

lofAR velden BEKABELING

Verslag van een stageproject in het kader van de studie Informatica aan de NHL Hogeschool Leeuwarden. Het stageproject is uitgevoerd bij ASTRON.

oKTOBER 2018

tHOMAS TIJSMA



LOFAR velden bekabeling;

Verslag van een stageproject in het kader van de studie Informatica aan de NHL Hogeschool Leeuwarden. Het Stageproject is uitgevoerd bij ASTRON ter Dwingeloo.

Stageproject van periode September 2018 – Januari 2019

Uitgevoerd voor ASTRON afdeling R&D

Onder toezicht van Dr. T.J. Dijkema

Schoolbegeleider ing. W. D. Brinksma

Inhoud

[1. Inleiding 4](#_Toc534816477)

[1.1 Wat is ASTRON? 4](#_Toc534816478)

[1.2 Context en aanleiding 4](#_Toc534816479)

[1.3 Probleemstelling 4](#_Toc534816480)

[1.4 Afbakening van de opdracht 6](#_Toc534816481)

[2. Probleemanalyse 6](#_Toc534816482)

[2.1 Eisen en wensen 6](#_Toc534816483)

[2.1.1 Eisen 6](#_Toc534816484)

[2.1.2 Wensen 6](#_Toc534816485)

[2.2 Randvoorwaarden 6](#_Toc534816486)

[2.3 Risicoanalyse 7](#_Toc534816487)

[3. Onderzoek 7](#_Toc534816488)

[3.1 inleiding 7](#_Toc534816489)

[3.2 Hoofd- en deelvragen 8](#_Toc534816490)

[3.3 Methoden 8](#_Toc534816491)

[3.4 Theoretisch kader 8](#_Toc534816492)

[3.4.1 Bekende technieken 9](#_Toc534816493)

[3.4.1.1 Rondreizende handelaar probleem 9](#_Toc534816494)

[3.4.1.2 Hamiltonian pad 9](#_Toc534816495)

[3.4.1.3 Chinese postbode probleem 10](#_Toc534816496)

[3.4.1.4 Drie huizen probleem 10](#_Toc534816497)

[3.4.1.5 Zeven bruggen van Königsberg 11](#_Toc534816498)

[3.4.1.6 Kortste pad probleem 11](#_Toc534816499)

[3.4.1.7 Minimaal opspannende boom 12](#_Toc534816500)

[3.4.1.8 Steiner boom 12](#_Toc534816501)

[3.5 Resultaten 13](#_Toc534816502)

[3.5.1 Welke bekende technieken zijn er? 13](#_Toc534816503)

[3.5.2 Zijn er specifieke technieken voor dit probleem? 17](#_Toc534816504)

[3.6 Analyse van onderzoeksresultaten 18](#_Toc534816505)

[3.7 Conclusie en besluit 18](#_Toc534816506)

[4. Ontwerp 20](#_Toc534816507)

[4.1 UML diagrammen 20](#_Toc534816508)

[4.1.1 Architectuurniveau als een component diagram: 20](#_Toc534816509)

[4.1.2 Detailniveau als flowchart: 21](#_Toc534816510)

[4.2 Motivatie 22](#_Toc534816511)

[4.3 Tools 22](#_Toc534816512)

[4.4 Testen 22](#_Toc534816513)

[4.4.1 Validatie eisen 22](#_Toc534816514)

[4.4.2 Validatie wensen 23](#_Toc534816515)

[5. Realisatie 24](#_Toc534816516)

[5.1 Eindproduct 24](#_Toc534816517)

[5.2 Validatie eisen en wensen 25](#_Toc534816518)

[5.2.1 Validatie eisen 25](#_Toc534816519)

[5.2.2 Validatie wensen 25](#_Toc534816520)

[6. Conclusie en aanbeveling 25](#_Toc534816521)

[7. Reflectie 26](#_Toc534816522)

[7.1 Onderzoek 26](#_Toc534816523)

[7.2 Product 26](#_Toc534816524)

[7.3 Eigen handelen 26](#_Toc534816525)

[7.4 Leerdoelen 26](#_Toc534816526)

[8. Bronnen 27](#_Toc534816527)

[9. Leerdoelen 28](#_Toc534816528)

[10. Planning 29](#_Toc534816529)

[11. Versiebeheer 30](#_Toc534816530)

[12. Logboek 30](#_Toc534816531)

# Inleiding

## 1.1 Wat is ASTRON?

ASTRON is het Nederlandse instituut van Radio Astronomie. Ze observeren en bestuderen de radiogolven die het universum uitstraalt. Met als doel het doen van ontdekkingen in de radio astronomie. Dit betekent niet alleen in dat ze fundamenteel astronomisch onderzoek doen, maar ze ontwerpen en onderhouden ook radiotelescopen die de koplopers van de wereld zijn op hun eigen gebied, de limieten van technologie verbreden om nog betere en nog preciezere instrumenten te ontwerpen.

ASTRON’s astronomen en ingenieurs zijn bekend in de astronomie wereld, De astronomen voeren grensverleggend onderzoek uit naar niet alleen onze Melkweg, maar ook op sterrenstelsels die zo ver weg liggen dat je zou kunnen zeggen dat ze op de rand van het universum liggen. De ingenieurs ontwikkelen innovatieve antennes en hightech elektronica. Met dank aan constante ontwikkeling is de WSRT(Westerbork Synthesis Radio Telescope) nog steeds een van de beste telescopen in de wereld.

LOFAR(Low frequency array) is een uniek instrument ontwikkeld door ASTRON. Het meet de vroegste fases van het universum, radioflitsen aan de hemel, draaiende neutron sterren, en botsende zwarte gaten door lage frequentie radiostralingen te meten.

Verder is ASTRON betrokken met grote namen in de radioastronomie zoals : SKA, NOVA, JIVE, ERIC en NWO.

## 1.2 Context en aanleiding

Elk LOFAR veld heeft zijn eigen unieke kabelplan nodig. Aangezien dit een week tijd kost om met de hand te doen wordt er een systeem gevraagd die dit automatisch en sneller kan.

## 1.3 Probleemstelling

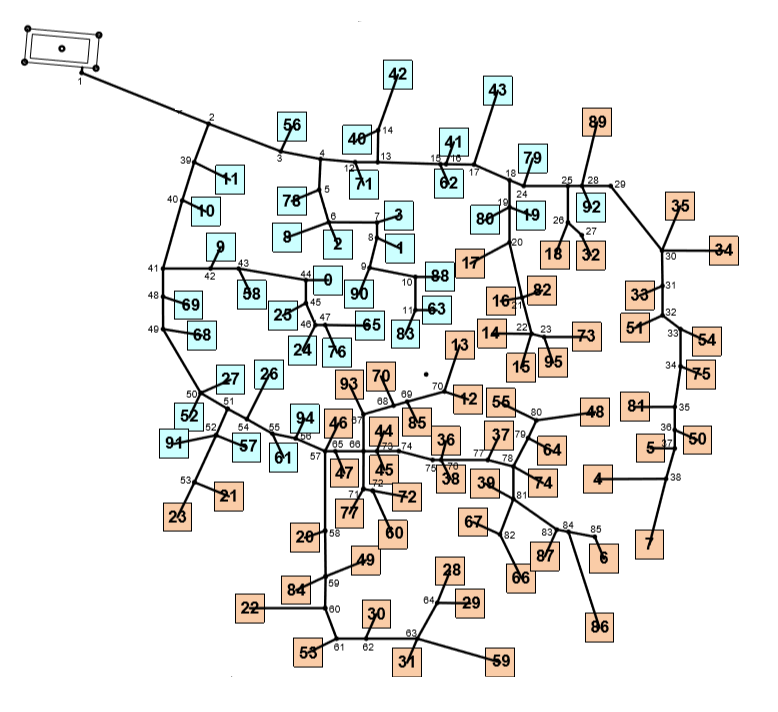
ASTRON beheert een heleboel LOFAR velden, LOFAR velden zijn velden die bestaan uit bijna honderd antennes die samenwerken om signalen goed op te vangen. Er worden constant nieuwe LOFAR velden gemaakt die allemaal een kabel lay-out nodig hebben.

