelijk.
Ga de (mogelijke) buren b van t af en bereken telkens $d = d$ OR bb, waar bb de bitvector is die

de (mogelijke) buren van b voorstelt. Nu heeft d een 1 op alle plaatsen die op afstand kleiner of gelijk aan 2 van t liggen. Nu kijken we voor elke top s of die een buur heeft die element is van de verzameling voorgesteld door d; als bs de bitvector is die de verzameling van buren van s voorstelt, dan geldt dat bs AND $d = 0 \Leftrightarrow s$ heeft geen buren in d. Als bs AND $d = 0$ zullen s en t dus minstens op afstand 4 liggen in de uiteindelijke graaf die we aan het bouwen zijn, en kunnen we dus backtracken.

3.6.4 Aanpassing lijsten met mogelijke burenverzamelingen

Zoals je kan zien in Pseudocode 3.1, moeten we in elke recursiestap, voor elke verzameling van buren B die we kiezen, voor elke top b, de lijst van mogelijke verzamelingen van buren aanpassen (voor toppen waarvoor de buren al vastliggen is dit sowieso in orde, dus die negeren we). Als we in deze recursiestap de buren van top t vastleggen, dan moeten we de lijst L_b met mogelijke burenverzamelingen van b vervangen door L'_b. L'_b bestaat uit alle burenverzamelingen van L_b die t bevatten als b een element is van B en L'_b bestaat uit alle burenverzamelingen van L_b die t niet bevatten als b geen element is van B.
Er zijn dus twee mogelijkheden voor L'_b en de unie van deze twee mogelijkheden levert terug L_b op. Als we terugkeren in de recursie moet de lijst met mogelijke burenverzamelingen voor b opnieuw L_b zijn. Hiermee rekening houdend, hebben we een methode (Pseudocode 3.2) ontworpen die deze lijsten efficiënt splitst en zodat de lijsten ook héél snel terug hersteld kunnen worden.
In deze methode gaan we de lijst met burenverzamelingen tegelijk af in beide richtingen, dus van begin naar einde en van einde naar begin. Als we bij het overlopen van de lijst van begin naar einde een burenverzameling tegenkomen die t niet bevat en bij het overlopen van de lijst van einde naar begin een burenverzameling tegenkomen die t wel bevat, dan wisselen we beide verzamelingen van plaats in de lijst en gaan we in beide richtingen verder met zoeken. We doen dit tot we uit beide richtingen op dezelfde positie l komen en we de hele lijst bekeken hebben. De lengte van de eerste lijst is nu l, de pointer naar de tweede lijst is de originele pointer plus l en de lengte van de tweede lijst is gelijk aan de lengte van de originele lijst min l. De eerste lijst bevat nu alle elementen van L_b die t bevatten, en de tweede lijst bevat alle elementen van L_b die t niet bevatten. Om beide lijsten samen te voegen moeten we gewoon de pointer van de eerste lijst behouden en lengtes van beide lijsten optellen zodat we terug de originele lengte en originele lijst (maar misschien in een andere volgorde) uitkomen.
De voor de hand liggende methode is om telkens een nieuwe lijst L'_b aanmaken en de oude lijst L_b bij te houden voor wanneer er teruggekeerd wordt in de recursie. De methode die we hier besproken hebben heeft naast minder geheugengebruik, wat in deze toepassing niet zo belangrijk is, een tweede voordeel. We hoeven deze lijsten nu niet meer voor elke burenverzameling B die we voor t kiezen opnieuw te berekenen. We hebben namelijk beide mogelijkheden berekend, en moeten dus enkel de juiste pointer en de juiste lengte meegeven.

```
# Lb: lijst met mogelijke verzamelingen van buren voor een top b
# T: de top waarvoor we de buren aan het vastleggen zijn
# Lengte: aantal burenverzamelingen in Lb (zal worden aangepast)
# LengteEen: de lengte van de eerste lijst (return variabele)
# LengteTwee: de lengte van de tweede lijst (return variabele)
FUNCTIE aanpassen (burenverzamelingenlijst Lb, top T,
                   int Lengte, int LengteEen, int LengteTwee)

  int i = 0
  int j = size-1

  WHILE TRUE
    WHILE T element van Lb[i] && i<size-1
      i++
    END WHILE
    WHILE T geen element van Lb[j] && j>0
      j--
    END WHILE

    IF i < j && i<size-1 && j>0
      wissel Lb[i] en Lb[j]
    ELSE
      break;
    END IF
  END WHILE

  LengteEen = i;
  LengteTwee = Lengte-i;
  RETURN
  # Door LengteEen, LengteTwee en de nieuwe inhoud van Lb
  # liggen de twee nieuwe lijsten volledig vast
```

Pseudocode 3.2: Methode om lijsten met burenverzamelingen te splitsen.

Hoe groter de lijst van burenverzamelingen voor t, hoe meer voordeel deze methode heeft ten opzichte van de voor de hand liggende methode.

Hoofdstuk 4

Resultaten

4.1 Vergelijking methodes

We hebben twee programma's geschreven die de connectiegrafen opbouwen, één programma dat boog per boog toevoegt aan de graaf en één programma dat als oplosser van ons independent set probleem kan gezien worden. Verder kunnen we ons independent set probleem ook met een algemene independent set oplosser (Niskanen & Östergård (2003) en Östergård (2002)) oplossen.

