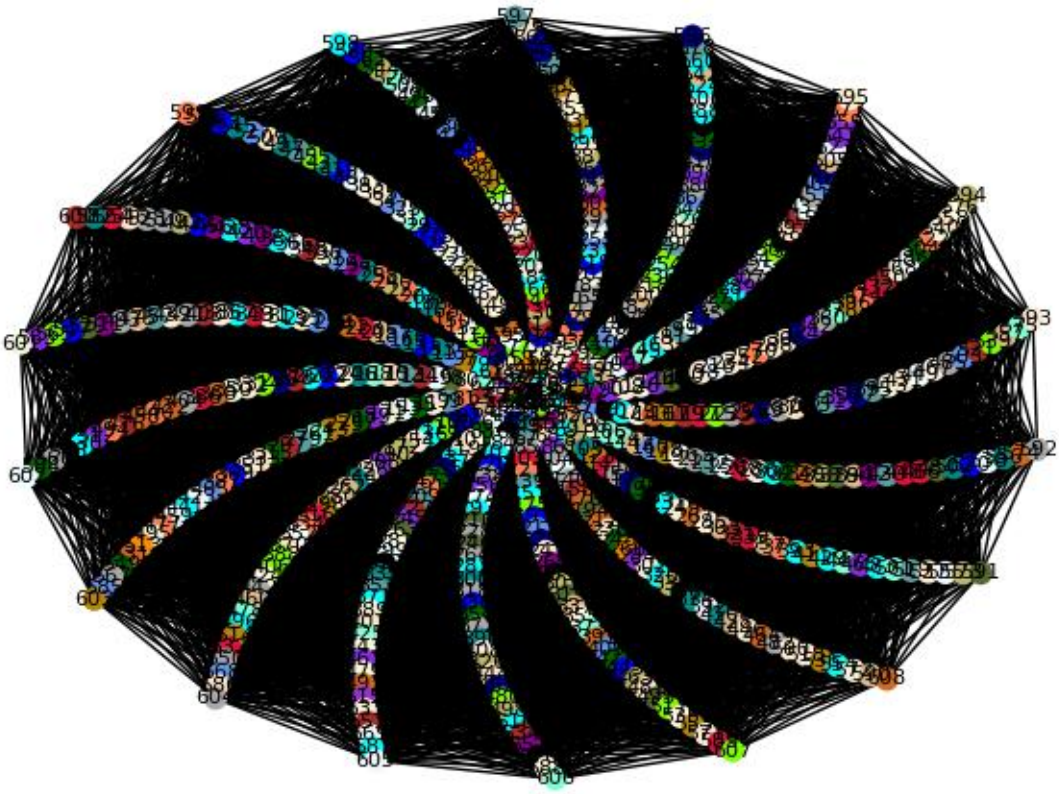
Profielwerkstuk roosteralgoritmiek

Door: Joep van Dijk & Sam Staijen



Begeleider: T. Nooijen

21 december 2021

Abstract

Roosteralgoritmiek is niet meer weg te denken uit het hedendaagse middelbare schoolrooster. Roosters zijn te complex geworden om met de hand te genereren. Met het profielwerkstuk roosteralgoritmiek is onderzocht hoe een middelbare schoolrooster ontworpen kan worden vanuit verschillende wiskundige concepten. Om een schoolrooster te ontwerpen, zijn lineair programmeren, branch-and-bound, en grafentheorie onderzocht, alvorens hiermee geëxperimenteerd werd. Hieruit bleek dat lineair programmeren de beste kandidaat was voor het maken van een roosteralgoritme. De resultaten hiervan staan in dit verslag beschreven.

Inhoudsopgave

Hoofdstuk 1

Inleiding

Elke middelbare scholier krijgt er wel eens mee te maken, een onhandig rooster. Het lijkt alsof de roostermaker altijd precies jou moet hebben. Ook wij hebben hier al vijf jaar last van. Een jaar geleden kwamen wij tot de conclusie dat hier wiskundige optimalisatie op toegepast zou moeten kunnen worden. Echter hadden wij de kennis en tijd niet om dit uit te zoeken. Het profielwerkstuk was het uitgesproken moment om dit te realiseren, en hopelijk een bijdrage te leveren aan de kwaliteit van roosters, ofwel om te onderzoeken hoe het ontwerp van een rooster precies werkt.

1.1 Doel

Het doel van dit profielwerkstuk is het onderzoeken hoe roosteroptimalisatie werkt, en hoe dit toegepast kan worden voor een middelbare school.

1.2 Onderzoeksvragen

1.2.1 Hoofdvraag

Is het mogelijk een programma te schrijven welke in een realistische tijdsspanne een functioneel rooster voor een middelbare school van 1000 leerlingen kan genereren?

1.2.2 Deelvragen

Deze onderzoeksvraag zal door een aantal deelvragen beantwoord worden:

- Hoe wordt een rooster nu ontworpen?
- Van welke factoren en voorkeur hangt het maken van een rooster af?
- Hoe worden soortgelijke problemen aangepakt?
- Welk van de afhankelijke voorkeuren vindt een middelbare scholier zwaarder wegen?
- Op welke manier kan het roosteroptimalisatieprobleem aangepakt worden?

1.3 Opbouw verslag

Eerst zal een groot deel van de theoretische achtergrond behandeld worden. Hierin komen onderwerpen als de huidige gang van zaken bij het ontwerpen van roosters, 'kleinste-kwadraten-methode', 'grafentheorie' en 'lineair-programmeren'.

Vervolgens zal het toegepaste onderdeel van dit verslag plaatsvinden. Hierin zal een wiskundig model gebouwd worden, en zal deze in de praktijk getest worden met behulp van de programmeertaal Python en de vakkenpakketten van de leerlingen op het Openbaar Lyceum in Zeist. Dit model zal menig maal geïtereerd worden, met elke generatie een kleine aanpassing aan de parameters, om een zo optimaal mogelijk rooster te kunnen genereren.

Tot slot zullen de resultaten uit het model geanalyseerd worden, en zal antwoord op de hoofdvraag gegeven worden. Achterin het verslag hebben we appendices die te raadplegen zijn, hier is een begrippenlijst te vinden en zijn onze broncodes en algoritmes te vinden.

Hoofdstuk 2

Werkwijze

2.1 Indexeren probleem, en afhankelijke variabelen soortelijke problemen

In het theoretische onderzoek dienen de theoretische deelvragen beantwoord te worden. Door deze deelvragen te beantwoorden, zal duidelijk worden hoe soortelijke problemen opgelost worden, welke afhankelijke variabelen deze hebben, en welke problemen bij het oplossen van dit probleem bleven bestaan. Door het vergelijken van verschillende soortgelijke problemen, wordt een duidelijk overzicht van mogelijke oplossingen in dit wiskundige veld verkend en onderzocht.

2.2 Vergaren informatie over eigen problemen, en haar variabelen

Gedurende, en na het vergaren van de theorie over soortgelijke problemen, dienen de afhankelijke variabelen van ons eigen onderzoek vergeleken te worden met die van soortgelijke probleen, en dient er voor elke afhanelijke variabele een oplossing bedacht te worden.

2.3 Ontwerpen algoritme

Na het doen van het theoretische onderzoek, dient de theorie concreet toegepast te worden op de onderzoeksvraag van dit profielwerkstuk. Alle deelvragen zijn reeds beantwoord (de theorie is immers af, en alle deelvragen zijn theoretisch). Er wordt duidelijk verslag gedaan van de formules, en de code. De formules en code dienen uitgelegt te worden in het verslag. Alle afhankelijke variabelen worden verwerkt in het algoritme, ofwel worden toegelicht waarom deze niet gebruikt zijn.

Hoofdstuk 3

Theorie

3.1 Verschillende invalshoeken optimaal rooster

Het ontwerp van een schoolrooster bestaat niet uit één beste optie. Er zijn een hoop verschillende invalshoeken voor het ontwerp van een rooster. Zo heeft een leerling eisen, maar ook de docent kan eisen hebben. De overheid kan normen stellen aan vakken en daardoor roosters, een onderwijskundige kan zijn opinie geven, enzovoorts (Dijk, Staijen en Finkelnberg, 2021). Hieronder staan enkele invalshoeken en afhankelijke factoren beschreven.

3.1.1 Invalshoek - Leerling

Een leerling kan het fijn vinden om vroeg uit te zijn, en dus minder tussenuren te hebben (het aantal lesuren zal niet zomaar minderen, aangezien deze vastgesteld staan).
Ook kan een leerling het fijn vinden om variatie in type lessen te hebben, om zo het hoofd 'fris' te houden (niet teveel bétavakken achter elkaar, niet teveel alfavakken achter elkaar, et cetera).
Tot slot kan het voor een leerling bervordelijk zijn om niet elke les van een vak achter elkaar te hebben, maar deze te spreiden over de week, zodat de stof beter onthouden wordt (Dijk, Staijen en Pomstra, 2021).

3.1.2 Invalshoek - Docent

Een docent heeft meerdere afwegingsfactoren die een grote rol kunnen spelen in het ontwerp van een schoolrooster. Zo is in de CAO vastgesteld dat met een 0.6 baan, een docent maar 3 van de 5 mogelijke dagen kan werken. Dan kan het nog steeds zo zijn dat een docent op 5 dagen beschikbaar is, maar dan mogen maar 3 van die 5 dagen benut worden. (Staijen en Pluut, 2021)
Tevens bleek uit een gesprek met enkele docenten (Staijen en Pluut, 2021), dat het voor docenten allicht juist bevorderlijk zijn tussenuren in te lassen. Docenten kunnen in deze tijd aan hun eigen ontwikkeling, lesvoorbereiding, nakijkwerk of andere bezigheid werken.
Tot slot kan het zo zijn dat docenten uren op vaste momenten vrij willen hebben, voor sectieoverleg ofwel schoolbreed overweg.

3.1.3 Invalshoek - Roostermaker

Hoewel in theorie een roostermaker natuurlijk zo min mogelijk leerlinguren wil gebruiken om zoveel mogelijk in te roosteren, kan het zijn dat een roostermaker bewust een tussenuur laat staan of toevoegt.

3.1.4 Invalshoek - Onderwijskundige

Voor een onderwijskundige kan het van belang zijn om vakken met de stamklas zo veel mogelijk te spreiden over de week, om zo het aantal contactmomenten te vergroten.
Ook wil een onderwijskundige (zoals ook vastgesteld staat in de CAO), dat een docent bevoegd is om een klas les te geven, om zo de vereiste kwaliteit te kunnen leveren. (Staijen en Pluut, 2021)

3.1.5 Aanname profielwerkstuk

Bij het maken van het model van dit profielwerkstuk zal alleen gekeken worden vanuit de invalshoek van de leerling. Het optimale vanuit meerdere, ofwel alle invalshoeken wordt vooralsnog als te complex ingeschat.