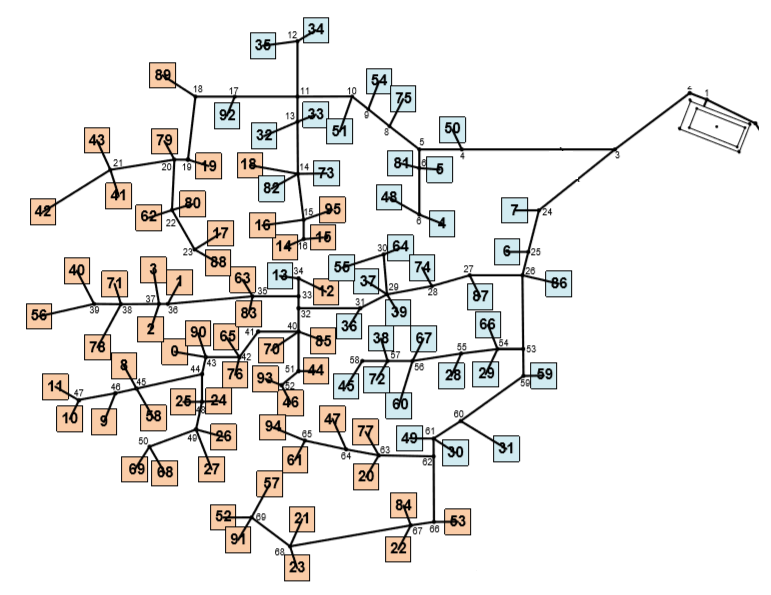
Elk veld is uniek, maar er zijn ook een aantal overeenkomsten. Elk veld bestaat uit dezelfde 96 antennes die allemaal relatief van elkaar op dezelfde plek staan, het verschil tussen de velden echter is dat elk valk een andere rotatie heeft. Een andere overeenkomst tussen de velden is dat elk veld een zogenaamd kabelhuisje heeft, dit is een huisje die ergens buiten het veld staat (voor elk veld is deze plek anders) die altijd het startpunt is voor alle kabels, vanaf dit huisje mag geen enkele kabel langer zijn dan 115 meter.

Er ook eisen aan hoe de kabels moeten liggen:

* De kabels kunnen niet onder antennes door, ze moeten er altijd ruim omheen gaan, in veel situaties liggen de antennes zo dicht op elkaar dat er geen kabels tussendoor kunnen, dit zorgt voor verplichte omwegen.
* Elke antenne heeft acht haringen, het blok is in de grond geslagen op de 4 hoeken en het midden van elk zijvlak. de kabels mogen niet langs deze acht punten, dus ze zullen er tussendoor moeten gaan.

Hier zijn twee LOFAR velden om de verschillen aan te duiden:

**Figuur 1:** *LOFAR veld PL611 “Lazy”*



**Figuur 2:** *LOFAR veld PL612 “Baldy”*

Let op, de antennes hebben op beide afbeeldingen dezelfde nummers, maar de velden staan anders gedraaid. Ook staat het hokje relatief van de antennes op een andere locatie.

## 1.4 Afbakening van de opdracht

ASTRON wil natuurlijk dat ik zuinige kabelplannen kan maken met mijn systeem, maar die kabelplannen moeten ook aan een aantal eisen voldoen:

* Het programma moet niet te zwaar/langzaam zijn
* De kabels kunnen fysiek niet door bepaalde punten van de antennes heen, daar moet mijn kabelplan dus ook rekening mee houden.
* Het kabelplan moet minstens even goed zijn als handgemaakte kabelplannen.
* Het is niet netjes om door antennes heen te verbinden, dat mag ik dus ook niet doen.

# Probleemanalyse

## 2.1 Eisen en wensen

Deze eisen en wensen zijn opgesteld aan de hand van overleg tussen de opdrachtgever en opdrachtnemer.

### 2.1.1 Eisen

Functionele eisen:

* De software moet een score leveren voor een kabelplan
* De software moet data kunnen uitlezen uit een juist geformatteerd .csv bestand
* De software moet een kabelplan genereren
* De software slaat het kabelplan netjes op in een .csv

Non-functionele eisen:

* De software is gedocumenteerd
* De software voldoet aan de ASTRON stijl-voorschriften
* De software is geschreven in Python3

### 2.1.2 Wensen

Functionele wensen:

* Het algoritme doet het minstens zo goed als handmatig gemaakt kabelplannen
* Een handige editor om zelf kabelplannen mee te bouwen

Non-functionele wensen:

* Het kabelplan ziet er stijlvol uit

## 2.2 Randvoorwaarden

Spullen die nodig zijn bij het goed uitvoeren van mijn stageopdracht worden waar mogelijk door ASTRON verzorgd, ASTRON zorgt ervoor dat ik mijn stageopdracht goed kan uitvoeren.

Benodigdheden bedragen:

* Werkplek (stoel, bureau, beeldscherm etc.)
* Begeleiding binnen ASTRON (stagebegeleider)
* Begeleiding buiten ASTRON (docent begeleider, school)
* account gegevens met toegang tot ASTRON voor werknemers

Als er spullen missen kan dit aangegeven worden bij de stagebegeleider en dan worden deze spullen geregeld.

## 2.3 Risicoanalyse

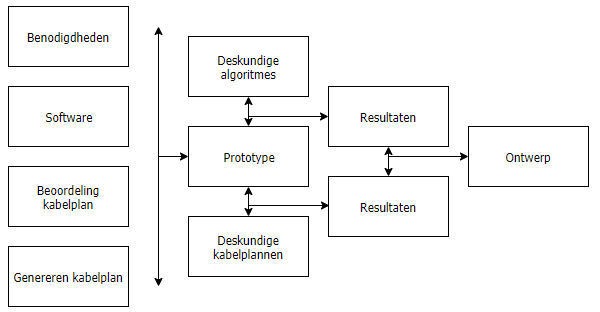
In principe is er geen risico bij het uitvoeren van de opdracht naast uiteraard het risico van teleurstelling. Omdat de opdracht los uitvoer van alle ASTRON systemen en er maar aan wordt gewerkt door één persoon kan alleen de uitvoerder zelf last hebben van eventuele fouten.

Alleen als het resultaat één op één wordt overgenomen en verwerkelijkt wordt, is er risico dat het veld niet klopt vanwege een te lange kabelafstand. Alleen zal dat risico niet meer aanwezig zijn wanneer het veld wordt overgenomen in het programma dat door ASTRON medewerkers wordt gebruikt aangezien ze dan het probleem zo kunnen zien en oplossen.

# Onderzoek

## 3.1 inleiding

De oplossing voor de opdracht zou gelijkenis kunnen hebben met een oplossing voor een bekend route probleem in de grafentheorie, maar de beste oplossing zou ook een unieke oplossing kunnen zijn die alleen toepasbaar ik bij dit probleem. In het onderzoek loop ik een aantal bekende problemen langs en bedenk een aantal potentiële oplossingen, daarna analyseer ik elke keuze en maak ik aan de hand van de resultaten een voorstel over welke oplossing de beste is.



**Figuur 3:** *Onderzoeksmodel*

## 3.2 Hoofd- en deelvragen

1. Welke bekende technieken zijn er?

1.1 Bieden deze technieken een goede oplossing?

1. Zijn er specifieke technieken voor dit probleem?

2.1 Bieden deze technieken een goede oplossing?

In de eerste hoofdvraag wordt ge-analyseer hoe goed de route problemen die in het theoretisch kader zijn beschreven van pas kunnen komen bij het oplossing van het probleem. De acht route problemen die behandelt worden zijn gebaseerd op (Graph theory, n.d.), volgens deze bron zijn dit de acht meest belangrijke route problemen.

In de tweede hoofdvraag word een oplossing die zelf bedacht is behandelt, deze oplossing is niet gebaseerd op andere oplossingen die in de literatuur te vinden zijn.

## 3.3 Methoden

Voor het opzoeken van informatie voor het theoretisch kader heb ik (Desk research, n.d.) toegepast. Daarna heb ik deze informatie gebruikt voor een (Available product analysis, n.d.). Voor het bedenken van oplossingen heb ik (Brainstorm, n.d.) toegepast.

Om het probleem te analyseren zodat ik aan de hand daarvan kan bepalen welke oplossingen toepasbaar zijn en welke niet een juiste oplossing bieden heb ik gebruik gemaakt van (Problem analysis, n.d.).