4.1.1 Uitkomst

Met al deze programma's kunnen we alle 4-reguliere deelgrafen van G_{Pos} opbouwen, met uredop waarde kleiner dan een gegeven bovengrens. Deze moeten natuurlijk allemaal dezelfde grafen opleveren, om toch enige zekerheid te hebben dat de programma's die we geschreven hebben, correct werken. Omdat het over enorm veel grafen gaat, hebben we enkel sommige speciale grafen, zoals bijvoorbeeld deze met kleinste uredop waarde, met elkaar vergeleken. Verder hebben we wel van alle connectiegrafen die we vonden, de diameter berekend en de aantallen van de grafen voor elke diameter vergeleken. Gegeven een bovengrens voor de uredop waarde moet het aantal grafen dat per diameter gevonden wordt voor elk programma hetzelfde zijn, en daaruit volgt onmiddellijk dat het totaal aantal grafen dat gevonden wordt ook gelijk zal zijn. Door enkele specifieke gevallen te vergelijken en voor zo'n groot aantal grafen dezelfde diameters uit te komen, kunnen we vrij zeker zijn dat de programma's correct zijn, en dat we dus geen grafen overslaan of teveel tellen. Op deze manier kunnen we dus wel degelijk de beste grafen vinden. In Tabel 4.1 zie je het aantal grafen van elke diameter, zoals die in de programma's gevonden werd, met uredop waarde kleiner dan een gegeven uredop waarde. Deze resultaten werden ook bevestigd door een implementatie van de independent set methode door Gunnar Brinkmann.

Tabel 4.1: Aantal grafen per diameter met uredop waarde kleiner dan een gegeven bovengrens.

Maximum uredop waarde	diam 3	diam 4	diam 5	diam 6	diam 7	diam 8	onsamenhangend
0.90	0	0	0	0	0	0	0
0.92424	0	2	0	0	0	0	0
0.93	0	10	11	0	0	0	0
0.94	0	538	310	6	0	0	0
0.95	0	6428	2861	51	0	0	0
0.96	0	64387	30719	425	7	0	0
0.97	1	1025835	354578	3760	42	0	0
0.98	6	5981974	1760076	13129	111	1	0
0.99	20	29671937	7583544	45631	321	4	3
1.00	104	89844371	20179130	101680	736	10	33

4.1.2 Tijdsmetingen

Voor de tijdsmetingen (Tabel 4.2) zijn de programma's allemaal uitgevoerd op een Dell Inspiron 6400 met Intel Core Duo T2250 processor met klokfrequentie 1.73Ghz.
De algemene kliek oplosser is natuurlijk veel trager dan de andere twee programma's, deze zijn namelijk specifiek voor dit probleem geschreven. Meer resultaten halen voor de algemene kliek oplosser was dan ook niet meer relevant en zou duidelijk veel tijd kosten. Als we de boog-per-boog methode met de independent set methode vergelijken, weten we dat de boog-per-boog methode iets meer rekenwerk vraagt in elke stap, maar er wordt soms iets vroeger gesnoeid. Uit de resultaten blijkt dat het extra rekenwerk meer tijd inneemt dan de tijd die we winnen omdat we vroeger kunnen snoeien. We kunnen met de boog-per-boog methode geen extra voordeel halen ten opzichte van de independent set methode als we de bound op diameter in rekening brengen, omdat we onze benedengrens voor de diameter enkel kunnen beperken telkens alle buren van een top bepaald zijn. Om deze reden, en omdat de independent set methode al sneller was, hebben we er ook voor gekozen deze extra bound enkel in de independent set methode in te voeren. Zoals je kan zien in Tabel 4.2, zorgt dit voor een duidelijke versnelling van het programma, maar op deze manier genereren we natuurlijk enkel met een bepaalde diameter alle connectiegrafen. Als we de methode beschreven in Sectie 3.6.3 toepassen, vinden we bijvoorbeeld enkel met diameter 3 alle connectiegrafen.

Tabel 4.2: Benaderde tijd die een programma nodig heeft om de connectiegrafen met een uredop waarde kleiner dan een gegeven bovengrens te berekenen.

Maximum uredop waarde	algemene kliek oplosser	boog-per-boog methode	independent set methode	independent set methode met diameter bound
0.90	31 min 15 sec	0.0 sec	0.0 sec	0.0 sec
0.92424	19 min 49 sec	0.4 sec	0.3 sec	0.2 sec
0.93	58 min 9 sec	1.2 sec	1.0 sec	0.7 sec
0.94	300 min 19 sec	7.7 sec	7.5 sec	4.3 sec
0.95		48 sec	46 sec	25 sec
0.96		4 min 6 sec	3 min 45 sec	1 min 47 sec
0.97		17 min 49 sec	15 min 44 sec	5 min 43 sec
0.98		75 min 43 sec	68 min 35 sec	22 min 17 sec
0.99			247 min 32 sec	64 min 2 sec
1.00			643 min 6 sec	140 min 7 sec