3.2 Huidig ontwerp rooster

Aan de roostermakers van het Openbaar Lyceum Zeist is een aantal vragen gesteld om erachter te komen hoe momenteel de rooster gemaakt worden (Dijk, Staijen en Finkelnberg, 2021).
Bij het ontwerp van een rooster horen de volgende drie stappen:

1. Vergaren van informatie
2. Computer gegenereerd rooster
3. Handmatig aangepast rooster

3.2.1 Stap 1 - vergaren van informatie

De eerste stap bij het maken van een rooster is het vergaren van de benodigde informatie. Bijvoorbeeld: hoeveel leerlingen zijn er? Welke vakken hebben die leerlingen?
Tevens wordt er gedurende het voorgaande schooljaar al een analyse op de prestaties van alle leerlingen losgelaten, om een grove schatting te kunnen maken van het aantal leerlingen dat in een jaarlaag, en niveau terecht zal komen. Verder worden beschikbaarheden van docenten en pakketkeuzes van leerlingen vergaard.
Echter zal deze stap voor ons onderzoek niet van toepassing zijn, omdat ons onderzoek gaat om het ontwerpen van een schoolrooster (informatieverwerking), en niet het vergaren van de informatie (informatievoorziening).

3.2.2 Stap 2 - computer gegenereerd rooster

De roostermakers spreken over twee roosters, het eerste rooster is een rooster dat compleet automatisch gemaakt is door het roosterprogramma van Zermelo[1]. Hier wordt er voor gezorgd dat er een 'correct' rooster is, zonder bijvoorbeeld dubbele uren voor docenten of leerlingen. Hierbij wordt al wel rekening gehouden met voorkeuren voor bijvoorbeeld leerlingen en docenten door middel van een strafpuntensysteem (Dijk, Staijen en Finkelnberg, 2021).

3.2.3 Stap 3 - handmatig aangepast rooster

Het tweede rooster bouwt voort op het eerste rooster en wordt handmatig gemaakt.
Bij het tweede rooster wordt rekening gehouden met persoonlijke voorkeur op basis van de mening van de roostermaker *(dit kan per roostermaker en per school verschillen, op het Openbaar Lyceum Zeist wordt dit wel gedaan).*
Bij deze stap kijkt de roostermaker of bepaalde dingen uit het rooster verbetert kunnen worden. Dit gebeurt handmatig en duurt lang, hierin kan de roostermaker zelf bedenken wat het beste voor de leerlingen en docenten is, maar moet elk gevolg van een wijziging uitzoeken en vervolgens de afweging maken deze wel of niet door te voeren. Deze wijzigingen zullen per school verschillen.

3.3 Afhankelijke factoren en voorkeuren rooster

3.3.1 Afhankelijke factoren rooster

Er zijn een aantal factoren waar rekening mee gehouden moet worden bij het ontwerpen van een rooster:

- Docent
  - Beschikbaarheid (welke uren kan een docent werken)
  - Vak (welke vakken is een docent bevoegd te geven)
  - werklast (hoeveel uren kan een docent werken?)
  - Bevoegdheid bovenbouw (mag een docent lesgeven aan de bovenbouw?)
- Leerling
  - Vakken (welke vakken heeft een leerling)
  - Clustergroepen (in welke clustergroepen zit een leerling)
- Clustergroep
  - welk vak krijgt de clustergroep?
  - welke leerlingen zitten in de clustergroep?

[1]Zermelo: www.zermelo.nl

- Vak
  - Vakken die gegeven worden op de school
  - Docenten die de vakgroep les mogen geven
  - Clustergroepen die het desbetreffende vak krijgen
- Tijdssloten
  - Aantal tijdssloten op een dag
  - Aantal dagen in een periode (week)
- Lokalen
  - Aantal lokalen
  - Verzameling lokaaltypen

(Gunawan e.a., 2006, Dijk, Staijen en Pomstra, 2021)

3.3.2 Afhankelijke voorkeuren rooster

Voor het handmatig aanpassen van het rooster, zoals bij 'rooster twee' wordt gedaan, worden verschillende afhankelijke voorkeuren gebruikt. De eerste, en primaire voorkeursvariabele is het totaal aantal tussenuren in de gehele school (Dijk, Staijen en Finkelnberg, 2021).

3.4 Weging verschillende voorkeuren

Hoewel de afhankelijke variabelen verplicht zijn om aan te voldoen indien een functioneel rooster gemaakt moet worden, zijn de voorkeuren alleen nodig voor het fijner maken van het rooster. Om deze reden dient een rooster eerst ontworpen te worden op basis van de afhankelijke variabelen, en later pas aangepast worden op basis van de voorkeuren, mits dit conform de variabelen kan.

3.5 Kleinste-kwadratenmethode

De kleinste-kwadratenmethode is een methode om uit een verzameling punten de best passende curve te bepalen. De best passende curve is de curve waarvan de som van de kwadraat van de verticale afstand van elk punt tot de curve het kleinste is.
De regressielijn[2] kan worden opgesteld door door de a (richtingscoëfficiënt) en b (verschuiving) in de standaardformule voor een lineaire lijn te berekenen (formule 3.1).

$$y = a \cdot x + b \tag{3.1}$$

[2]Een regressielijn is een lijn die zo dicht mogelijk bij alle punten op een grafiek ligt.

De richtingscoëfficiënt a kan berekent worden door de covariantie van x en y (formule 3.3) te delen door de variantie van x (formule 3.4).

$$rc_{regressielijn} = a = \frac{\text{cov}(x, y)}{var(x)} \tag{3.2}$$

$$\text{cov}(x, y) = \frac{\sum(x_i - \overline{x}) \cdot (y_i - \overline{y})}{n} \tag{3.3}$$

$$\text{var}(x) = \frac{\sum(x_i - \overline{x})^2}{n} \tag{3.4}$$

Bij beiden bovenstaande formules is i het i-de punt in de lijst van punten met lengte n, en n is het totaal aantal punten in de lijst. De hoofdletter sigma is een sommatie-teken en een variabel met een streep erboven betekent het gemiddelde van die variabel, zoals beschreven in de functielijst (appendix A). Substitutie en vereenvoudiging van formule 3.3 en formule 3.4 in formule 3.2 geeft:

$$a = \frac{\sum(x_i - \overline{x}) \cdot (y_i - \overline{y})}{\sum(x_i - \overline{x})^2} \tag{3.5}$$

Vervolgens kan de afwijking b berekend worden met formule 3.6:

$$b = \overline{y} - a \cdot \overline{x} \tag{3.6}$$

Wanneer a en b terug gesubstitueerd worden in formule 3.2, is de formule voor de trendlijn met de kleinste som van kwadraten opgesteld (Bakker e.a., 2017).

3.6 Lineair programmeren

Lineair programmeren is te gebruiken als er een (lineaire) doelfunctie opgesteld kan worden en als er beperkende voorwaarden opgesteld kunnen worden als lineaire functies. De doelfunctie moet gemaximaliseerd of geminimaliseerd waarbij voldaan wordt aan de beperkende voorwaarden.
Bijvoorbeeld:
Maximaliseer

$$z = 3 \cdot x_1 + 2 \cdot x_2 \tag{3.7}$$

Met respect tot

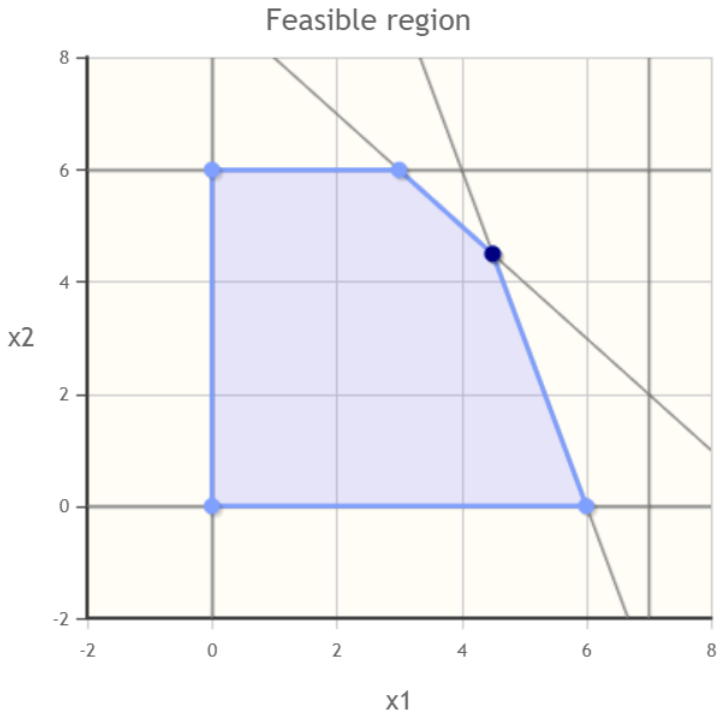
$$\begin{aligned} x_1 &\geq 0 \\ x_2 &\geq 0 \\ x_1 + x_2 &\leq 9 \\ 3 \cdot x_1 + x_2 &\leq 18 \\ x_1 &\leq 7 \\ x_2 &\leq 6 \end{aligned} \tag{3.8}$$

In deze lineaire programmering worden twee variabelen gebruikt, x_1 en x_2. Deze variabelen moeten aan een aantal voorwaarden voldoen. De variabelen moeten bijvoorbeeld groter of gelijk aan 0 zijn, bij elkaar opgeteld mogen ze niet meer dan 9 zijn, enzovoorts.

Alle beperkende voorwaarden zorgen voor een vlak dat het toegestane gebied heet, dit is het blauwe vak in figuur 3.1.

Dit resulteert in de optimale waarden in formule 3.9 en figuur 3.1.

$$\begin{aligned} x_1 &= 4.5 \\ x_2 &= 4.5 \end{aligned} \tag{3.9}$$



Figuur 3.1: Voorbeeld lineair programmeren, met doelfunctie formule 3.7 met de randvoorwaarden van formule 3.8. De donkerblauwe punt is het optimale punt voor de combinatie van x_1 en x_2.

Substitutie van de optimale waarden in de functie die gemaximaliseerd moet worden geeft:

$$3 \cdot 4.5 + 2 \cdot 4.5 = 22.5 \tag{3.10}$$

De maximale waarde voor deze lineare programmering is dus 22.5.

3.6.1 Het simplexalgoritme

Een LP-probleem kan met verschillende algoritmes opgelost worden. Het simplexalgoritme is het meest gebruikte algoritme om LP-problemen op te lossen en is ook het algoritme dat wij gebruiken. Het simplexalgoritme is ontworpen door de Amerikaanse wiskundige George B. Dantzig in 1947, dit is het begin van het lineaire programmeren zoals we het vandaag de dag kennen (Katwijk, 2013).