## 3.4 Theoretisch kader

Bekende route problemen in de grafentheorie worden toegelicht in acht verschillende secties. De acht secties zijn gebaseerd op (Graph theory, n.d.) en (Graph theory related route problems explained, n.d.).

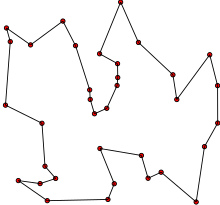
* + - 1. Is beantwoord m.b.v. (Travelling salesman problem, n.d.)
      2. beantwoord m.b.v. (Hamiltonian path, n.d.)
      3. beantwoord m.b.v. (Seven bridges of Königsberg, n.d.)
      4. beantwoord m.b.v. (Three cottage problem, n.d.)
      5. beantwoord m.b.v. (Seven bridges of Königsberg, n.d.)
      6. beantwoord m.b.v. (Dijkstra shortest path, n.d.)
      7. beantwoord m.b.v. (Minimum spanning tree, n.d.)
      8. beantwoord m.b.v. (Steiner tree, n.d.) en (Steiner tree problem, n.d.)

De minimale spanningsboom sectie van de resultaten is gebaseerd op de Minimum spanning tree implementatie van (Scipy Minimum spanning tree, n.d.).

Het Steiner boom onderdeel van de resultaten is gebaseerd op onderzoek dat is uitgevoerd met de (Networkx Steiner tree, n.d.) en (Geosteiner, n.d.) implementatie van de Steiner boom.

### 3.4.1 Bekende technieken

#### 3.4.1.1 Rondreizende handelaar probleem

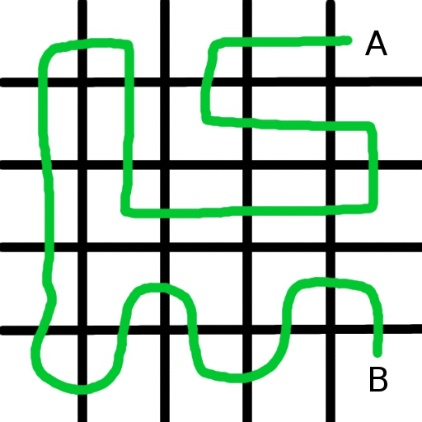


**Figuur 4:** *Oplossing van een Rondreizende handelaar probleem.*

Het Rondreizende handelaar probleem is als volgt: In **Figuur 4** is de input de rode puntjes, elk puntje symboliseert een stad. Nou is er een handelaar die graag door alle steden, rode puntjes, wil gaan om in elke stad zijn handelswaar te kunnen verkopen.

De oplossing van het probleem is de kortste route die de handelaar kan nemen om door alle steden te komen en dan uiteindelijk weer terug te komen waar hij was begonnen, de oplossing is dus een cyclus.

#### 3.4.1.2 Hamiltonian pad

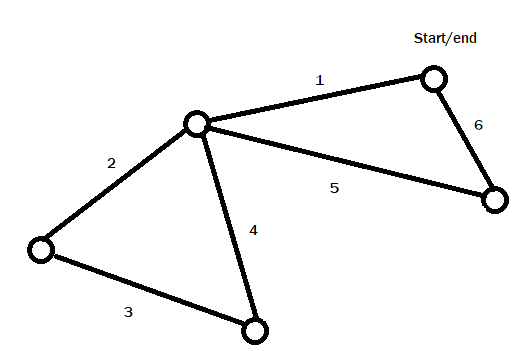


**Figuur 5:** *Voorbeeld Hamiltonian Pad*

Hamiltonian pad is onderdeel van de grafentheorie Route problemen, Het is een vrij simpel principe, “voor een gegeven raster, is het mogelijk om elk vakje precies 1 keer te bezoeken?”

Als het mogelijk is elk vakje maar 1 keer te bezoeken en dat dan het begin en eindpunt naast elkaar liggen is er sprake van een “Hamiltonian cyclus”.

#### 3.4.1.3 Chinese postbode probleem

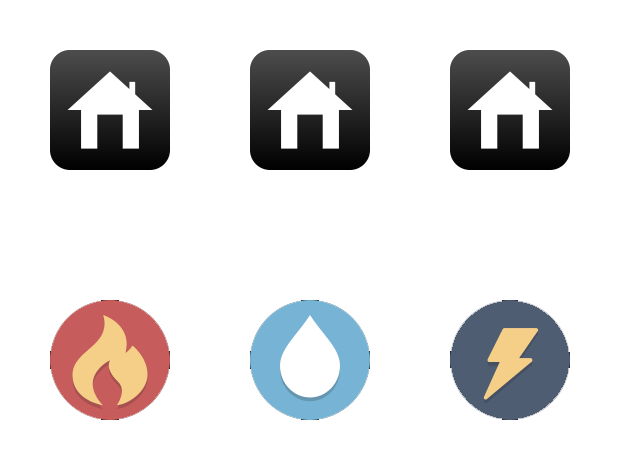


**Figuur 6:** *Chinese postbode probleem*

Het Chinese postbode probleem vraagt aan ons “Wat is de kortste route waarbij ik langs elke **weg** kom, en eindig waar ik begonnen ben?”

Zoals te zien in het plaatje loopt de postbode langs elke straat en komt weer terug waar hij begonnen was, het is een uitgebreide versie van “Rondreizende handelaar probleem” waarbij je niet alleen langs alle steden moet, maar elk pad minstens één keer moet betreden.

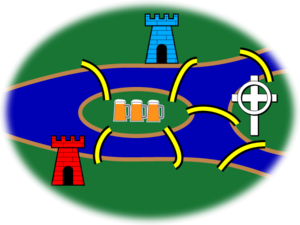
#### 3.4.1.4 Drie huizen probleem

**Figuur 7:** *ondersteunende illustratie voor uitleg Drie huizen probleem*

Het Drie huizen probleem valt ook onder route problemen. De vraag is of er een route bestaat die alle drie huizen een directe connectie laat hebben met gas, water en elektriciteit zonder dat ze elkaars kabels snijden.

Het antwoord is nee, niet zonder een extra dimensie toe te voegen. Wat je ook doet, zonder “vals te spelen” kan je ze niet alle drie aansluiten voor elk huis, er zal altijd 1 huis overblijven die geen toegang heeft tot gas of elektriciteit, echter als je een nieuwe dimensie toevoegt kan het wel.

#### 3.4.1.5 Zeven bruggen van Königsberg



**Figuur 8:** *ondersteunende illustratie voor uitleg Zeven bruggen van Königsberg*

Het Zeven-bruggen probleem valt ook onder route problemen, al is het meer een raadsel, de vraag luidt als volgt:

**De mannen in Königsberg houden wel van een biertje, de kroeg staat op het middelste eiland dus daar zijn ze vaak te vinden. Ze hebben een spel bedacht waarbij ze iedereen uitdagen om een rondje te lopen en alle bruggen precies 1 keer te passeren.**

Het probleem lijkt heel veel op een Hamiltonian Cyclus maar het is net eventjes iets anders omdat je over bepaalde vakken meerdere keren moet komen en omdat de hoeveelheid aanliggende vakken vaker verschilt.

Het is niet mogelijk om alle bruggen precies één keer te passeren.

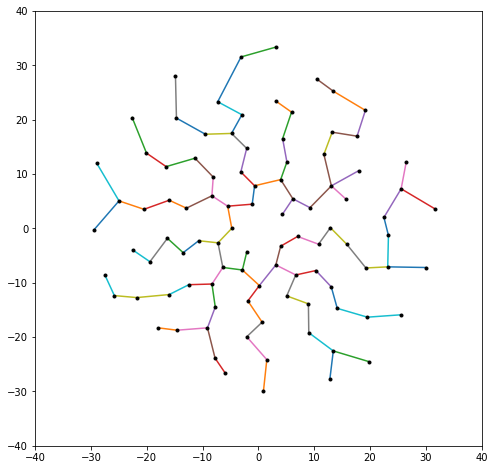
#### 3.4.1.6 Kortste pad probleem

De vraag van het kortste pad probleem is “Wat is het kortste pad van punt A naar punt B ?”. De vraag klinkt heel erg simpel maar er komt meer bij kijken, het kortste pad moet namelijk gebruik maken van beschikbare routes. Eigenlijk zou de vraag moeten zijn “Wat is het kortste pad van punt A naar punt B wanneer ik alle obstakels meereken en rekening houdt met alle mogelijkheden.