4.1.3 VDOP en maximale uredop

We hebben de programma's ook opgestart met de VDOP waarden en met de maximale uredop waarden als input. Voor de maximale uredop waarden kregen we gelijkaardige resultaten als bij de gewone (95%) uredop waarden, met dat verschil dat de maximale uredop waarden iets hoger liggen. Dit komt omdat er bij kleine uredop waarden meestal weinig verschil is tussen 95% en de maximale uredop waarden. Zo hebben natuurlijk ook alle grafen met maximale uredop waarde kleiner dan u, een 95% uredop waarden kleiner dan u.
De VDOP waarden leveren duidelijk héél andere resultaten op. Ter vergelijking: er zijn 1421 combinaties met VDOP waarde kleiner dan 0.848 (zie Figuur 4.1) en 1473 combinaties met uredop waarde kleiner dan 0.97 (zie Figuur 1.1), het aantal grafen dat gevonden wordt is echter respectievelijk 68.861.162 en 1.384.216 (zie Tabellen 4.5 en 4.1). Als we alle grafen met VDOP waarde kleiner dan 0.85 genereren, maken we gebruik van 1546 combinaties en is het programma meer dan 2 dagen bezig met genereren. Er worden dan ook al 2.378.958.923 grafen gegenereerd. Met een uredop waarde kleiner dan 0.98 maken we gebruik van 1593 combinaties en vinden we 7.755.297 grafen in iets meer dan een uur. Uit de resultaten (Tabellen 4.2 en 4.6) blijkt ook dat de diameter bound een veel enorme impact heeft als we de VDOP waarden als input beschouwen. We kunnen besluiten dat de uitvoeringstijden en resultaten sterk afhankelijk zijn van de waarden die we gebruiken.

Tabel 4.3: Aantal grafen per diameter met maximale uredop waarde kleiner dan een gegeven bovengrens.

Maximum uredop waarde	diam 3	diam 4	diam 5	diam 6	diam 7	diam 8	onsamenhangend
0.92	0	0	0	0	0	0	0
0.93	0	0	1	0	0	0	0
0.94	0	57	48	0	0	0	0
0.95	0	1336	823	25	0	0	0
0.96	0	14023	7142	121	2	0	0
0.97	0	97353	46749	780	14	0	0
0.98	0	1346008	456790	4570	54	0	0

Tabel 4.4: Benaderde tijd die een programma nodig heeft om de connectiegrafen met een maximale uredop waarde kleiner dan een gegeven bovengrens te berekenen.

Maximum maximale uredop waarde	boog-per-boog methode	independent set methode	independent set methode met diameter bound
0.92	0.0 sec	0.0 sec	0.0 sec
0.93	0.4 sec	0.4 sec	0.3 sec
0.94	3.2 sec	3.0 sec	2.1 sec
0.95	17.2 sec	16.1 sec	8.9 sec
0.96	1 min 42 sec	1 min 29 sec	45 sec
0.97	5 min 22 sec	5 min 30 sec	2 min 5 sec
0.98	30 min 38 sec	23 min 38 sec	10 min 5 sec

4.2 Beste grafen

Als we het programma voor te kleine uredop waarden als bovengrens laten lopen (waarvoor dus geen grafen bestaan) dan is het aantal mogelijke combinaties per top zeer beperkt. Het programma vindt in die gevallen dan ook heel snel dat er geen mogelijkheden zijn. De grafen met kleinste uredop waarden werden om deze reden dan ook heel snel gevonden. Deze grafen hebben echter diameter 4. Aangezien veel van de uredop waarden dicht bij elkaar liggen, zorgen kleine stijgingen in de bovengrens voor een grote extra tijd die nodig is om alle grafen te berekenen. Door op deze manier de bovengrens langzaam te laten stijgen kwamen we tot de beste grafen. Grotere waarden als bovengrens u leveren een groter aantal goede grafen op. De verzameling van grafen gegenereerd onder een kleinere bovengrens dan u, is namelijk steeds een deelverzameling van deze gegenereerd met u als bovengrens.

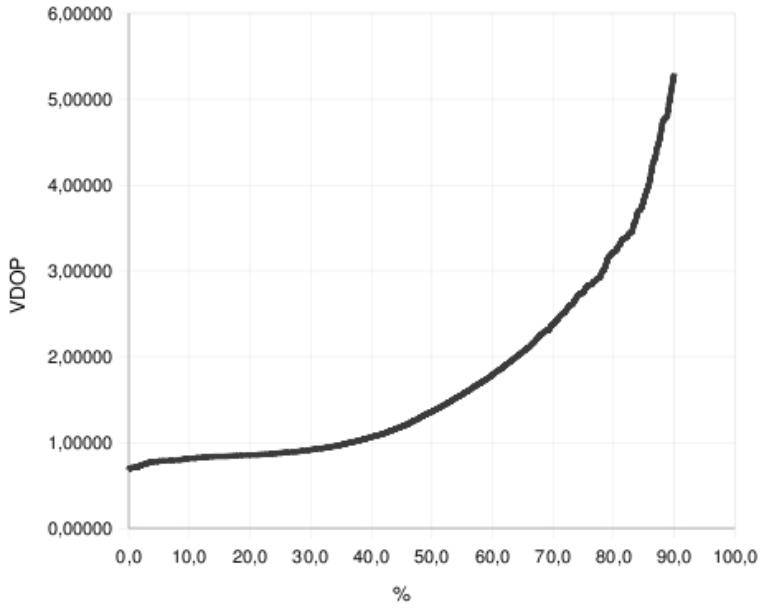
Tabel 4.5: Aantal grafen per diameter met VDOP waarde kleiner dan een gegeven bovengrens.