Het algoritme gaat op zoek naar de punten waar de lijnen van de voorwaarden elkaar snijden. Als het een ILP (Integer Linear Programming) is en geen LP (Linear Programming) gaat het op zoek naar het optimale punt waar de lijnen van de voorwaarden elkaar kijken, en zoekt het in de omgeving naar een punt dat én voldoet aan de voorwaarden én de hoogste waarde van de doelfunctie genereert (Bazett, 2021 en with Mr Orys, 2019).

Voorbeeld - het timmermanprobleem

Als voorbeeld zal het timmermanprobleem gebruikt worden. Een timmerman kan tafels en boekenkasten maken. Het maken van een tafel kost 10 eenheden hout en 5 uur aan werktijd, maar de timmerman verdient per tafel €180,-. Een boekenkast kost 20 eenheden hout en 4 uur aan werktijd maar levert de timmerman €200,- op. De timmerman heeft 200 eenheden hout beschikbaar en 80 uur aan werktijd. Een oplossing van dit probleem in Python staat in appendix C.

Als we het aantal gemaakt tafels x noemen en het aantal gemaakte boekenkasten y, krijg je de doelfunctie zoals beschreven in formule 3.11.

$$\begin{aligned} f(x,y) &= 180 \cdot x + 200 \cdot y \\ \text{Maximaliseer: } & f(x,y) \end{aligned} \tag{3.11}$$

Dit probleem heeft een aantal voorwaarden, er is namelijk maar een beperkte hoeveelheid hout en tijd beschikbaar voor de timmerman, de eerste voorwaarde (formule 3.12) zorgt ervoor dat de timmerman niet meer dan 80 uur werkt. De tweede voorwaarde (formule 3.13) zorgt ervoor dat de timmerman niet teveel hout gebruikt, niet meer dan 200 eenheden hout. Vervolgens mag er geen negatief aantal tafels of boekenkasten gemaakt worden, die voorwaarden zijn beschreven in formule 3.14 en 3.15. De laatste twee voorwaarden zorgen ervoor dat voor x en y alleen gehele getallen gebruikt mogen worden, zoals te zien is in formules 3.16 en 3.17.

$$5 \cdot x + 4 \cdot y \leq 80 \tag{3.12}$$

$$10 \cdot x + 20 \cdot y \leq 200 \tag{3.13}$$

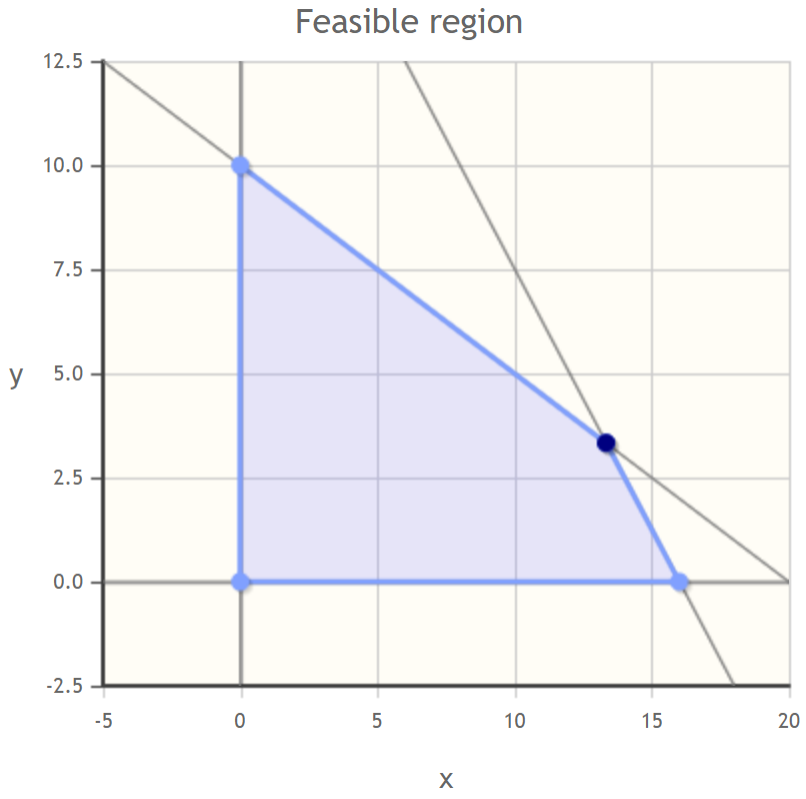
$$x \geq 0 \tag{3.14}$$

$$y \geq 0 \tag{3.15}$$

$$x \in \mathbb{Z} \tag{3.16}$$

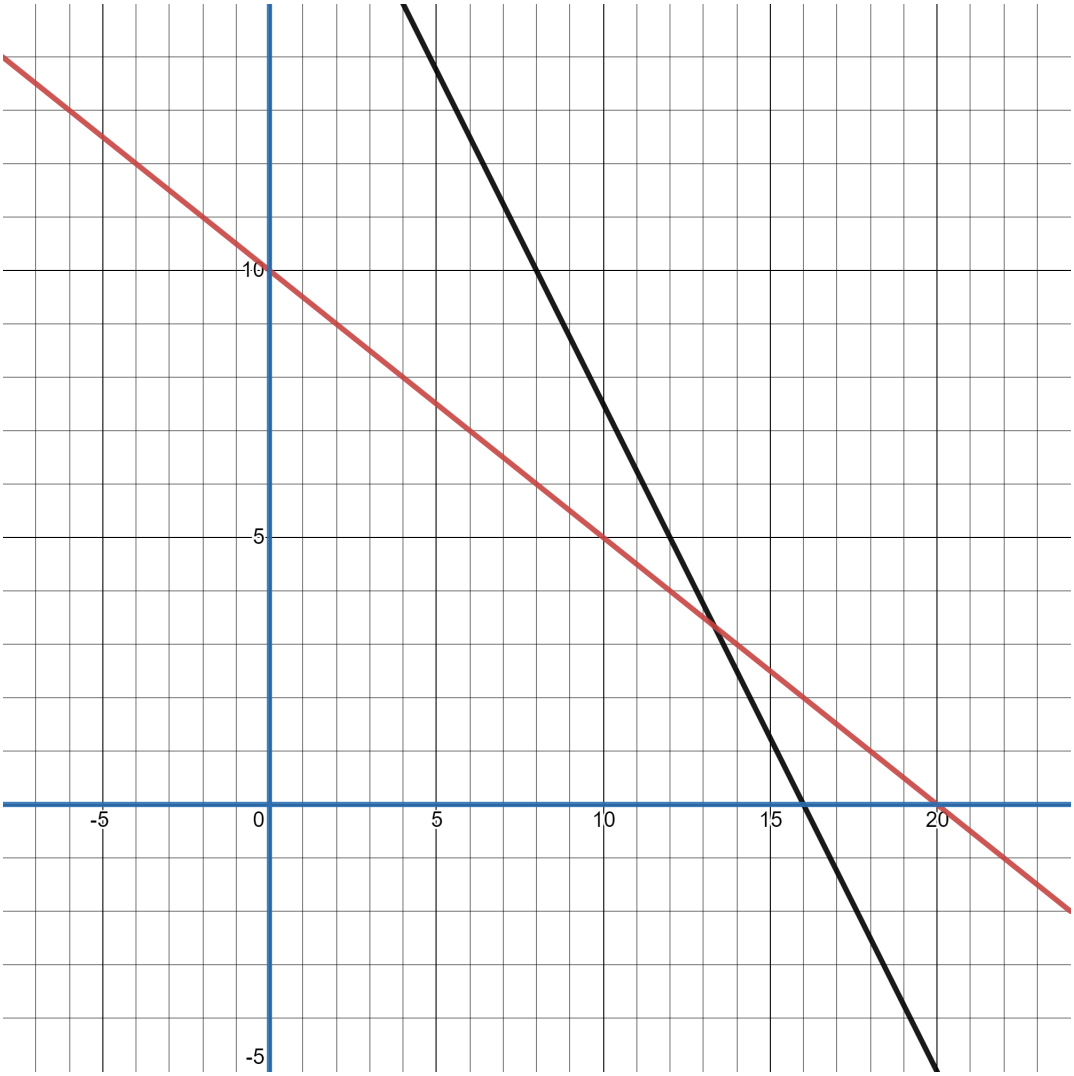
$$y \in \mathbb{Z} \tag{3.17}$$

Dit resulteert in een "mogelijk gebied"zoals te zien is in figuur 3.2. Ieder punt in het blauwe gebied voldoet aan de beperkingen, er is echter maar één punt dat tot een optimale waarde van de doelfunctie zal leiden.



Figuur 3.2: Toepassing simplexmethode op het timmermanprobleem

De lijnen in figuur 3.2 geven de beperkingen aan. De maximale waarde zal altijd op een punt zitten waar de lijnen van de beperkingen elkaar kruizen omdat je dan maximaal gebruikmaakt van de materialen die je tot je beschikking hebt.



Figuur 3.3: Beperkingen van het timmermanprobleem

In formule 3.18 is te zien wat het resultaat is.

$$f(13\frac{1}{3}, 3\frac{1}{3}) = 180 \cdot 13\frac{1}{3} + 200 \cdot 3\frac{1}{3} = 3066\frac{2}{3} \tag{3.18}$$

In figuur 3.4 zijn alle punten te zien waarbij lijnen van voorwaarden elkaar kruizen. Een aantal waarden kunnen we uitsluiten, zoals $(0, 0)$ omdat dit zal resulteren in 0, maar we willen de doelfunctie juist maximaliseren.

Punt $(0, 20)$ ligt buiten het "toegestane gebied", evenals punt $(20, 0)$. Dan moeten dus de volgende punten bekeken worden: $(0, 10)$, $(16, 0)$ en $(13\frac{1}{3}, 3\frac{1}{3})$. Hoewel dat laatste punt niet mag omdat het niet bestaat uit gehele getallen kan het wel zijn dat de optimale waarde in de buurt van dat punt licht.

Als $(0, 10)$ ingevuld wordt in de doelfunctie krijgen we $f(0, 10) = 180 \cdot 0 + 200 \cdot 10 = 1000$.

Als $(16, 0)$ ingevuld wordt in de doelfunctie krijgen we $f(16, 0) = 180 \cdot 16 + 200 \cdot 0 = 2880$, dit is hoger dan het vorige punt dus alleen dit punt en het laatste punt moeten vergeleken worden.

Als $(13\frac{1}{3}, 3\frac{1}{3})$ ingevuld wordt in de doelfunctie krijgen we $f(13\frac{1}{3}, 3\frac{1}{3}) = 180 \cdot 13\frac{1}{3} + 200 \cdot 3\frac{1}{3} = 3066\frac{2}{3}$. Dit is de hoogste waarde, dus dit is het optimale punt. Omdat dit punt fracties bevat zal in de buurt gezocht moeten worden naar het optimale punt dat wel aan de voorwaarde voldoet dat alleen gehele getallen gebruikt mogen worden.