Er zijn veel “Pathfinders” die de kortste route kunnen zoeken tussen twee punten, dit probleem komt vaak voor bij games en daarom is er een ondersteuning voor.

Het kortste pad probleem komt ook aan bod bij het LOFAR probleem. Om te berekenen of de kabel vanaf het kabelhuisje naar een antenne langer of korter is dan 115 meter wordt gebruik gemaakt van het Dijkstra algoritme.

#### 3.4.1.7 Minimaal opspannende boom



**Figuur 9:** *Een Minimaal opspannende boom van de antennes van het veld in Ierland.*

De Minimaal opspannende boom is wanneer alle punten van een set met elkaar verbonden zijn met zo’n min mogelijke afstand aan lijnen, zoals in het figuur te zien zijn alle punten met elkaar verbonden door lijnen. De Minimaal opspannende boom is een versoepelde versie van het handelsreizigersprobleem; De punten moeten met zo’n laag mogelijke totale afstond verbonden worden maar het hoeft niet te worden gedaan in een cyclus.

#### https://i.gyazo.com/768090c2f73f69231190cc5ac6701009.png3.4.1.8 Steiner boom

**Figuur 10:** *Een Steiner boom van de zes punten*

De Steiner boom is het zelfde principe als de Minimaal opspannende boom, alleen gaat de Steiner boom nog een stap verder. Alle punten van een set moeten weer met elkaar worden verbonden maar in dit geval mogen er ook extra zogenaamde “Steiner points” worden toegevoegd die de totale spanning van de boom nog meer verminderen. In **Figuur 10** zijn vier Steiner punten te zien, tussen elke zet van zwarte puntjes kom de twee lijnen samen, die ontmoeting heet een Steiner punt. Daarna komen alle lijnen samen in het midden in een nieuwe Steiner point.

Een Minimaal opspannende boom zou **Figuur 10** hebben opgelost als een soort hexagoon die één zijde mist afhankelijk vanaf welk punt het algoritme was begonnen met rekenen.

## 3.5 Resultaten

### 3.5.1 Welke bekende technieken zijn er?

Zoals toegelicht in het theoretisch kader heb ik de acht bekendste problemen in de grafentheorie bekeken, deze acht problemen neem ik hier één voor één door een analyseer ik waarom ze wel of niet van pas kunnen komen.

* Rondreizende handelaar probleem

Het Rondreizende handelaar probleem is vergelijkbaar met het probleem van de LOFAR velden, de duidelijke overeenkomt is dat het gaat om het verbinden van alle punten en dat er een duidelijk startpunt is. Bij het Rondreizende handelaar probleem moeten alle steden direct met elkaar verbonden worden, voor LOFAR zou dat betekenen dat antennes direct met elkaar verbonden worden en dat er wordt “doorverbonden”, dat is geen optie. Daarnaast wil de Rondreizende handelaar altijd thuis komen zodat hij weer opnieuw kan beginnen of even een vakantie kan nemen in zijn eigen stad, maar als de antennes van de LOFAR velden weer terug werden verbonden aan hij startpunt, zou dat een hoop geld kosten aan overbodige kabels.

Het Rondreizende handelaar probleem is niet relevant.

* Hamiltonian pad

Het maken van een Hamiltonian pad is zeker een route problem, maar er zijn verder geen relevante overeenkomsten met het probleem van de LOFAR velden, Hamiltonian pad algoritme is dus geen potentiële oplossing.

* Chinese postbode probleem

Het Chinese postbode probleem komt in de buurt van het Rondreizende handelaar probleem, en om dezelfde redenen is het Chinese postbode probleem ook niet een goeie oplossing voor het probleem van de LOFAR velden. Het Chinese postbode probleem wil namelijk ook terug komen waar hij begonnen was, en deze keer wordt er niet alleen door alle steden gereisd, maar ook elke straat moet meegepakt worden. Het is geen relevant probleem.

* Drie huizen probleem

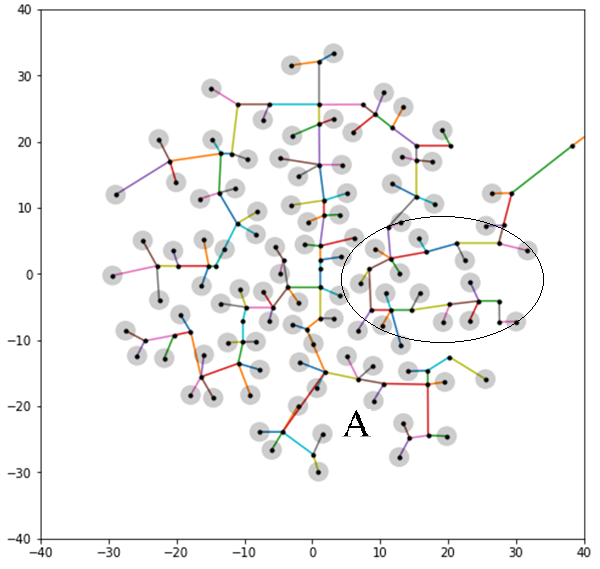
Het Drie huizen probleem is meer een raadsel waar het probleem van de LOFAR velden niks aan heeft. Maar het is wel een erg interessant idee hoe je de antennes zou kunnen verbinden door de kabels onder de antennes door te laten gaan in plaats van er omheen, alleen kost het meer meters om dieper grond in te gaan dan hoeveel meters het kost om er gewoon omheen te gaan.

* Zeven bruggen van Königsberg

Net zoals het Drie huizen probleem lijkt Zeven bruggen van Königsberg eerder op een raadsel. Helaas kan het raadsel op geen enkele manier gebruikt worden voor het oplossen van het Probleem van de LOFAR velden omdat er geen sterke gelijkenissen tussen de twee zijn.

* Kortste pad probleem

Het Kortste pad probleem is wel degelijk een route problem, en waarschijnlijk ook het meest bekende probleem van allemaal. Het Kortste pad probleem komt zeker van pas bij het oplossen van het LOFAR velden probleem om het aantal meters aan kabel te kunnen berekenen. Ondanks dat het van pas komt bij de oplossing, is het meer een oplossing voor een klein onderdeel. Het grote probleem, het leggen van de paden, kan het Kortste pad probleem helaas niet.

* Minimale spanningsboom

**Figuur 11:** *Minimale spanningsboom van de huidige indeling van een LOFAR-veld in Polen.*

De Minimale spanningsboom algoritme lijkt de eerste echte hulp te zijn, wanneer je een Minimale spanningsboom spant over een set punten waarbij de antenne punten zwaarder zijn dan de piket punten zal de Minimale spanningsboom in de buurt komen van een prima oplossing. Helaas zijn er een paar problemen die ge-highlight zijn in **figuur 11**; Zoals te zien in het cirkeltje aan de rechter zijde van het figuur heeft het Minimale spanningsboom algoritme een omweg geprioriteerd, deze weg zorgt inderdaad voor een lagere totale spanning maar de antenne rechts onderin het cirkeltje heeft nu misschien een langere kabel nodig. Kortom, De Minimale spanningsboom houdt alleen rekening met de totale spanning van de boom, niet met de kosten die langere kabels met zichzelf meebrengen.

Het tweede probleem is dat het Minimale spanningsboom algoritme als het ware blind is voor alles behalve de spanning, dus ook voor de veiligheidsvlakken van de antennes. Zoals te zien in het figuur heeft elke antenne een grijs vakje die een soort veiligheidsvlak indiceert, het is niet de bedoeling dat er kabels doorheen gaan behalve de kabels die rechtstreeks op die antenne zelf afgaan omdat er een risico is dat de kabels kapot gaan bij het aanleggen van de antennes.

Tot slot maakt het Minimale spanningsboom algoritme niet zelf piketten aan, en hij kan ze ook niet negeren wanneer ze geen nut hebben. Deze drie problemen samen betekenen dat de Minimale spanningsboom niet goed genoeg is om het probleem van de LOFAR velden op te kunnen lossen.