Maximum VDOP waarde	diam 3	diam 4	diam 5	diam 6	diam 7	diam 8	onsamenhangend
0.82	0	0	0	0	0	0	0
0.822705	0	31	13	0	0	0	0
0.83	0	1736	437	0	0	0	0
0.84	0	7151585	5576121	66038	476	9	0
0.848	0	44200350	24492936	167012	855	9	0
0.85	7	1497881476	873926484	7103318	46716	596	326

Tabel 4.6: Benaderde tijd die een programma nodig heeft om de connectiegrafen met een VDOP waarde kleiner dan een gegeven bovengrens te berekenen.

Maximum VDOP waarde	boog-per-boog methode	independent set methode	independent set methode met diameter bound
0.82	0.0 sec	0.0 sec	0.0 sec
0.822705	0.0 sec	0.0 sec	0.0 sec
0.83	0.4 sec	0.4 sec	0.2 sec
0.84	18 min 0 sec	18 min 10 sec	15 sec
0.848	112 min 30 sec	118 min 44 sec	1 min 7 sec
0.85	54 uur 35 min		10 min 32 sec

De twee connectiegrafen met kleinste uredop waarden hebben een uredop waarde van 0.92423 en diameter 4, en ten hoogste diameter 5 na het verwijderen van een willekeurige boog (Tabellen A.1 en A.2). De 2 beste connectiegrafen die na het verwijderen van een willekeurige boog ten hoogste diameter 4 hebben, zijn terug te vinden in Tabellen A.3 en A.4. Deze hebben beiden diameter 4 en een uredop waarde van 0.93317. Met diameter 3 is er één beste connectiegraaf met kleinste uredop waarde 0.96598 (Tabel A.5). Deze heeft na het verwijderen van een willekeurige boog ten hoogste diameter 5. Ten slotte is er nog de beste connectiegraaf met diameter 3, die na het verwijderen van een willekeurige boog ten hoogste diameter 4 heeft. Deze heeft een uredop waarde van 0.97854 en is terug te vinden in Tabel A.6 en Figuur A.6. Al deze beste grafen zijn terug te vinden in appendix A.