In tabel 3.1 staat dit uitgewerkt. We ronden de getallen die we hebben gekregen af, dus $x = 13$ en $y = 3$, en we zoeken in de omgeving van die getallen. Hoe meer getallen je bezoekt hoe zekerder je uitkomst zal zijn, maar ook hoe langer het berekenen duurt (with Mr Orys, 2019). Uiteindelijk komen we uit als optimale waarden $x = 12$ en $y = 4$, ofwel twaalf tafels en vier boekenkasten. Dit zorgt voor een opbrengst van €2960,-.

Var.	**Var.**	**Voorwaarde**	**Voorwaarde**	**Doelfunctie**
x	y	$5 \cdot x + 4 \cdot y \leq 80$	$10 \cdot x + 20 \cdot y \leq 200$	$180 \cdot x + 200 \cdot y$
13	3	77	190	2940
13	4	81	210	2140
13	2	73	170	2740
14	3	82	200	3120
14	4	86	220	3320
14	2	78	180	2920
12	3	72	180	2760
12	2	68	160	2560
12	4	76	200	2960

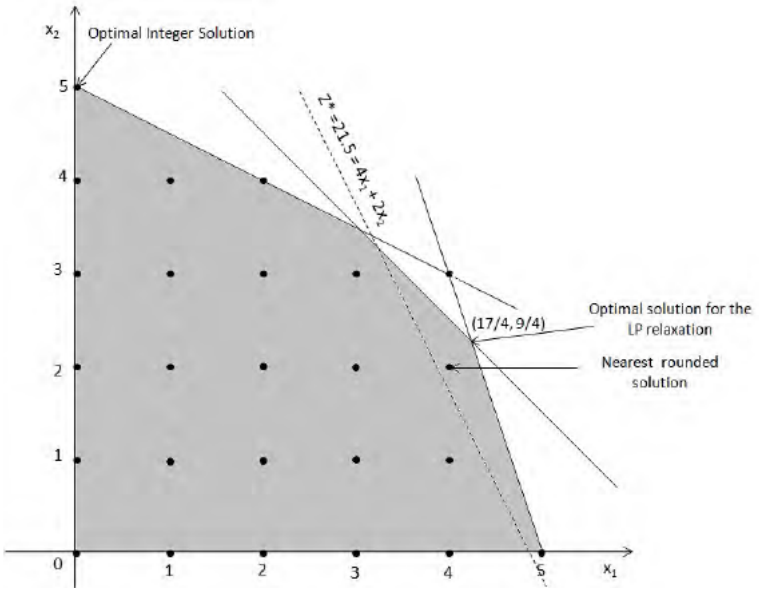
Tabel 3.1: Uitwerking van het simplex algoritme, rijen met één of meerdere rode cellen voldoen niet aan de voorwaarden

Figuur 3.4: Alle snij-/raakpunten (oranje)

3.7 Branch and bound

Branch and bound (vanaf nu naar verwezen als B&B) is een systematische optimalisatiestrategie voor lineair programmeren waarbij het mogelijkheidsvlak bij elke branch (vertakking) verdeeld wordt in verschillende stukjes.
B&B is alleen nuttig wanneer er een restrictie ligt op de waarde van de variabelen. Zo kan het zijn dat (een deel van) de variabelen gehele getallen (integers) moeten zijn (MILP of ILP), of dat (een deel van) de variabelen binair moeten zijn (waarde 1 of 0). Het hebben van deze resticties, zorgt ervoor dat niet elke plek in het mogelijke vlak een oplossing kan zijn, en dat deze dus niet per se op een van de grenslijnen hoeft te liggen. Zie figuur 3.5 voor een grafische weergave van het verschil tussen het oplossingenvlak en de mogelijke oplossingen.

B&B bomen bestaan uit twee onderdelen; nodes en branches. Nodes zijn lineaire programmeringen, met een boven- en ondergrens (word verderop uitgelegd). Nodes worden vaak weergegeven als bolletjes, of vakjes. Een branch is een extra restrictie die op basis van de



Figuur 3.5: Integers hoeven niet altijd op de optimale plek te liggen

uitkomst van de LP erboven wordt opgelegd aan de node eronder. Branches worden vaak weergegeven als vertakkingen (zie figuur 3.7).

Waar een conventionele systematische optimalisatiestrategie elke oplossing langsgaat (ook wel brute-force genoemd), kan B&B branches vormen op de meest kansgevende node, en het gewenste oplossingsveld steeds iets verder verkleinen tot een optimale oplossing is gevonden (Casquilho, g.d.). Hierdoor kunnen vele mogelijkheden worden uitgesloten, waardoor de rekenkost verlaagd kan worden. In figuur 3.6 is zichtbaar hoe extra restricties, die besloten worden in de B&B boom, het mogelijkheidsveld kunnen verkleinen. Het blauwe vlak is het nieuwe mogelijkheidsveld nadat de extra resticties $x_1 \geq 2$ en $x_2 \geq 1$. Het principe achter B&B is dat dit verkleinen bij elke node gebeurt, waardoor een kleiner zoekvlak gemaakt kan worden tot er maar 1 oplossing over blijft.

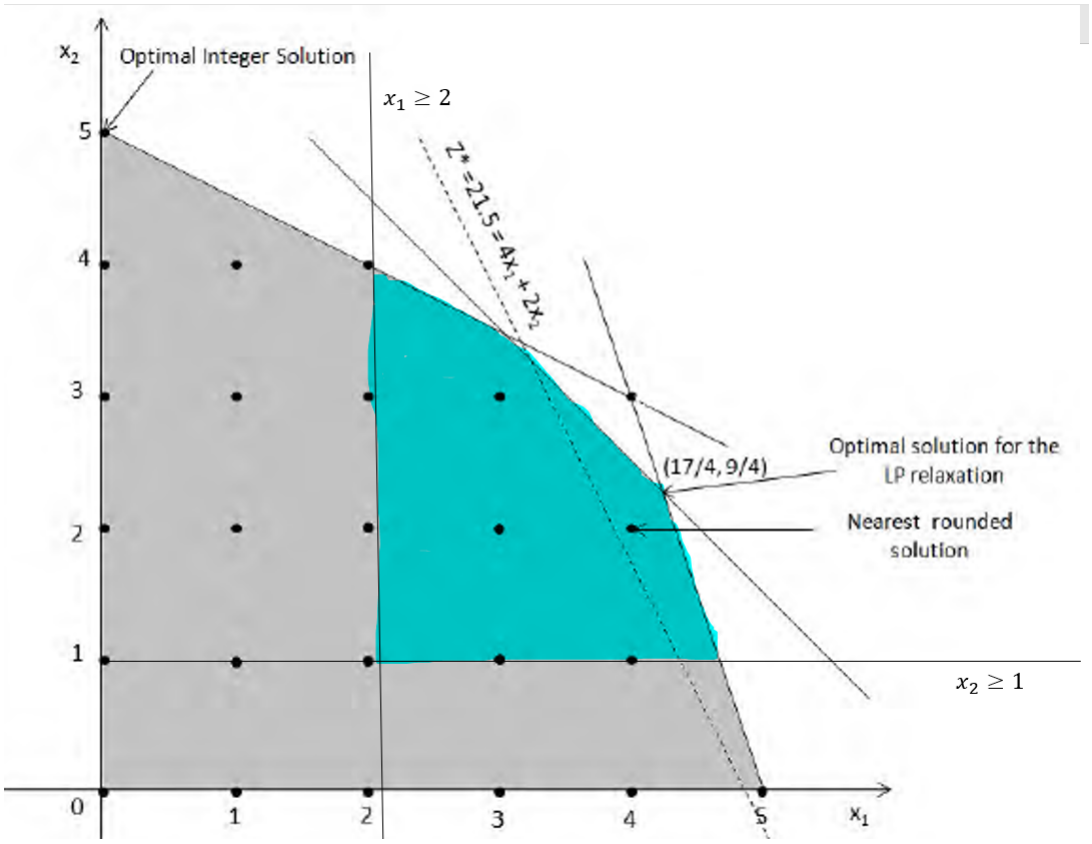
3.7.1 Branch and Bound - aanpak

Stappen

Het oplossen van een B&B kan opgebroken worden in een aantal stappen. Hieronder zullen de 6 stappen uitgelegd worden, waarna een voorbeeld gegeven zal worden. Er wordt voor de uitleg uitgegaan van een ILP; aan het eind worden de aanpassingen voor een MILP of of 0-1-integer lineaire programmering (ook wel binaire lineaire programmering) toegelicht.

Stap 1 - LP-relaxatie

De eerste stap van B&B is het relaxeren van de LP. Dit wordt ook wel LP-relaxatie genoemd. Men gaat er tijdelijk vanuit dat ook niet-gehele-getallen als antwoord kunnen worden geno-



Figuur 3.6: ILP met extra restricties, en een nieuw mogelijkheidsveld

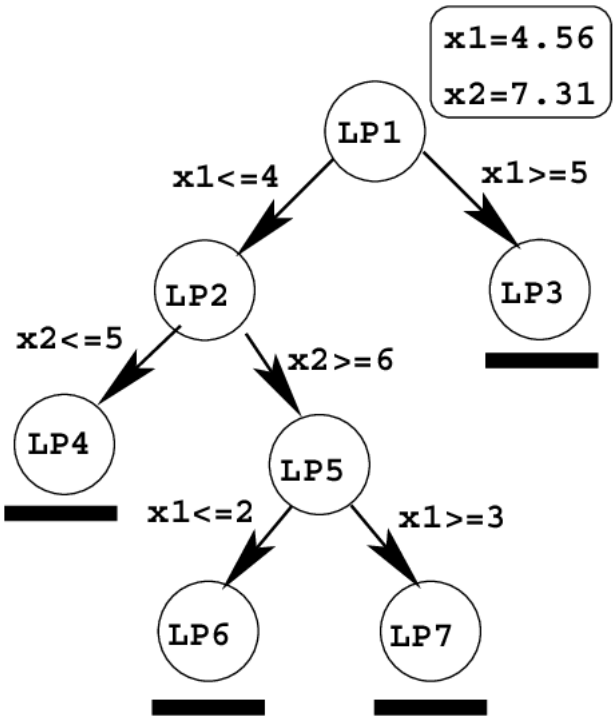
men, en dat formules dus exact opgelost kunnen worden.

Stap 2 - optimale LP-gerelaxeerde oplossing

Vervolgens kan de optimale oplossing voor de LP-gerelaxeerde functie gevonden worden door het algebraïsch oplossen van de functie. De beste oplossing voor de niet-gerelaxeerde LP kan nooit hoger zijn dan de optimale oplossing (de optimale oplossing ligt immers in het mogelijkheidsvlak van de lijnen zoals in figuur 3.5), dus wordt er gesproken over deze oplossing als zijnde een bovengrens. Wanneer de waardes voor de variabelen naar beneden worden afgerond, kan de ondergrens besloten worden. Er is immers bekend dat dit een mogelijke oplossing is, maar idealiter zou een betere oplossing gevonden worden (Casquilho, g.d.).
Er is dus een kleiner zoekgebied gemaakt, door het toevoegen van een boven- en ondergrens.