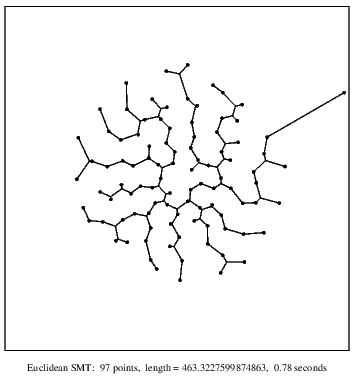
* Steiner boom

De Steiner boom komt het meest in de buurt van het probleem van de LOFAR velden, in theorie zou een versie van de Steiner boom een erg elegante en goede oplossing zijn.

Deze specifieke versie van de Steiner boom zou een perfect, of een heel erg goed, resultaat moeten opleveren. Helaas is het Steiner boom een erg lastig probleem en zijn er niet veel algoritmes beschikbaar die een Steiner boom kunnen maken, er zijn er maar 2; Geosteiner in C, en Networkx in Python.

De Geosteiner implementatie heeft een geeft erg goede oplossingen en werkt bijzonder snel, aan de hand van een paar tests lijkt het een bijna perfecte oplossing te zijn die weinig problemen met zich meebrengt. Helaas kan je alleen niks met Geosteiner behalve het maken van punt-collecties en het genereren van Steiner bomen. Om het probleem van de LOFAR velden op te lossen heb je meer nodig dan alleen een Steiner boom, zoals eerder vermeld; Een aangepaste, probleem specifieke versie. Om het probleem aan te pakken met Steiner boom is een vereiste dat je de Steiner boom zou kunnen behangen met gewichten, zodat het algoritme geen antennes door gaat verbinden en zodat je kan voorkomen dat het algoritme de aanpassende kabel lengtes gaan negeren.

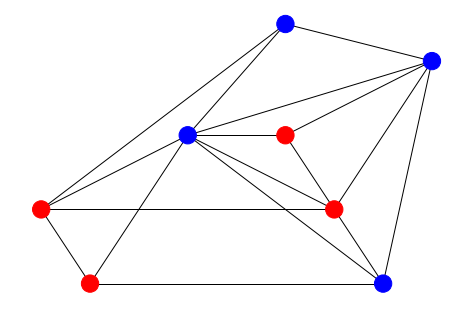
De boom behangen met gewichten zou je kunnen doen wanneer je de optionele lijnen zelf mee kunt geven aan het Steiner tree algoritme en je bij het meeleveren van de lijnen, ook een gewicht op elke lijn kunt plakken. Standaard is het gewicht van een lijn het aantal meters dat de lijn lang is maar als je zelf een gewicht zou kunnen invullen dan kan je lijnen die er voor zorgen dat een antenne een 115 meter kabel moet gebruiken i.p.v. een 85 meter kabel bestraffen door er deze lijn extra zwaar te maken, hierdoor zal het algoritme gebruik maken van lichtere lijnen die deze situatie kunnen voorkomen.

**Figuur 12:** *De Steiner boom van een LOFAR-veld in Polen gemaakt met Geosteiner*

Zoals in **Figuur 12** te zien, is een onaangepaste Steiner boom niet wat ASTRON zoekt als oplossing voor het probleem van de LOFAR velden. Het algoritme negeert net zoals de Minimale spanningsboom gewoon de kosten die de lengte van de kabels met zich mee brengen omdat er geen gewicht aan de boom zijn toegevoegd. Daarnaast zijn ook erg weinig piketten te zien in dit figuur, dat komt omdat de Steiner boom gewoon antennes doorverbind wanneer dat de goedkopere oplossing voor het totaal gewicht. Hieruit kan je concluderen dat je niks hebt aan een Steiner boom zolang je niet de gewichten van de lijnen kunt aanpassen.

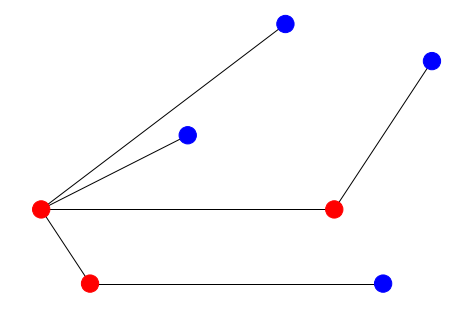
Gelukkig is er nog een andere implementatie beschikbaar, namelijk Networkx.

Networkx laat toe dat je lijnen plaatst die het Steiner boom algoritme kan kiezen te gebruiken of weg te laten, ook laat Networkx toe dat je gewichten aan elke lijn hangt, met de mogelijkheid om deze twee dingen te doen kan je het Steiner boom algoritme zo manipuleren dat het altijd piketten prioriteert en daarmee voorkom je dat antennes worden doorverbonden.



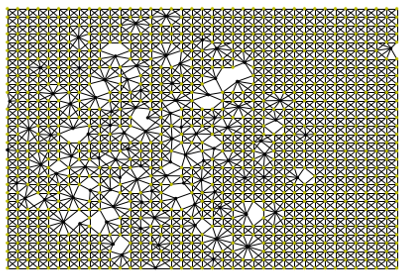
**Figuur 13:** *Een graaf met Terminals(blauw), Steiner points(rood) en lijnen die zo wegen dat ze vermijden dat de Terminals meer dan één connectie leggen.*

In **Figuur 13** is een klein voorbeeldje te zien van de Networkx Steiner boom, de rode puntjes stellen de piketten voor, en de blauwe puntjes zijn de antennes. Het doel dat we willen bereiken is dat we een figuur krijgen waarbij alle blauwe puntjes indirect met elkaar zijn verbonden, dat zou bewijzen dat het algoritme geen antennes door verbind, ook niet wanneer dit voor een kleinere spanning zorgt.

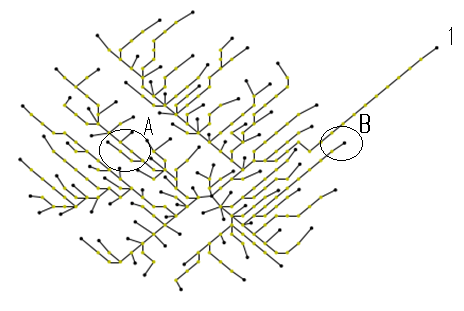
**Figuur 14:** *Het resultaat van Figuur 14 geproduceerd met de Networkx Steiner boom functie*

Een groot succes, de piketten zijn met elkaar verbonden en de antennes zijn allemaal ook indirect met elkaar verbonden.

Met deze kennis kan er verder gewerkt worden met de Networkx Steiner boom.

**Figuur 15:** *Een Grafiek van een LOFAR-veld in Polen zoals meegegeven aan de Networkx Steiner boom functie.*

**Figuur 15** is een Pools LOFAR veld, het gehele oppervlak van het veld is gevuld met mogelijke piketten en lijnen, het is zo gemaakt dat de piketten nooit te dicht op de antennes staan en de lijnen nooit door antennes heen gaan, wat de Networkx Steiner boom nu zal doen is het zoeken van de kortste route die alle zwarte punten aan elkaar verbind, en daarna gooit het alle overbodige piketten en lijnen weg.

**Figuur 16:** *Het resultaat van Figuur 12 geproduceerd door Networkx de Steiner boom functie*

Zoals te zien in **Figuur 16** bij het cirkeltje gemarkeerd met een “A”, maakt de Networkx Steiner boom geen goede keuzes, gewichten worden soms genegeerd omdat de formule waarmee de Steiner boom mee wordt berekent niet goed genoeg is. Daarnaast is bij cirkel “B” te zien dat het algoritme een gigantische omweg maakt vanuit het startpunt “1”, dat komt omdat het algoritme niet weet waar het startpunt ligt. Dat zou kunnen worden opgelost maar aangezien deze implementatie geen goeie oplossing kan uitrekenen is het niet nodig om verdere ontwikkeling uit te voeren.

### 3.5.2 Zijn er specifieke technieken voor dit probleem?

Natuurlijk is er ook de optie om zelf oplossingen te bedenken, want die kunnen er ook zijn.  
  
De zelf-ontwikkelde techniek die behandeld wordt kan het beste worden beschreven als “bestaande velden kopiëren”. Aangezien er al twaalf buitenlandse velden bestaan die als voorbeeld kunnen functioneren (de andere velden hebben een out-datet lay-out die niet meer gebruikt mag worden), is het goed mogelijk om de bestaande velden als het ware dynamisch te maken door ze allemaal te herbouwen op hetzelfde middelpunt met een licht aangepaste kabelligging die rekening houdt met de rotatie die de velden zal doormaken. Na het maken van de twaalf bestaande velden kan er natuurlijk ook een uitbreiding komen in het assortiment door bestaande velden deels te kopiëren en daarmee een andere rotatie te kunnen voorzien van een goed veld.  
  