Figuur 4.1: Eerste 90 % van de connectiecombinaties met hun VDOP waarde

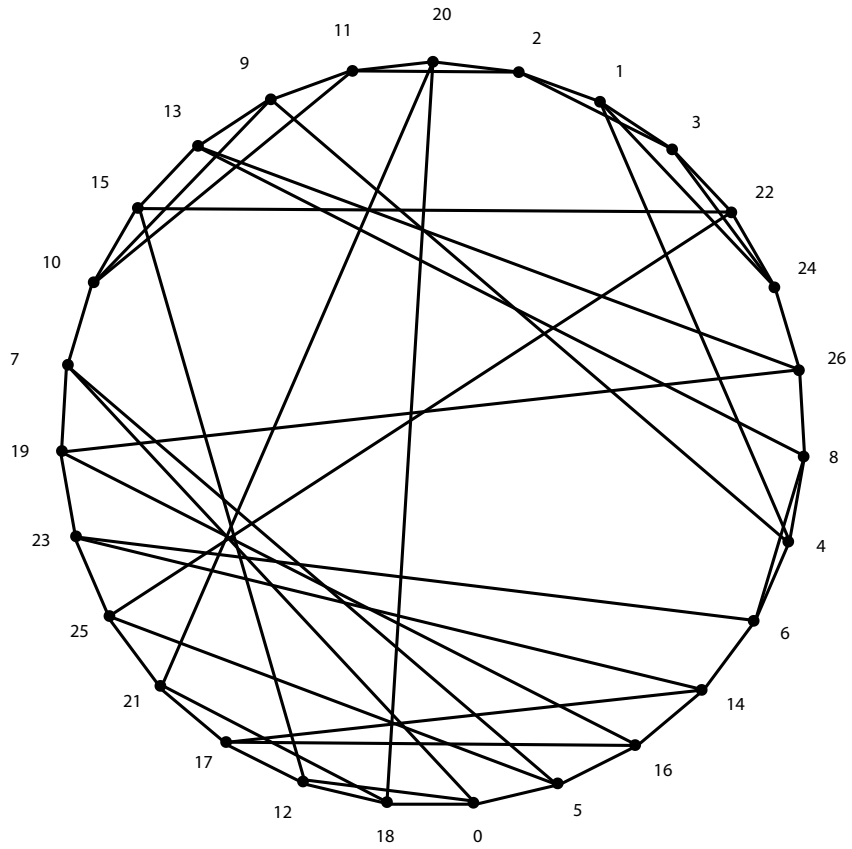
Hoofdstuk 5

Conclusie

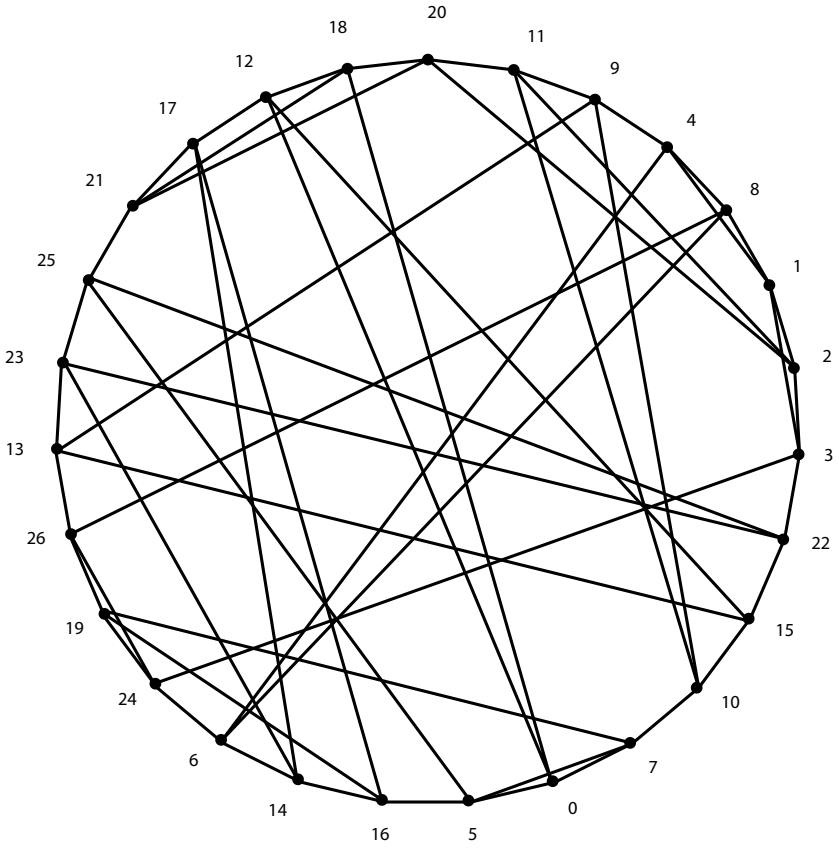
Lockheed Martin zal communicatie moeten opzetten tussen de satellieten van het komende GPS III systeem. Hiervoor willen ze een zo goed mogelijk communicatienetwerk tussen deze satellieten kennen. In deze thesis hadden we als doel de connectiegraaf te vinden die tot het beste communicatienetwerk leidt. Om deze beste connectiegraaf te vinden hebben we de data gekregen voor een constellatie van 27 satellieten, waar elke satelliet met 4 andere verbonden is. We hebben uiteindelijk een programma geschreven, gebaseerd op wat we de independent set methode genoemd hebben, dat de beste connectiegrafen vindt. Men kan denken dat we onmiddellijk de independent set methode hadden kunnen implementeren. In plaats hiervan zijn we begonnen met een deelgraaf algoritme te schrijven om zo de restricties in meer detail te bestuderen. Deze eerste stappen waren nodig om een goed beeld te krijgen van het probleem dat ons voorgeschoteld werd. Rekening houdend met de eisen die ons opgelegd waren en met de context van het probleem, hebben we de eigenschappen bepaald die tot beste grafen kunnen leiden. Het programma dat we geschreven hebben, vindt deze beste grafen snel wanneer we het toepassen op de input die we gekregen hebben. Het programma kan natuurlijk ook gebruikt worden om de beste connectiegrafen te vinden voor een ander aantal satellieten, andere waarden om de kwaliteit van de combinaties van buren voor te stellen en een verschillend aantal connecties per satelliet. We hebben echter aangetoond dat we met een NP-compleet probleem werken, en dus kan een kleine stijging in het aantal satellieten voor een heel grote stijging in uitvoertijd zorgen. Het is nu aan Lockheed Martin om te bepalen hoe de connectiegrafen er precies moeten uitzien en wat ze als de beste connectiegraaf gaan beschouwen. Met de hier beschreven methode kan Lockheed Martin de beste connectiegraaf van het communicatienetwerk voor het GPS III systeem vinden.

Bijlage A

Beste grafen



Figuur A.1: Beste graaf 1 met diameter 4 en diameter 5 na het verwijderen van een willekeurige boog.



Figuur A.2: Beste graaf 2 met diameter 4 en diameter 5 na het verwijderen van een willekeurige boog.

Tabel A.1: Eerste graaf met diameter 4, uredop waarde 0.92423 en na het verwijderen van een willekeurige boog diameter 5.