Stap 3 - branchen op grootste fractie

Hierna kan een eerste branch gemaakt worden. De branch die gemaakt wordt is de vertakking naar het gehele getal boven en onder grootste fractie (2.55 is een grotere fractie dan 5.14). Je vertakt op de grootste fractie, omdat je streeft naar twee gehele getallen, en de grootste fractie hier dus het verst vanaf ligt. De grootste fractie kan gevonden worden met behulp van modulorekenen. Wanneer je de modulus neemt van een getal, trek je de modulusfactor van het getal af tot het niet meer geheel eraf kan. Wiskundig kan de grootste fractie gevonden



Figuur 3.7: Een voorbeeld van een B&B boom bij een ILP

worden met:

$$fracMax(x_i) = 0.25 - (0.5 - (x_i \cdot mod(1)))^2 \tag{3.19}$$

Wanneer de grootste fractie gevonden is, worden twee branches gevormd. De ene branch rond de waarde voor de variabele met de grootste fractie af naar beneden, en stelt dat de variabele kleiner dan of gelijk aan dat gehele getal moet zijn. De andere branch rond naar boven af stelt dat de variabele groter dan of gelijk aan dat gehele getal moet zijn. Bijvoorbeeld:

$$\begin{aligned} fracMax(x_1) = fracMax(5.44) = 0.25 - (0.5 - (5.44 \cdot mod(1)))^2 = 0.2464 \\ fracMax(x_2) = fracMax(5.89) = 0.25 - (0.5 - (5.89 \cdot mod(1)))^2 = 0.0979 \\ fracMax(x_1) > fracMax(x_2) \end{aligned} \tag{3.20}$$

Nu bekend is welke variabele de grootste fractie heeft in deze gerelaxeerde oplossing, kunnen de branches gevormd worden. De lagere branch voegt de randvoorwaarde toe met het lagere gehele getal, de hogere branch met het hogere gehele getal. De wiskundige symbolen zijn weergegeven in de appendices bijlage A

$$\begin{aligned} upperBranch &\longrightarrow x_i \geq \lfloor x_i \rfloor \\ lowerBranch &\longrightarrow x_i \leq \lfloor x_i \rfloor \end{aligned} \tag{3.21}$$

Stap 4 - nieuwe nodes maken en weer branchen

Stap 4 is het maken van nieuwe nodes. De LP van de erboven staande node kan overgenomen worden in deze node, waarbij de restrictie van de branch die de erboven staande node verbind met deze toegevoegd kan worden. Beide nodes zullen weer een eigen upper, en lower bound hebben. Deze is te vergaren door de gerelaxeerde LP op te lossen.

Stap 5 - itereren, of controleren

Stap 5 heeft 2 mogelijkheden; ofwel het herhalen van enkele stappen, of, wanneer een upper bound wordt gevonden die gelijk is aan de lower bound, het controleren of dit de maximale waarde is.

Optie 1: itereren
Indien er bij de node waar zojuist de boven- en ondergrens voor berekend zijn deze nog niet overeenkomen, worden stap 2 tot en met stap 4 herhaald tot dit wel het geval is.

Optie 2: controleren
Wanneer de boven- en ondergrens van een node overeenkomen, bestaat de kans dat dit een optimale oplossing is voor de (M)ILP in kwestie. Om dit te controleren, wordt de bovengrens vergeleken met de bovengrens van elke (op dat moment) eindnode[3]. Wanneer deze allemaal even groot of kleiner zijn dat de bovengrens van de node die je controleert, is een optimale oplossing gevonden. Wanneer een van de bovengrenzen groter is, wordt vanaf de node met een grotere bovengrens het proces vanaf stap 2 herhaald.

3.7.2 Voorbeeld

Neem als voorbeeld de ILP zoals beschreven in formule 3.22:

Maximaliseer:

$$z = 43x_1 + 122x_2 \tag{3.22}$$

Met respect tot:

$$\begin{aligned} 3x_1 + 2x_2 &\leq 500 \\ 235x_1 + 534x_2 &\leq 54328 \\ 150x_1 + x_2 &\leq 23456 \\ x_1 + 140x_2 &\leq 8375 \end{aligned} \tag{3.23}$$

$$\begin{aligned} x_1 &\in \mathbb{Z}^+ \\ x_2 &\in \mathbb{Z}^+ \end{aligned} \tag{3.24}$$

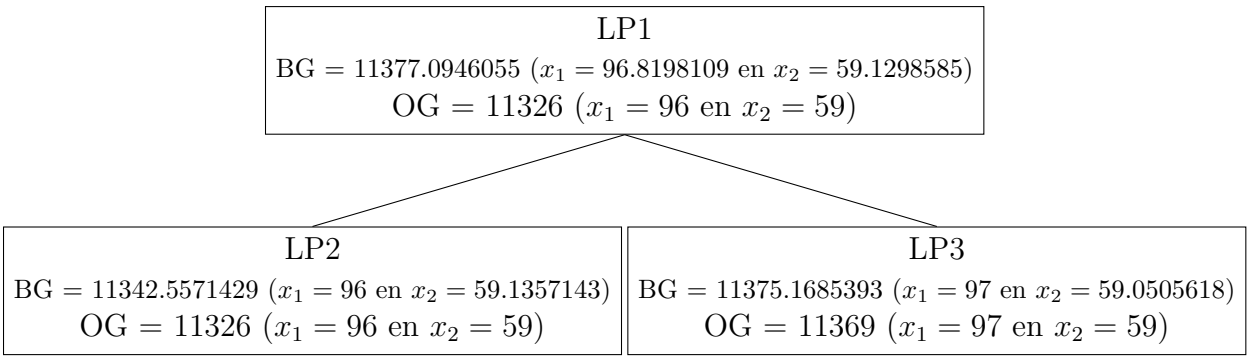
Allereerst kan hiervoor de eerste node voor ontworpen worden:

LP1
bovengrens = 11377.0946055 ($x_1 = 96.8198109$ en $x_2 = 59.1298585$)
ondergrens = 11326 ($x_1 = 96$ en $x_2 = 59$)

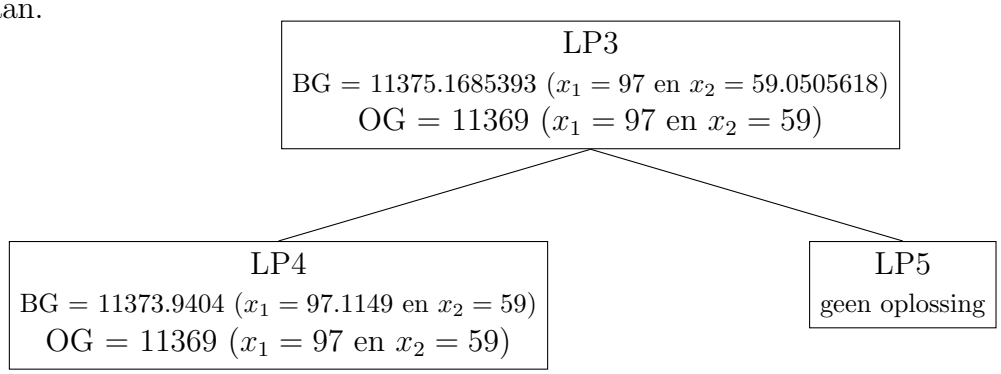
Omdat $fracMax(x_1) > fracMax(x_2)$, vind de afsplitsing van branches hier plaats op basis van x_1. De ene branch krijgt als randvoorwaarde erbij $x_1 \leq 96$, en de ander branch krijgt als randvoorwaarde erbij $x_1 \geq 97$.
Vervolgens worden de nodes hieronder berekend. Deze nodes hebben de randvoorwaarden

[3]Eindnode: node waar geen nodes meer onder zitten

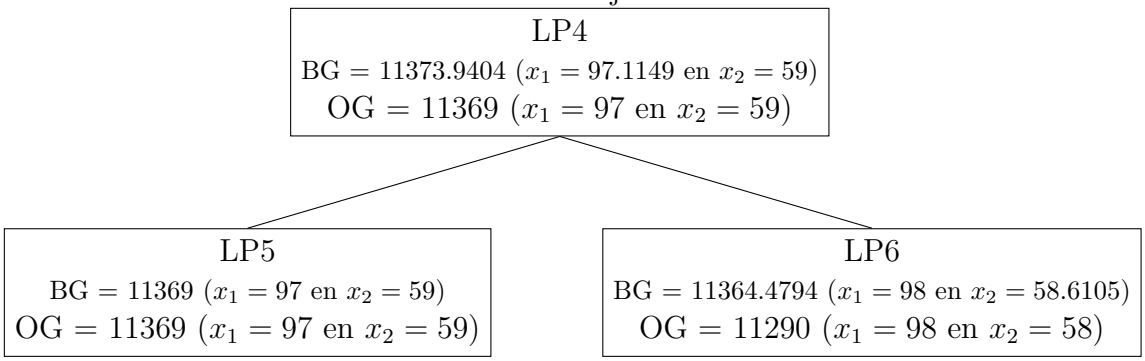
van LP1, met de randvoorwaarde van hun branch eraan toegevoegd.



Omdat de ondergrens bij LP3 hoger ligt dan bij die van LP2, wordt verder gewerkt op LP3. $fracMax(x_2) > fracMax(x_1)$, dus er wordt vertakt met een extra randvoorwaarde op x_2. De ene branch krijgt als randvoorwaarde erbij $x_2 \leq 59$, en de ander branch krijgt als randvoorwaarde erbij $x_2 \geq 60$. De randvoorwaarde van de branch ervoor ($x_1 \geq 97$) blijft staan.



LP5 heeft geen oplossing, dus wordt er verder gegaan op LP4. Omdat $fracMax(x_1) > fracMax(x_2)$, wordt er vertakt met een extra randvoorwaarde op x_1. De ene branch krijgt als randvoorwaarde erbij $x_1 \leq 97$ en de andere branch krijg als randvoorwaarde erbij $x_1 \geq 98$. De randvoorwaarden van de branches ervoor blijven ook voor deze node staan.



LP5 is de beste oplossing met x_1 en x_2 als integers.

3.7.3 Aanpassingen; Mixed integer lineaire programmering

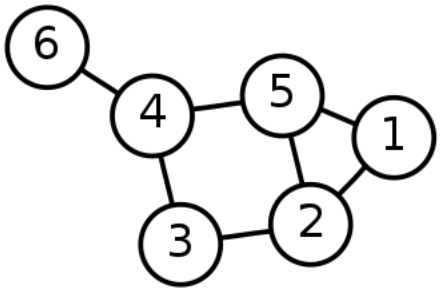
Voor mixed-integer programmering worden de niet-integer variabelen wel meegenomen, maar deze blijven exact (worden niet afgerond), en deze worden ook niet gebruikt voor het vertakken. Enkel de integer variabelen worden hiervoor gebruikt. Verder blijven de regels en stappen van branch and bound voor integer programmering van kracht.