Dit is allemaal mogelijk omdat alle velden precies dezelfde antennes hebben, de antennes lijken alleen anders van elkaar te liggen omdat het veld altijd een andere rotatie heeft, maar wanneer je alle velden naar dezelfde rotatie roteert zijn de enige verschillen de locatie van het kabelhuisje en de rotatie van de antennes (de antennes zijn vierkante blokken). Het kabelhuisje is natuurlijk wel een erg belangrijke factor in het ontwerp van de LOFAR velden omdat alle kabels vanuit dit punt komen en dus vanuit dit punt de kortst mogelijke afstond moeten leggen.  
  
Door een hoop “voorbeeld velden” te maken die het programma kan gebruiken, kan je goede velden voor alle posities van het kabelhuisje in een cirkel van 360 graden voorzien, en dat is het principe van deze techniek.

Nadat er een veld is gekozen kan er een algoritme worden losgelaten op het verplaatsen van de piket posities zodat de kabels niet meer langs haringen zullen lopen waar ze niet langs mogen komen vanwege veiligheidsredenen.   
  
Het volgende probleem is kiezen welk veld het beste veld is voor de locatie van het kabelhuisje dat je zoekt, natuurlijk is daar ook een oplossing voor. Door het maken van een speciale kostenfunctie die het veld beoordeelt door er een prijskaartje aan te hangen kan je altijd het beste veld kiezen door het veld met de laagste prijs te selecteren. Dit werkt alleen als de prijzen accuraat zijn, daarom moet de kostenfunctie gebaseerd worden op de daadwerkelijke kosten en die erg accuraat verwerken.

## 3.6 Analyse van onderzoeksresultaten

De eerste zes route problemen zijn eigenlijk niet relevant voor mijn probleem, maar het is toch goed dat ik ze onderzocht heb want nu kan ik ze met zekerheid uitsluiten. Ook heb ik het één en ander geleerd over de werking van de meeste grafentheorieën, en de generieke gedachtegang die er achter zit.

De Minimale spanningsboom is een interessant probleem met enige relatie tot mijn probleem, vandaar het extra onderzoek. Maar bij nader inzien is het toch niet helemaal de oplossing.

De Steiner boom zou een goeie oplossing zijn, helaas zijn er online geen implementaties beschikbaar die precies bieden wat benodigd is om het LOFAR velde probleem op te lossen en daarom is dat eigenlijk ook geen goede keuze.

De zelf-ontwikkelde techniek waarbij het kopiëren van bestaande velden centraal staat lijkt uit dit onderzoek de beste oplossing te zijn, aangezien het de enige oplossing is doe voldoet aan alle eisen en wensen van de opdracht die eerder in dit verslag zijn vermeld.

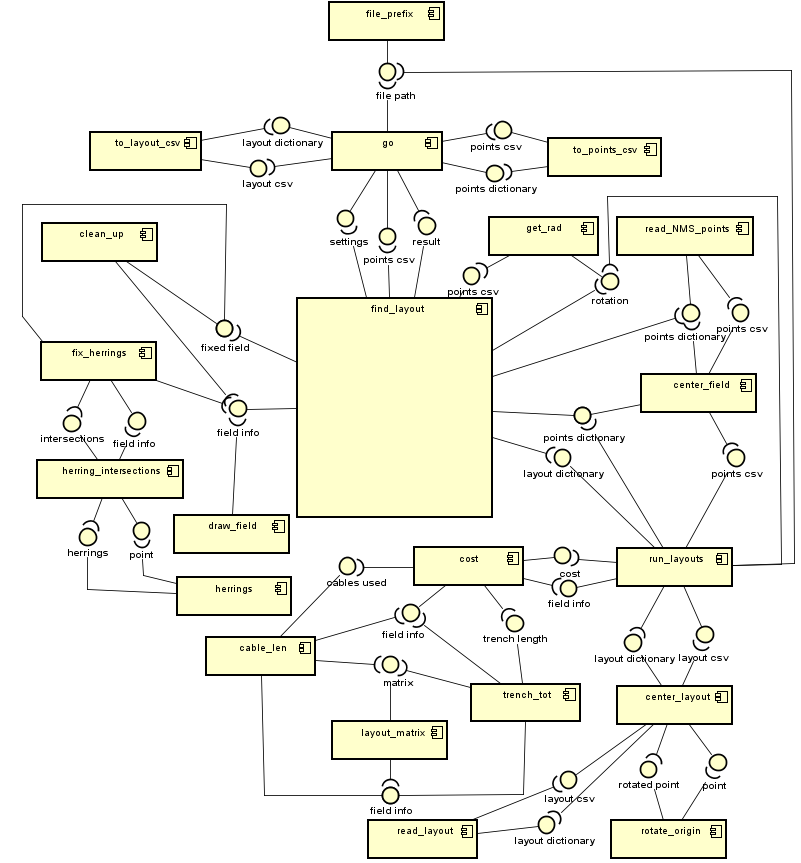
## 3.7 Conclusie en besluit

In de grafentheorie zijn een paar route problemen te vinden die redelijk nauw in de buurt komen van het probleem van de LOFAR velden, de meest veelbelovende route problemen waren de Minimale spanningsboom, en de Steiner boom. Helaas is de Minimale spanningsboom simpelweg geen oplossing, het probleem komt redelijk in de buurt maar vanwege de unieke factoren die een grote rol spelen in het LOFAR probleem kan de Minimale spanningsboom niet worden toegepast als een oplossing.  
  
De Steiner boom zou niet alleen een goede oplossing zijn geweest maar ook een erg elegante oplossing, het gebruiken van een slim algoritme die het probleem tot in de puntjes perfect oplost zou een droom zijn geweest maar helaas mocht het niet zo zijn. Door het tekort aan implementaties beschikbaar is er geen implementatie aanwezig die de gebruiker toelaat om de velden aan te passen, en ook nog goede resultaten levert op het gebied van totale boom spanning. Om deze redenen is de Steiner boom geen oplossing die gebruikt kan worden voor de LOFAR velden bekabeling.  
  
De techniek die beschreven wordt in hoofdstuk 3.5.2 is de beste oplossing, omdat het de enige oplossing is die een compleet resultaat kan produceren.   
  
De keuze is gevallen op het hergebruiken van bestaande velden, deze oplossing biedt goede velden die voldoen aan alle eisen die worden gesteld aan de LOFAR velden, het enige nadeel is dat het algoritme redelijk in waarde daalt als ASTRON ooit kiest om een andere kabel lay-out te gebruiken voor nieuwe LOFAR velden, maar in dat geval zou iemand de velden die in het programma staan kunnen vervangen door voorbeeldvelden van de nieuwe lay-out. Natuurlijk was het Steiner boom algoritme de geprefereerde keuze geweest aangezien die in feite “lay-out dynamisch” is en kan worden toegepaste op elke lay-out, maar helaas was het niet realistisch mogelijk om een goed algoritme te maken die gebruik maakt van de Steiner boom.

4. Ontwerp

4.1 UML diagrammen

### 4.1.1 Architectuurniveau als een component diagram:



### https://i.gyazo.com/8c5ce7eef97b7eaa6df51224345a38d2.png4.1.2 Detailniveau als flowchart:

4.2 Motivatie

De keuze is gevallen op dit ontwerp tijdens het project, aan de hand van advies van Tammo Jan ben ik het project imperatief gaan programmeren.

Aangezien de enige eis/wens die verbonden is aan het ontwerp “De software voldoet aan de ASTRON stijl-voorschriften” is, is er natuurlijk geen betere keuze dan het ontwerp te gebruiken dat is aanbevolen door een ASTRON medewerker.

4.3 Tools

Om dit ontwerp werkelijkheid te maken is alleen een editor nodig waarin python3 code gemaakt en ge-compiled kan worden. Binnen de python3 code zal gebruik worden gemaakt van verscheidene libraries die online verkrijgbaar zijn;

* sys
* numpy
* scipy
* shapely
* matplotlib
* os
* Descartes
* CSV
* math

4.4 Testen

### 4.4.1 Validatie eisen

Functionele eisen:

* De software moet een score leveren voor een kabelplan.