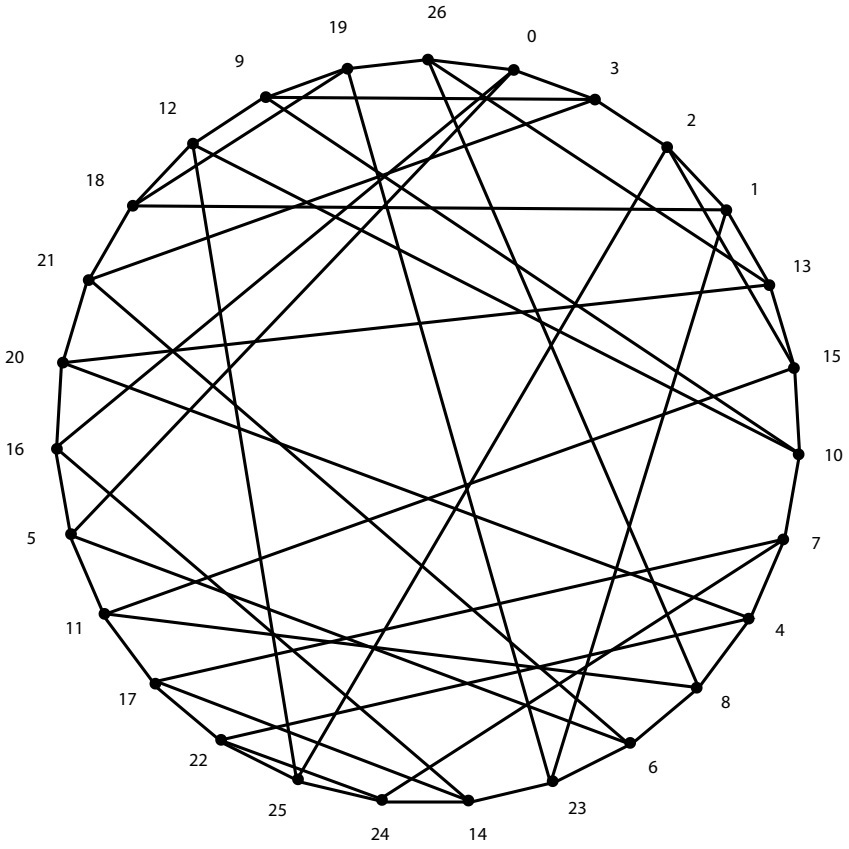
Top	Buur 1	Buur 2	Buur 3	Buur 4
0	5	7	12	18
1	2	3	4	24
2	1	3	11	20
3	1	2	22	24
4	1	6	8	9
5	0	7	16	25
6	4	8	14	23
7	0	5	10	19
8	4	6	13	26
9	4	10	11	13
10	7	9	11	15
11	2	9	10	20
12	0	15	17	18
13	8	9	15	26
14	6	16	17	23
15	10	12	13	22
16	5	14	17	19
17	12	14	16	21
18	0	12	20	21
19	7	16	23	26
20	2	11	18	21
21	17	18	20	25
22	3	15	24	25
23	6	14	19	25
24	1	3	22	26
25	5	21	22	23
26	8	13	19	24

Tabel A.2: Tweede graaf met diameter 4, uredop waarde 0.92423 en na het verwijderen van een willekeurige boog diameter 5.

Top	Buur 1	Buur 2	Buur 3	Buur 4
0	5	7	12	18
1	2	3	4	8
2	1	3	11	20
3	1	2	22	24
4	1	6	8	9
5	0	7	16	25
6	4	8	14	24
7	0	5	10	19
8	1	4	6	26
9	4	10	11	13
10	7	9	11	15
11	2	9	10	20
12	0	15	17	18
13	9	15	23	26
14	6	16	17	23
15	10	12	13	22
16	5	14	17	19
17	12	14	16	21
18	0	12	20	21
19	7	16	24	26
20	2	11	18	21
21	17	18	20	25
22	3	15	23	25
23	13	14	22	25
24	3	6	19	26
25	5	21	22	23
26	8	13	19	24

Tabel A.3: Eerste graaf met diameter 4, uredop waarde 0.93317 en na het verwijderen van een willekeurige boog nog steeds diameter 4.