3.7.4 Aanpassingen: 0-1-integer lineaire programmering

Voor 0-1 integer programmering wordt slechts een extra regel toegepast aan reguliere integer programmering. Voor elke integer variabele wordt de restrictie $x_i \in \{0, 1\}$ toegevoegd. Verder blijven de regels en stappen van branch and bound voor integer programmering van kracht.

3.8 Grafentheorie

De grafentheorie is een manier om verbindingen tussen dingen wiskundig weer te geven. Een graaf bestaat uit een verzameling knopen (punten), waarvan sommige verbonden kunnen zijn door zijden (lijnen).
Wanneer een zijde door de toepassing slechts één kant op gericht is (vaak weergegeven met een pijl), wordt er gesproken van een gerichte graaf.
Aan zijden kunnen gewichten gehangen worden. Het gewicht stelt dan bijvoorbeeld de afstand tussen de twee punten voor. Hierbij kun je een pathfinding[4] algoritme gebruiken om de korste weg te vinden. Wanneer er gewichten gebruikt worden heet de graaf ook wel een gewogen graaf (Flovik, 2021).
Een voorbeeld van een graaf staat in figuur 3.8.



Figuur 3.8: Een graaf met zes knopen

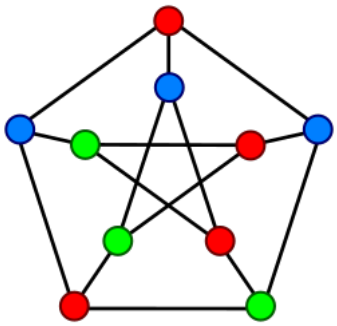
3.8.1 Kleuren van grafen

Een toepassing van een graaf is het kleuren ervan. Het doel hierbij is om met zo min mogelijk kleuren, alle buren[5] van een punt een andere kleur te geven. Een kleur kan natuurlijk ook

[4]Pathfinding: een algoritme om de kortste weg tussen twee of meerdere punten te vinden: https://en.wikipedia.org/wiki/Pathfinding

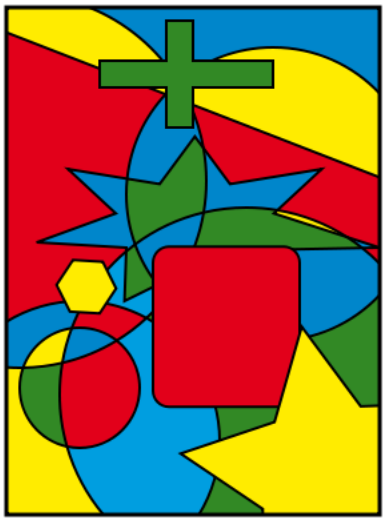
[5]Buren: alle punten waarmee een punt verbonden is via zijdes

een natuurlijk getal, of iets anders zijn; dit verandert niks in het gebruik. Echter wordt voor voorbeelden vaak voor een visuele scheiding van buren gekozen, omdat dit het onderscheid voor mensen makkelijk toont (Wikipedia-bijdragers, 2021). Een voorbeeld hiervan is te zien in figuur 3.9.



Figuur 3.9: Een ingekleurde graaf met 3 kleuren

De oorsprong van dit probleem komt voort uit het kleuren van kaarten. Het doel was om met zo min mogelijk kleuren geen enkel grenzend land dezelfde kleur te geven. Met de vierkleurenstelling is bewezen dat hiervoor maximaal vier kleuren nodig zijn (Wikipedia-bijdragers, 2020). Een voorbeeld hiervan is te zien in figuur 3.10.



Figuur 3.10: Een kaart ingekleurt met vier kleuren

Toepassing - roosteren

Het kleuren van grafen is te gebruiken bij het maken van een rooster. Het belangrijkste van een rooster is dat alle lessen gevolgt en gegeven kunnen worden. Een leerling kan bijvoorbeeld niet op hetzelfde moment twee lessen hebben, en een docent kan niet twee lessen tegelijk geven. Om dit te voorkomen kan iedere les die gegeven moet worden, voorgesteld worden als een knoop. Tussen die knopen worden verbindingen aangelegd. Wanneer twee lessen

eenzelfde leerling dan wel docent hebben, dan moeten ze verbonden worden. De lessen kunnen dan namelijk niet op hetzelfde moment gegeven worden. Alle verbonden lessen moeten een andere kleur krijgen. Een knoop mag dus niet dezelfde kleur hebben als een van de buren[5].
Hierdoor kan dus iedere knoop met dezelfde kleur sowieso op hetzelfde moment geroosterd worden. Een volgende stap kan zijn om te beginnen met het inroosteren van de kleuren van knopen die het meest voorkomen, omdat je dan in een minimaal aantal uren zoveel mogelijk lessen in kunt plannen.

3.9 Complexiteit

De complexiteit van een probleem geeft aan hoe moeilijk het probleem is. De complexiteitstheorie wordt gebruikt om computationele problemen te classificeren in catergoriën en de oplosbaarheidsgraad van dit probleem aan te geven. De primaire 'kost' of 'moeilijkheid' van een probleem is te tijd die het kost om het probleem op te lossen (Wikipedia-contributors, 2021a).

3.9.1 Grootte van probleemgevallen

De moeilijkheidsgraad van een probleem kan gemeten worden door te kijken naar de tijd die het kost als je een optimaal algoritme gebruikt om het probleem op te lossen.
Neem de grootte van de invoer n, dan kan de looptijd uitgedrukt worden als functie van n. En omdat ook de looptijd kan verschillen voor problemen van dezelfde grootte kijken we altijd naar het probleem met de langste looptijd, de slechtste-geval complexiteit $T(n)$.

3.9.2 Snel of langzaam?

Algoritmes kunnen gecategoriseerd worden op basis van de zwaarste bepalende factor van de functie in de slechtste-geval complexiteit $T(n)$. De 'zwaarst' wegende polynomiale factor is dan de grote-O van de functie $T(n)$. De grote O toont niet hoe snel of langzaam een algoritme is, maar wel hoe snel of langzaam de complexiteit van een functie toeneemt. Een aantal veel voorkomende grote-O's zijn (oplopend in complexiteit) (Wikipedia-contributors, 2020, Wikipedia-contributors, 2021b):

- O(1); waar de rekentijd onafhankelijk is van de variabele n
- O($log(n)$); waar de rekentijd logaritmisch evenredig is aan de variabele n
- O(n); waar de rekentijd lineair evenredig is aan de variabele n
- O($n \cdot log(n)$); waar de rekentijd lineair maal logaritmisch evenredig is aan n
- O(n^2); waar de rekentijd kwadratisch evenredig is aan de variabele n
- O(2^n); waar de rekentijd exponentieel evenredig is aan de variabele n
- O($n!$); waar de rekentijd factoriaal evenredig is aan de variabele n

Voorbeeld

$$T_1(n) = 3n^2 + 8log(n) + 8n - 5 \tag{3.25}$$

Formule 3.25 heeft als zwaarst wegende polynomiale factor $3n^2$, want als n groter wordt maken $8log(n)$ en $8n$ minder en minder uit ten opzichte van $3n^2$ voor toename van de functie. Echter wordt voor het bepalen van de grote-O de constante voor de variabele niet meegenomen, omdat deze geen tijd aan de rekentijd toevoegt. De grote-O van $T_1(N)$ (formule 3.25) is dus n^2.

$$T_1(n) = 3n^2 \tag{3.26}$$

Belangrijk

Hoewel formule 3.25 en formule 3.26 dezelfde grote O hebben, zal voor eenzelfde n formule 3.26 altijd sneller zijn dan formule 3.25. Echter neemt de complexiteit en dus rekentijd wel even snel toe.

3.9.3 Classificatie algoritmes in context

Hoewel in theorie enkel de zwaarst wegende polynomiale factor de grote O bepaald, hoeft dit in de praktijk niet zo te zijn. Elke computatie die rondom een functie in de computercode staat, heeft invloed op de computatietijd. Zo kan het zijn dat je een lineare functie berekent (deze heeft dus $O(n)$), maar dat deze in een nested for loop staat (nog uitleggen wat een nested for loop is). Dan heb je een functie die n maal itereert, met daarin een functie die n maal itereerd, met daarin een lineare functie. Wanneer de lineaire functie ook afhankelijk is van n, en enkel n, is de algehele grote-o-functie is dan $O(n^3)$. Wanneer de lineaire functie van een andere variabele afhankelijk is, bijvoorbeeld m, dan is de algehele grote-O-functie $O(n^2 \cdot m)$.

Hoofdstuk 4

Probleemstelling

Het probleem wat aangepakt zal worden is een onderbouwrooster voor een 'doorsnee' middelbare school. De 'doorsnee' middelbare school (waar vanaf nu naar verwezen zal worden als middelbare school), voldoet aan de volgende eisen:

- Heeft alleen klassen, geen clusterklassen
- Leerlingen hebben geen keuzevakken, alle leerlingen in een klas volgen dus dezelfde lessen
- Leerlingen volgen alle lessen met één en dezelfde klas
- Een docent geeft maar één vak
- Alle lessen die een klas van een vak krijgt, krijgt deze gedurende het gehele jaar van dezelfde docent
- Zowel leerlingen als docenten hebben maar één vak tegelijk; er zijn geen dubbel ingeplande uren

Bij deze school moeten lesuren ingeroosterd worden, conform de beschikbaarheid van docenten.

Deze eisen gelden ook meteen als afhankelijke variabelen.
Verder zijn er een aantal voorkeuren, welke op chronologisch aflopende volgorde prioriteit hebben bij het minimaliseren ofwel maximaliseren:

1. Minimaliseer totaal aantal tussenuren in de gehele school
2. Minimaliseer latere uren
3. Minimaliseer het aantal derde (of meer) uren van eenzelfde vak op een dag.

Hoofdstuk 5

Oplossingen

5.1 Kleinste kwadraten methode

De kleinste kwadraten methode kan gebruikt worden, door elke ongewenste eigenschap (bijv. een tussenuur) een strafpunt te geven. Dit strafpunt is de x-waarde van een punt. De frequentie waarmee dit strafpunt voorkomt (bijvoorbeeld de som van alle strafpunten met waarde 54) is de y-waarde van dit punt. Door alle punten kan een trendlijn getrokken worden. Wanneer het oppervlak onder de trendlijn geminimaliseerd wordt, worden zoveel mogelijk ongewenste eigenschappen geëlimineerd.