***Dit ontwerp bevat een cost-functie die een kabelplan beoordeeld door er een geschatte prijs aan te binden.***

* De software moet data kunnen uitlezen uit een juist geformatteerd .csv bestand.

***Dit ontwerp bevat read\_csv functions die verschillende .csv bestanden kan uitlezen.***

* De software moet een kabelplan genereren.

***Dit ontwerp genereerd kabelplannen als eindproduct.***

* De software slaat het kabelplan netjes op in een .csv.

***Dit ontwerp kan met de “xxx\_to\_csv” functie informatie naar een .csv bestand schrijven.***

* De software is gedocumenteerd

***Dit ontwerp kan fatsoenlijk gedocumenteerd worden.***

* De software voldoet aan de ASTRON stijl-voorschriften

***Dit ontwerp voldoet.***

* De software is geschreven in Python3

***Dit ontwerp kan geschreven worden in Python3.***

### 4.4.2 Validatie wensen

Functionele wensen:

* Het algoritme doet het minstens zo goed als handmatig gemaakt kabelplannen

***Dit ontwerp produceert gemiddeld betere velden dan de handgemaakte velden.***

* Een handige editor om zelf kabelplannen mee te bouwen

***Dit ontwerp bevat geen editor.***

* Het kabelplan ziet er stijlvol uit

***Dit ontwerp kan een extreem stijlvol kabelplan genereren.***

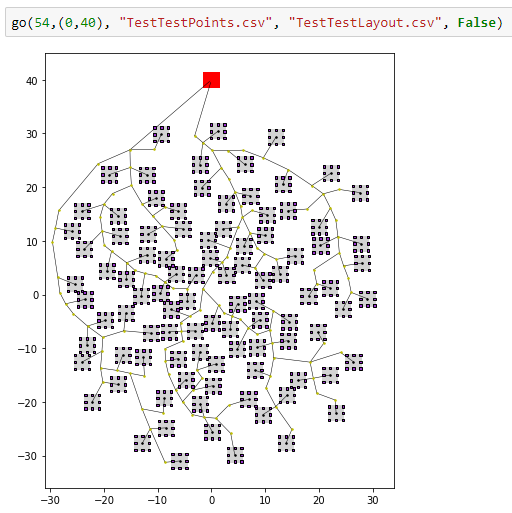
5. Realisatie

5.1 Eindproduct

Het eindproduct maakt erg veel gebruik van eerder gemaakte LOFAR velden, aan de hand van goede voorbeelden weet het project een goede oplossing te vinden voor elke mogelijke rotatie. Het project bevat Tweeëntwintig “voorbeeldvelden”. Wanneer je een nieuw veld wilt genereren geef je simpelweg de rotatie van het veld, en de locatie van het kabelhuisje. Het project maakt voor elk voorbeeldveld in combinatie met deze rotatie en dit kabelhuisje een veld, voor elk veld wordt door een kostenfunctie berekend hoeveel elk veld ongeveer kost, het veld dat uiteindelijk het minste kostte wordt teruggeven als het beste veld.

Na dat dit gedaan is zal het project het veld opschonen, door alle onnodige onderdelen te verwijderen en alle lijnen te corrigeren zodat ze niet langs plaatsen gaan waar ze niet langs kunnen gaan.

Dit “opschonen” word gedaan in twee porties, de eerste is een algoritme die redelijk lang duurt (ongeveer anderhalf tot twee uur). Deze eerste portie kijkt naar alle kabels die langs een paars blokje gaan, de paarse blokjes op elke antenne zijn de acht plaatsen waar haringen staan, wanneer zo’n kabel is gevonden zal het algoritme het piket waar de kabel uit komt verplaatsen zodat de kabel niet langer door de haring gaat. De tweede portie is het opruimen van onnodige piketten, in een hoop gevallen staan er piketten die voor het berekenen belangrijk waren maar die nu de berekeningen klaar zijn bijzonder weinig nut hebben, het gaat hier over piketten die de kabel geen nieuwe richting op sturen maar gewoon in de zelfde richting sturen als de richting waarmee de kabel de piket in komt.

**Figuur 17:** De line code en het resultaat, dit is het eindproduct

“go” is de functie die alles begint.

“54” is de rotatie van het gewenste veld.

“(0, 40)” zijn de gewenste x, en y locatie van het kabelhuisje (het rode blok).

“TestTestPoints.csv” en “TestTestLayout.csv”  
zijn de gewenste namen van de Points.csv en Layout.csv die worden opgeslagen in de standaard folder.

“False” of “True” indiceert of de gebruiker graag een veld wil genereren die klaar is in minder dan een minuut maar bepaalde intersecties met de antennes bevat, of dat de gebruiker graag een veld zonder intersecties wil genereren in ongeveer anderhalf uur (True is anderhalf uur, False is één minuut).

5.2 Validatie eisen en wensen

### 5.2.1 Validatie eisen

Functionele eisen:

* De software moet een score leveren voor een kabelplan.

***Beoordeling in de vorm van €***

* De software moet data kunnen uitlezen uit een juist geformatteerd .csv bestand.

***Niet meer nodig aangezien de informatie altijd hetzelfde blijft.***

* De software moet een kabelplan genereren.

***De software kan een kabelplan genereren***

* De software slaat het kabelplan netjes op in een .csv.

***De kabelplannen worden in twee .csv’s gezet tijdens het genereren***

* De software is gedocumenteerd

***Docstring bovenaan elke functie***

* De software voldoet aan de ASTRON stijl-voorschriften

***Voldoet aan pep8***

* De software is geschreven in Python3

***Software werkt in Python 3.6 of latere versies***

### 5.2.2 Validatie wensen

Functionele wensen:

* Het algoritme doet het minstens zo goed als handmatig gemaakt kabelplannen

***Gemiddelde resultaat van de software is beter dan de veertien velden die al bestonden***

***Zie Hoofdstuk 13***

* Een handige editor om zelf kabelplannen mee te bouwen

***Niet inbegrepen bij het project aangezien het niet gebruikt zal worden.***

* Het kabelplan ziet er stijlvol uit

***Klopt.***

# 6. Conclusie en aanbeveling

Met behulp van de LOFAR kabel generator kan iedereen met een computer met bijzonder weinig moeite een LOFAR veld genereren, daarmee is het probleem opgelost. Het genereren kost ongeveer 1 minuut, maar in het geval dat je een veld wilt krijgen die bepaalde problemen oplost die je ook redelijk snel met de hand zou kunnen doen, zal het ongeveer anderhalf tot twee uur duren.

Het project voldoet aan de meeste wensen en eisen, de wens waaraan het niet voldoet was bij nader inzien niet meer nodig omdat het op een andere manier wordt opgelost.

Er zijn ook nog een paar verbeterpunten, het eerste punt is dat de oplossing, ondanks dat het erg accuraat is, nog accurater kan. Door nóg meer en nóg betere voorbeelden toe te voegen aan het project zullen de oplossingen geleidelijk in kwaliteit stijgen.

Het tweede verbeterpunt is andere antenne lay-outs, aangezien het natuurlijk mogelijk is dat de LOFAR velden in de toekomst een soort “update” krijgen en dat de lay-out veranderd zal deze oplossing niet meer effectief zijn, in dat geval zullen nieuwe voorbeelden voor deze nieuwe lay-out moeten worden toegevoegd die vergelijkbaar zijn met de huidige voorbeelden op gebied van kwaliteit.

7. Reflectie

7.1 Onderzoek

Na uitbundig onderzoeken naar alle bekende route problemen in de grafentheorie heb ik verder onderzoek gedaan naar meest veelbelovende routeproblemen, Ik heb een hoop geleerd en ben tevreden over hoe het onderzoek is gegaan, het is redelijk jammer dat het me uiteindelijk alleen maar tijd heeft gekost en niet veel heeft opgeleverd voor het eindproduct maar dat maakt het niet een minder goed onderzoek, op deze manier heb ik geleerd dat ik wel in de goede hoek zocht, maar dat er niks te vinden was.