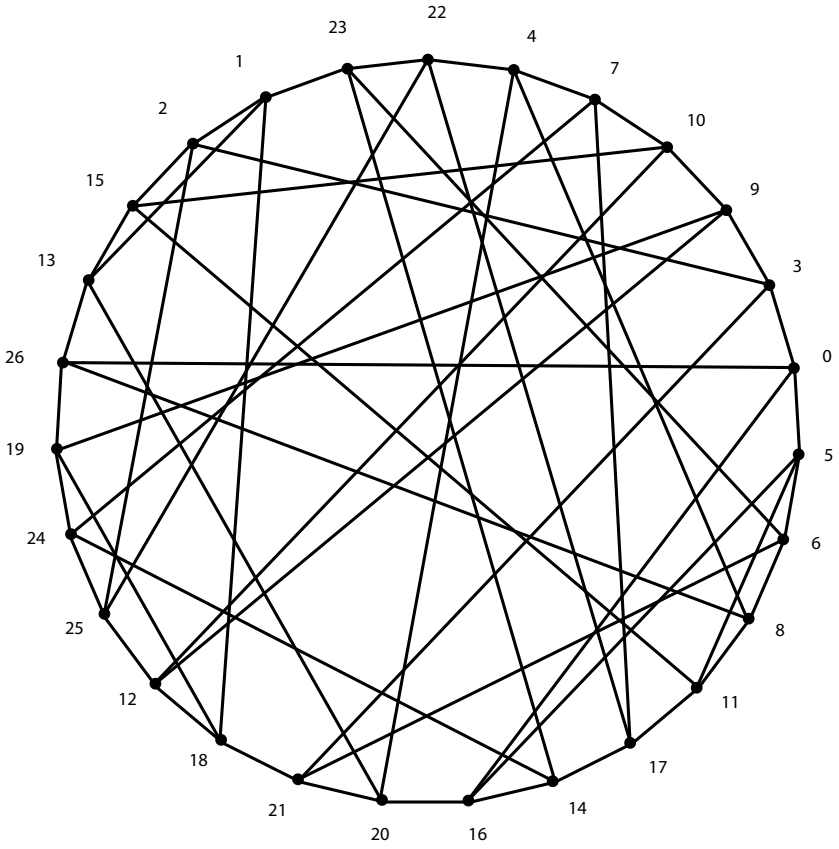
Top	Buur 1	Buur 2	Buur 3	Buur 4
0	3	5	16	26
1	2	13	18	23
2	1	3	15	25
3	0	2	9	21
4	7	8	20	22
5	0	6	11	16
6	5	8	21	23
7	4	10	17	24
8	4	6	11	26
9	3	10	12	19
10	7	9	12	15
11	5	8	15	17
12	9	10	18	25
13	1	15	20	26
14	16	17	23	24
15	2	10	11	13
16	0	5	14	20
17	7	11	14	22
18	1	12	19	21
19	9	18	23	26
20	4	13	16	21
21	3	6	18	20
22	4	17	24	25
23	1	6	14	19
24	7	14	22	25
25	2	12	22	24
26	0	8	13	19



Figuur A.3: Beste graaf 1 met diameter 4 en nog steeds diameter 4 na het verwijderen van een willekeurige boog.

Tabel A.4: Tweede graaf met diameter 4, uredop waarde 0.93317 en na het verwijderen van een willekeurige boog nog steeds diameter 4.

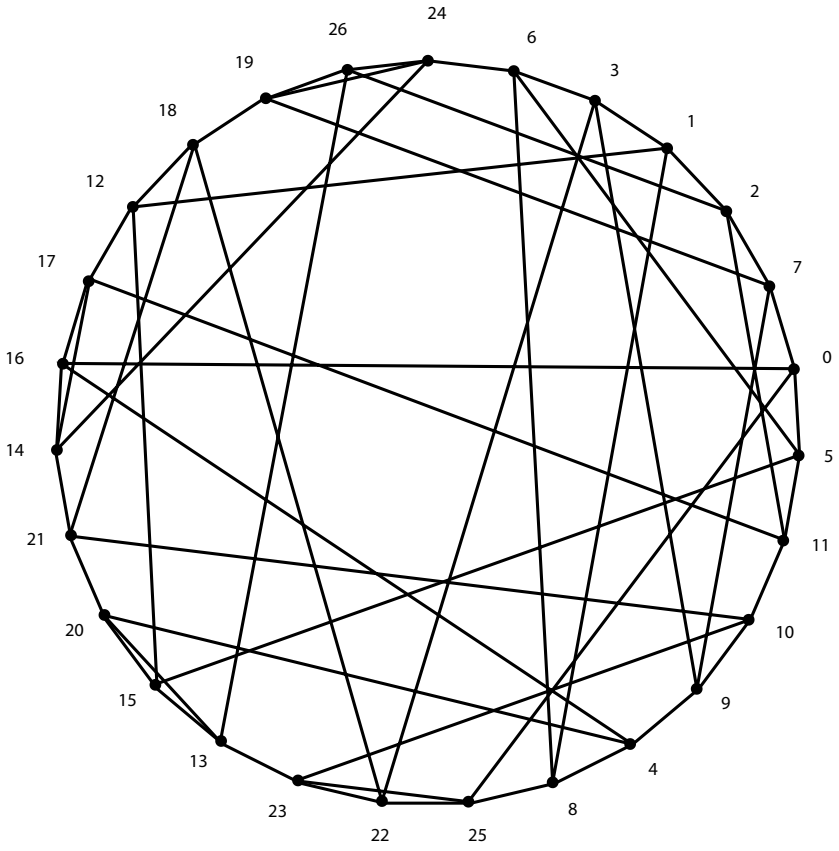
Top	Buur 1	Buur 2	Buur 3	Buur 4
0	3	5	16	26
1	2	13	18	23
2	1	3	15	25
3	0	2	9	21
4	7	8	20	22
5	0	6	11	16
6	5	8	21	23
7	4	10	17	24
8	4	6	11	26
9	3	10	12	19
10	7	9	12	15
11	5	8	15	17
12	9	10	18	25
13	1	15	20	26
14	16	17	23	24
15	2	10	11	13
16	0	5	14	20
17	7	11	14	22
18	1	12	19	21
19	9	18	24	26
20	4	13	16	21
21	3	6	18	20
22	4	17	23	25
23	1	6	14	22
24	7	14	19	25
25	2	12	22	24
26	0	8	13	19



Figuur A.4: Beste graaf 2 met diameter 4 en nog steeds diameter 4 na het verwijderen van een willekeurige boog.