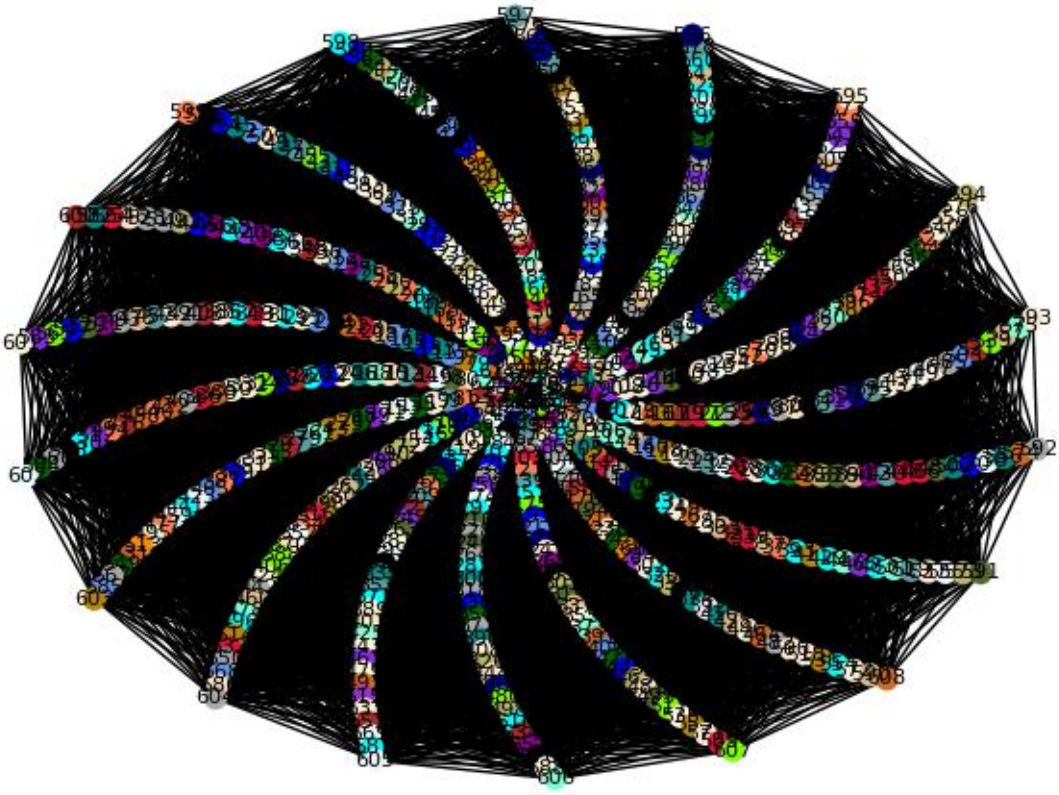
5.2 Grafentheorie

Graventheorie kan gebruikt worden om een oplossing te maken die voldoet aan de eis dat een leerling of docent niet twee lessen op hetzelfde moment mag hebben. Dit kan gedaan worden met behulp van graph colouring (3.8.1). Alle lessen die aan elkaar verbonden zijn krijgen dezelfde kleur, en mogen dus niet op hetzelfde moment plaatsvinden. Een verbinding tussen twee knopen betekent dat een leerling of de docent hetzelfde is.
Er dient dan een eigen ontworpen algoritme gemaakt te worden om tussenuren op te vullen, of bijvoorbeeld eerder op de dag in te roosteren.

5.2.1 Knopen verbinden

Alle lessen die een onderlinge connectie hebben, dezelfde docent of dezelfde leerling(en), worden verbonden met zijden. In appendix B staat beschreven hoe dit gedaan wordt.
Alle verbonden lessen mogen niet op hetzelfde moment ingeroosterd worden en krijgen een andere kleur.

Als dit gedaan wordt met de test data uit appendix E en het programma uit appendix F , wordt een graaf als figuur 5.1 verkregen.



Figuur 5.1: Graaf van test data (appendix E)

In figuur 5.1 is te zien dat er heel veel onderlinge verbinden zijn, er zijn 32 verschillende kleuren gebruikt voor deze graaf. Tussen alle 609 knopen (lessen) zijn 14.763 zijden getrokken om een onderlinge connectie van leerling of docent aan te geven.

5.3 Lineair programmeren

Om dit probleem op te lossen zal onder andere gebruik gemaakt worden van lineair programmeren. Elke doelfunctie en restrictie kan als een wiskundige formule worden genoteerd, alvorens geminimaliseerd danwel gemaximaliseerd te worden.

Bij dit onderzoek is gebruik gemaakt van de Python module python-mip[1]. Deze module lost de gegeven formules op. Deze module gebruikt ook een Branch & Bound (hoofdstuk 3.7) algoritme om het probleem te optimaliseren.

Allereerst moet het probleem verwoord worden als een LP-probleem (lineair programmeren probleem).

[1]Python-MIP: https://www.python-mip.com/

5.3.1 Verzamelingen

allereerst kan de verzameling `H` in formule 5.1 gegenereerd worden. Dit is de verzameling die alle uren van de week *(5 werkdagen, 9 uren per werkdag)* representeert. Dit resulteert in de dag *(d)* en het uur *(h)* door $(d, h) \in H$.

$$H \subseteq 5 \times 9 \tag{5.1}$$

De tweede verzameling bestaat uit alle lessen die ingeroosterd moeten worden, verzameling L is beschreven in formule 5.2, waarbij N staat voor het aantal lessen dat ingeroosterd moet worden.

$$L \subseteq (t, g) \times N \tag{5.2}$$

De functie `amount` geeft terug hoe vaak een les moet worden ingeroosterd. Daarnaast zijn er nog de verzamelingen met alle docenten en klassen, voor de docenten geldt $t \in T$ en voor de klassen geldt $g \in G$.

In formule 5.3 staat de verzameling met alle mogelijke lessen beschreven, verzameling S. Deze verzameling bevat alle mogelijkheden waarop een les ingeroosterd kan worden, in dit geval dus alle 45 uren die in een week zitten.

$$S = \{(d, h, t, g, 0) \mid (d, h) \in H \mid (t, g) \in L\} \tag{5.3}$$

De beslisvariabel, de 0, is een boolean (0 of 1) die aangeeft of de les ingeroosterd is of niet. De beslisvariabel is te verkrijgen met de functie `var`. De functie `tf` geeft de docent terug en de functie `gf` geeft de klas terug.

5.3.2 Voorwaarden

De eerste voorwaarde is dat alle lessen ingeroostert moeten worden, dit betekent dat de lengte van de set S gelijk moet zijn aan het aantal lessen dat ingeroostert moet worden volgens set L, zoals beschreven in formule 5.4.

$$|S| = \sum_{amount \in L} amount \tag{5.4}$$

De tweede voorwaarde zorgt ervoor dat een les niet te vaak ingepland wordt. Deze voorwaarde is beschreven in formule 5.5.

$$\sum \{s \mid s \in S, tf(s) = t, gf(s) = g\} \leq amount \; \forall \; (t, g, amount) \in L \tag{5.5}$$

Wanneer aan alle bovenstaande voorwaarden wordt voldaan (formules 5.4 en 5.5), wordt een rooster zonder conflicten gegenereerd. Dit is dus ook een mogelijke oplossing voor het haalbaarheidsprobleem.

5.3.3 Doelfuncties

Uren naar voren plaatsen

Er kan een poging gedaan worden om alle uren naar voren te plaatsen. Het naar voren plaatsen van uren minimaliseert indirect ook (deels) het aantal tussenuren. Het minimaliseren

van tussenuren bleek technisch gezien niet mogelijk, hierover meer in de discussie (hoofdstuk 8).
De doelfunctie hiervoor staat beschreven in formule 5.6.

$$f(G) = \sum_{g \in G} \Big(\sum_{(uur,var) \in g} uur \cdot var \Big) \qquad \text{Minimaliseer: } f(G) \tag{5.6}$$

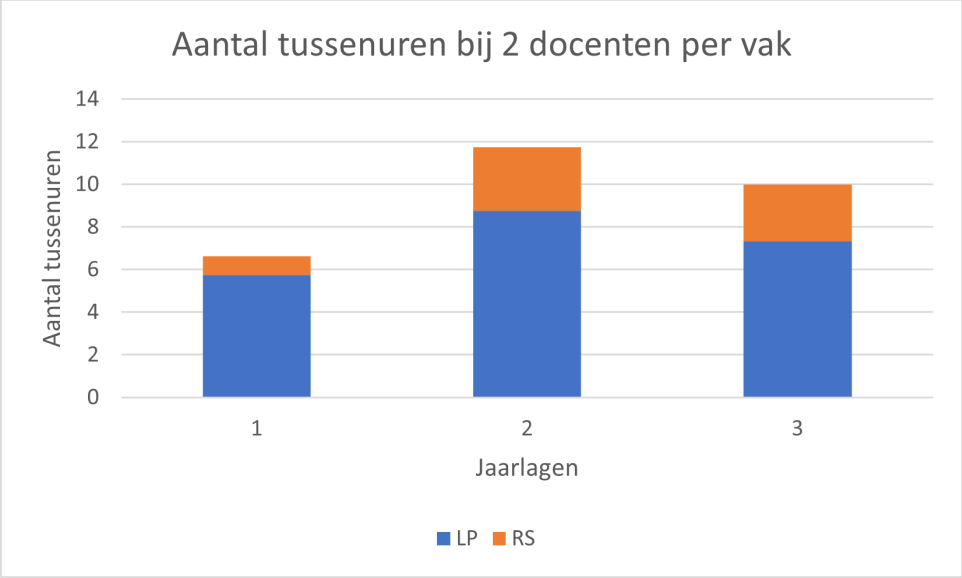
Hierbij is G een verzameling van alle groepen, waarbij iedere groep een verzameling aan alle mogelijke lessen heeft.

Hoofdstuk 6

Resultaten

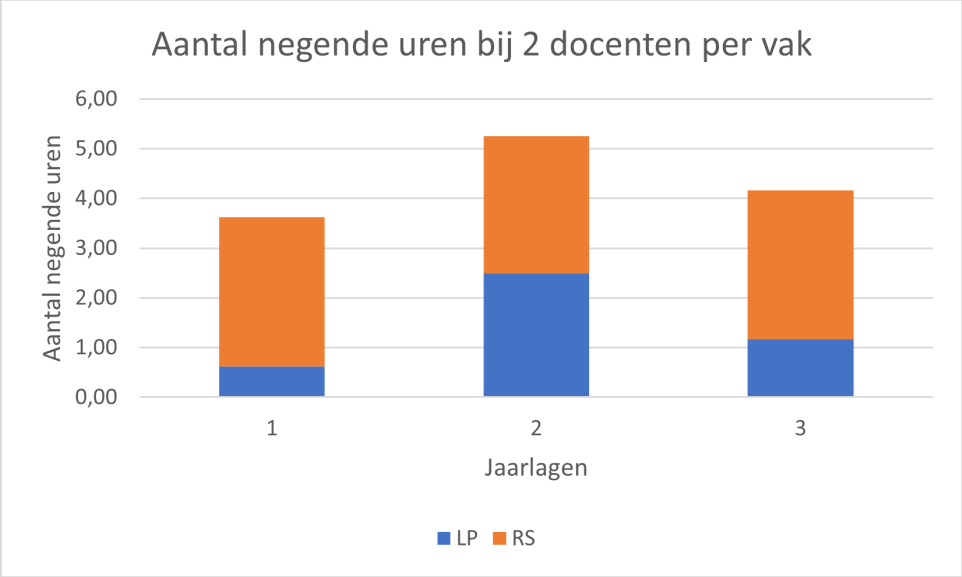
We zullen kijken naar twee van de voorkeuren van een leerling om te bepalen hoeveel ons algoritme toevoegt aan het roosteren. We gaan het aantal tussenuren vergelijken en het aantal negende uren dat een klas heeft met een variatie in het aantal docenten dat per vak beschikbaar is.