7.2 Product

Ondanks dat ik liever een mooie, slimme oplossing had gemaakt met behulp van het Steiner tree probleem, ben ik tevreden met mijn huidige oplossing. Het product is niet perfect omdat een update in LOFAR antenne lay-out het product nutteloos zou maken. Maar daar was het product ook niet voor bedoel, de opdracht was om een systeem te maken die een lay-out ontwerpt voor de huidige stand van zaken en dan doet het product erg goed.

7.3 Eigen handelen

Ik ben tevreden over mijn eigen handelen, ik heb effectief samengewerkt met mijn opdrachtgeven (Tammo Jan Dijkema) en deskundige op het gebied van LOFAR velden kabel lay-outs (Jeroen Herrewijn). Ook ben ik tevreden over mijn overleg over de beste manier om bepaalde code fragmenten aan te pakken met Tammo Jan Dijkema en Peter van der Velde

7.4 Leerdoelen

Ik heb al mijn leerdoelen bereikt en ben tevreden over hoe ver ik het doel ben gegaan bij een aantal (zoals ervaring opdoen met Python3)

8. Bronnen

*Available product analysis*. (sd). Opgehaald van ictresearchmethods: http://ictresearchmethods.nl/Available\_product\_analysis

*Brainstorm*. (sd). Opgehaald van ictresearchmethods: http://ictresearchmethods.nl/Brainstorm

*Desk research*. (sd). Opgehaald van Management study guide: https://managementstudyguide.com/desk-research.htm

*Dijkstra shortest path*. (sd). Opgehaald van sznur: http://sznur.com/dijkstra-shortest-path-ios

*directed Chinese postman* . (sd). Opgehaald van m9.ma.tum: https://www-m9.ma.tum.de/graph-algorithms/directed-chinese-postman/index\_en.html

*Geosteiner*. (sd). Opgehaald van Geosteiner: http://www.geosteiner.com/

*Geosteiner manual*. (sd). Opgehaald van geosteiner: http://www.geosteiner.com/geosteiner-5.1-manual.pdf

*Graph theory*. (sd). Opgehaald van wikipedia: https://en.wikipedia.org/wiki/Graph\_theory

*Graph theory related route problems explained*. (sd). Opgehaald van docsity: https://www.docsity.com/en/news/education-2/graph-theory-related-route-problems-explained-gifs/

*Hamiltonian path*. (sd). Opgehaald van mathworld: http://mathworld.wolfram.com/HamiltonianPath.html

*Minimum spanning tree*. (sd). Opgehaald van Wikipedia: https://en.wikipedia.org/wiki/Minimum\_spanning\_tree

*Networkx Steiner tree*. (sd). Opgehaald van Networkx: https://networkx.github.io/documentation/latest/reference/algorithms/generated/networkx.algorithms.approximation.steinertree.steiner\_tree.html?highlight=steinertree

*Problem analysis*. (sd). Opgehaald van ictresearchmethods: http://ictresearchmethods.nl/Problem\_analysis

*Scipy Minimum spanning tree*. (sd). Opgehaald van docs.scipy: https://docs.scipy.org/doc/scipy-0.15.1/reference/generated/scipy.sparse.csgraph.minimum\_spanning\_tree.html

*Seven bridges of Königsberg*. (sd). Opgehaald van Wikipedia: https://en.wikipedia.org/wiki/Seven\_Bridges\_of\_K%C3%B6nigsberg

*Steiner tree*. (sd). Opgehaald van geeksforgeeks: https://www.geeksforgeeks.org/steiner-tree/

*Steiner tree problem*. (sd). Opgehaald van Wikipedia: https://en.wikipedia.org/wiki/Steiner\_tree\_problem#cite\_note-FOOTNOTESmithWinter1995361-10

*Task analysis*. (sd). Opgehaald van ictresearchmethods: http://ictresearchmethods.nl/Task\_analysis

*Three cottage problem*. (sd). Opgehaald van academickids: https://academickids.com/encyclopedia/index.php/Three\_cottage\_problem

*Travelling salesman problem*. (sd). Opgehaald van wikipedia: https://en.wikipedia.org/wiki/Travelling\_salesman\_problem

9. Leerdoelen

* Ervaring opdoen met Python3
* Kennis maken met astronomie
* Ervaring opdoen met het gebruiken van algoritmes
* Kennis vergaren over de bestaande algoritmes
* Kennis maken met het bedrijfsleven
* Leren over grafentheorieën

# 10. Planning



---------- 3 September

Begin stage, inwerken en begin 10 September---- maken aan de interactieve editor   
 Editor uitwerken en laat editor  
 bestaande plannen uitlezen ----------- 17 September  
 Plan van aanpak uitwerken, code 24 September ---- editor afmaken, daarna opschonen  
meerdere kabelplannen inlezen  
en er een score aan geven ---------1 Oktober  
 Scoreberekening uitwerken en   
 verbeteren met feedback van  
 15 Oktober----- domeinexperts  
 Ontwikkelen algoritme met   
 minstens een output ---------12 November  
 19 November---- Eerste prototype opleveren   
Verbetering algoritme aan de   
hand van feedback eerste proto ---------26 November  
 Algoritme verder ontwikkelen  
 3 December------  
Tweede prototype opleveren,  
Deze bevat documentatie  
 ---------17 December  
 Eerste release

7 Januari ---------  
Definitieve versie eindrapport  
 ------- 14 Januari  
 Definitieve release, met volledige  
 documentatie   
 21 Januari--------   
 Eindpresentatie

------28 Januari  
 Definitief verslag ASTRON

11. Versiebeheer

|  |  |  |
| --- | --- | --- |
| 1.0 | 17/10/2018 | Onderzoek gemaakt, nog incompleet. |
| 1.1 | 19/10/2018 | Skelet gemaakt met alle nodige onderdelen. Planning en leerdoelen gemaakt. |
| 1.2 | 22/10/2018 | Inleiding en probleemanalyse geschreven, stijl aangepast. |
| 1.3 | 30/10/2018 | Methoden, bronnen en theoretisch kader update. |
| 1.4 | 17/12/2018 | Hoofdstuk 5, 6, 7, 12 & 13 gevuld. |
| 1.5 | 07/01/2019 | Ontwerp toegevoegd, feedback verwerkt |

12. Logboek

moeten worden toegevoegd die vergelijkbaar zijn met de huidige voorbeelden op gebied van kwaliteit.

Maandag 3 September – Maandag 17 September:  
Begin van de stage, na een aantal kleine dingen zoals inwerken, opening APERTIF bijwonen en workshop op school bijwonen, Heb ik de GUI gemaakt waarvan ik de rest van de stage erg veel gebruik van ga maken.

Dinsdag 18 September - Donderdag 4 Oktober:  
Kennis gemaakt met Jeroen(domeinexpert), Plan van aanpak gemaakt, gewerkt aan kleine functies voor het project zoals het inladen van informatie uit een .csv bestand.  
Verder nog een hoop uitbreidingen van de code toegevoegd.

Vrijdag 5 Oktober – Maandag 29 Oktober  
Stage event bijgewoond, Les op school bijgewoond, eerste versie stageverslag gemaakt, en een hele boel onderzoek gedaan naar bekende route problemen in de grafentheorie, Voornamelijk gekeken naar Minimum spanning tree en Steiner tree. Na een tijd zoeken naar wat er allemaal beschikbaar is en wat alle beschikbare mogelijkheden allemaal kunnen ben ik tot de conclusie gekomen dat dit niet het juiste pad is, en heb besloten een ander pad in te slaan.

Dinsdag 30 November – Dinsdag 20 November  
Het nieuwe concept is om bestaande velden te gebruiken als voorbeeld voor de nieuwe velden, dat concept heb ik uitgewerkt met de informatie die beschikbaar was, en die informatie was een set van outdated lay-outs. Ondanks het gebrek aan documentatie van voorgangers die velden hebben gemaakt heb ik het werkend gekregen, maar er blijven een paar onvermijdelijke foutmeldingen achter die ik denk te kunnen oplossen door correcte up-to-date informatie te verzamelen.

Woensdag 21 November – Dinsdag 11 December  
up-to-date informatie verzameld, bestaande code geperfectioneerd en algemene uitvoer van het nieuwe systeem uitbundig getest.

Woensdag 12 December – Donderdag 20 December  
Afronden, code “PEP8 approved” maken, het project omzetten naar een Python package, code en package OS-independent maken, testcases schrijven.

13. Bewijs verbetering