Tabel A.5: De graaf met diameter 3, uredop waarde 0.96598 en na het verwijderen van een willekeurige boog diameter 5.

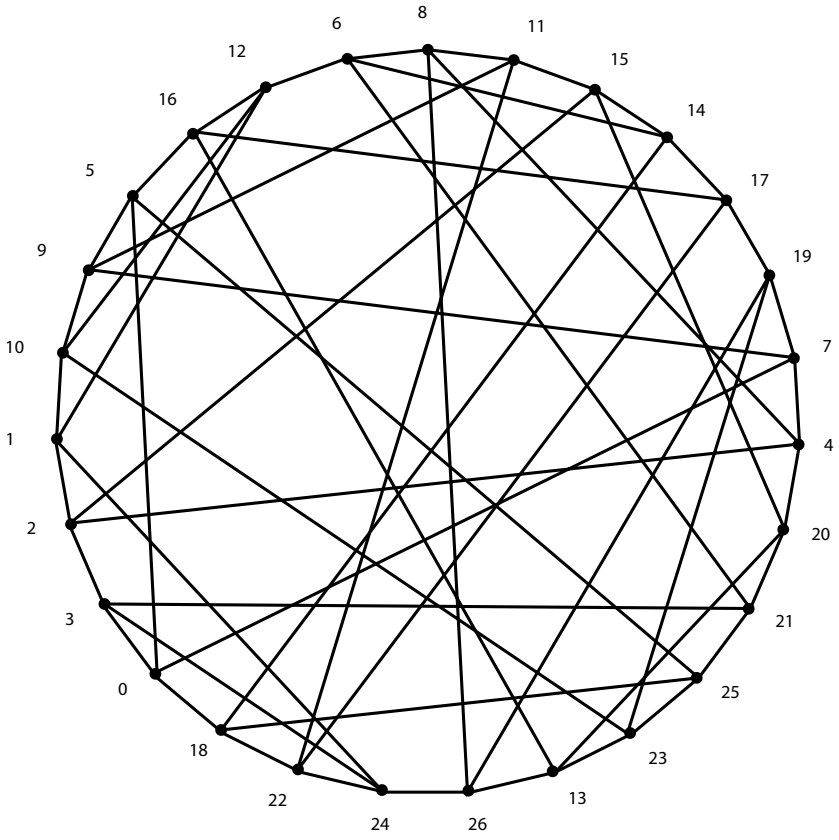
Top	Buur 1	Buur 2	Buur 3	Buur 4
0	5	7	16	25
1	2	3	8	12
2	1	7	11	26
3	1	6	9	22
4	8	9	16	20
5	0	6	11	15
6	3	5	8	24
7	0	2	9	19
8	1	4	6	25
9	3	4	7	10
10	9	11	21	23
11	2	5	10	17
12	1	15	17	18
13	15	20	23	26
14	16	17	21	24
15	5	12	13	20
16	0	4	14	17
17	11	12	14	16
18	12	19	21	22
19	7	18	24	26
20	4	13	15	21
21	10	14	18	20
22	3	18	23	25
23	10	13	22	25
24	6	14	19	26
25	0	8	22	23
26	2	13	19	24



Figuur A.5: Beste graaf met diameter 3 en diameter 5 na het verwijderen van een willekeurige boog.

Tabel A.6: De graaf met diameter 3, uredop waarde 0.97854 en na het verwijderen van een willekeurige boog diameter 4.

Top	Buur 1	Buur 2	Buur 3	Buur 4
0	3	5	7	18
1	2	10	12	24
2	1	3	4	15
3	0	2	21	24
4	2	7	8	20
5	0	9	16	25
6	8	12	14	21
7	0	4	9	19
8	4	6	11	26
9	5	7	10	11
10	1	9	12	23
11	8	9	15	22
12	1	6	10	16
13	16	20	23	26
14	6	15	17	18
15	2	11	14	20
16	5	12	13	17
17	14	16	19	22
18	0	14	22	25
19	7	17	23	26
20	4	13	15	21
21	3	6	20	25
22	11	17	18	24
23	10	13	19	25
24	1	3	22	26
25	5	18	21	23
26	8	13	19	24



Figuur A.6: Beste graaf met diameter 3 en diameter 4 na het verwijderen van een willekeurige boog.

Bibliografie

G. Brinkmann (2010). Algoritmische grafentheorie.

R. Diestel (2005). *Graph Theory*. Springer-Verlag, third edition.

M. Garey, D. Johnson & R. Tarjan (1976). The planar hamiltonian circuit problem is np-complete. *SIAM Journal on Computing*, 5(4):704–714.

M. R. Garey & D. S. Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co.

R. B. Langley (1999). Dilution of precision. *GPS World*, 10(5):52–59.

M. Meringer (1999). Fast generation of regular graphs and construction of cages. *Journal of Graph Theory*, 30:137–146.

S. Niskanen & P. R. J. Östergård (2003). *Cliquer User's Guide, Version 1.0*. Communications Laboratory, Helsinki University of Technology.

P. R. J. Östergård (2002). A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, 120(1-3):197–207.