6.1 Twee docenten per vak



Figuur 6.1: Aantal tussenuren bij 2 docenten per vak, LP = lineair programmeren en RS = random solver, willekeurige oplosser

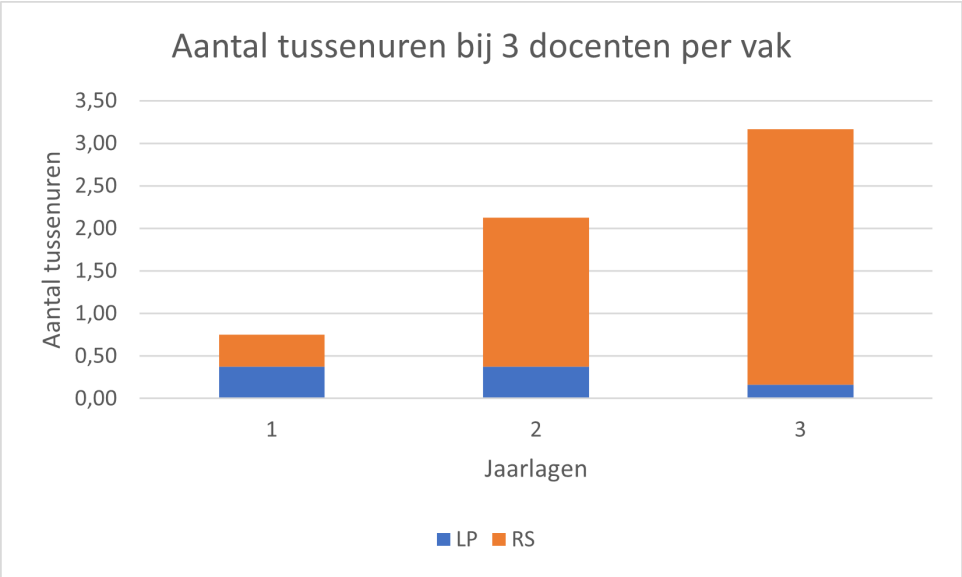
In figuur 6.1 is te zien dat met twee docenten per vak de meeste tussenuren ontstaan bij het gebruik van de lineair programmeren oplosser en dat de willekeurige oplosser de minste tussenuren genereert.



Figuur 6.2: Aantal negende uren bij 2 docenten per vak, LP = lineair programmeren en RS = random solver, willekeurige oplosser

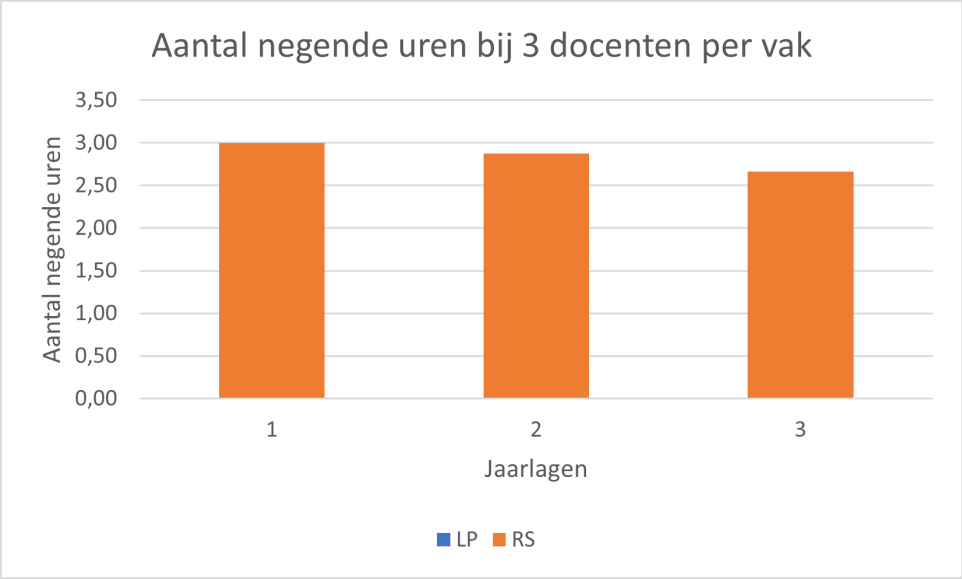
In figuur 6.2 is te zien dat met twee docenten per vak de meeste negende uren ontstaan bij het gebruik van de willekeurige oplosser en dat de lineair programmeren oplosser dit beter doet.

6.2 Drie docenten per vak



Figuur 6.3: Aantal tussenuren bij 3 docenten per vak, LP = lineair programmeren en RS = random solver, willekeurige oplosser

In figuur 6.3 is te zien dat met drie docenten per vak de meeste tussenuren ontstaan bij het gebruik van willekeurige oplosser en dat de lineair programmeren oplosser dit beter doet.



Figuur 6.4: Aantal negende uren bij 3 docenten per vak, LP = lineair programmeren en RS = random solver, willekeurige oplosser

In figuur 6.2 is te zien dat met drie docenten per vak de alle negende uren ontstaan bij het gebruik van de willekeurige oplosser en dat bij het gebruik van de lineair programmeren oplosser geen enkel negende uur ontstaat.

Hoofdstuk 7

Conclusie

7.1 Terugkoppeling deelvragen

Hoe wordt een rooster nu ontworpen?
Een rooster wordt op dit moment in vier stappen geschreven. Eerst worden gegevens verwerkt. Hierna wordt een initieel kloppend rooster gemaakt. Vervolgens worden doelfuncties losgelaten op het initiële rooster, om voorkeuren toe te passen. Of er hierna nog aanpassingen worden gedaan en welke aanpassingen verschilt per school, op het Openbaar Lyceum Zeist worden er hierna nog handmatig aanpassingen gedaan aan het rooster om het passender voor de school te maken.

Van welke factoren en voorkeur hangt het maken van een rooster af?
Het maken van een rooster hangt af van een aantal factoren. Zo moet een leerling beschikbaar zijn, een docent beschikbaar zijn, een lokaal beschikbaar zijn en een uur beschikbaar zijn. Wanneer alle 4 beschikbaar zijn, kan een uur ingeroosterd worden.

Hoe worden soortgelijke problemen aangepakt?
Het roosteringsprobleem wordt door Zermelo Roostermakers opgelost door het gebruik van graventheorie en hun eigen ontworpen algoritmes.

Welk van de afhankelijke voorkeuren vindt een middelbare scholier zwaarder wegen?
Een middelbare scholier heeft liever minder tussenuren dan dat deze eerder uit is. Dus het vijfde en het zesde les hebben is fijner dan het eerste en vierde. Verder vind een leerling het fijn om uren van eenzelfde vak te verspreiden over de week.

Op welke manier kan het roosteroptimalisatieprobleem aangepakt worden?
Er zijn een aantal strategieën waarmee geëxperimenteerd is gedurende dit onderzoek. Zo is lineair programmeren handig voor het toevoegen van doelfuncties, maar kan voor een initiaal rooster beter graventheorie gebruikt worden. Ook is er geëxperimenteerd met het minimaliseren van ongewenste roostereigenschappen door middel van het oppervlak van een trendlijn die is opgesteld met de kleinste-kwadraten-methode. Het gebruik van de kleinste-kwadraten-methode heeft echter niet geresulteerd in een werkend produkt.

7.2 Terugkoppeling hoofdvraag

Om terug te koppelen naar de hoofdvraag (Is het mogelijk een programma te schrijven welke in een realistische tijdsspanne een functioneel rooster voor een middelbare school van 1000 leerlingen kan genereren?):

Ja, dit is mogelijk. Echter zitten er een aantal haken en ogen aan. Het ontwerpen van een dergelijk algoritme is erg haalbaar, en wordt door bedrijven als Zermelo ook al gedaan. Echter is de ontwikkelingstijd van een profielwerkstuk te weinig om een rooster voor een school ter grootte van 1000 leerlingen een functioneel rooster te maken.

Het implementeren van de verschillende voorkeuren, restricties en randvoorwaarden kost tijd, en in de 80 uur per persoon die het profielwerkstuk ter beschikking stelt kunnen er maar enkelen geïmplementeerd worden. Zo is er omwille van tijd bij dit profielwerkstuk gekozen voor een doelfunctie die zo veel mogelijk uren naar voren plaatst (hoofdstuk 5.3.1). Het resultaat hiervan is zichtbaar in hoofdstuk 6. Wanneer we kijken naar figuur 6.3 is te zien dat het aantal negende uren in alledrie de jaarlagen afwezig is, terwijl bij het random inroosteren nog wel negende uren in het rooster zitten.

7.2.1 Onze resultaten

Er is een duidelijk resultaat zichtbaar in het ontworpen algoritme. Wanneer het algoritme dat willekeurig uren inroosterd vergeleken wordt met het geoptimalisteerde algoritme is zichtbaar dat als er genoeg docenten per vak beschikbaar zijn (3) het aantal tussenuren veel minder is bij de LP oplosser dan bij de RS (willekeurige oplosser). Het aantal negende uren bij de het geoptimaliseerde algoritme is zelfs 0 waarbij de willekeurige inrooster oplosser wel negende uren heeft. Wanneer er niet genoeg docenten zijn per vak (namelijk 2) is zichtbaar dat het aantal tussenuren veel hoger is bij de LP oplosser dan bij de RS oplosser, dit is echter te verklaren als we kijken naar het aantal negende uren. Hierbij wint de LP oplosser namelijk wel. De LP oplosser probeert uren naar eerder op de dag te plaatsen, maar houd niet direct rekening met tussenuren. De LP solver zorgt er dus wel voor dat je gemiddeld eerder uit bent en indirect voor een gedeelte de tussenuren eruit haalt, het algoritme wil namelijk dat je zo vroeg mogelijk thuis bent en tussenuren zijn onbenutte uren die wel tijd innemen.

Hoofdstuk 8

Discussie

8.1 Discussiepunten huidig onderzoek

Hoewel dit onderzoek maatschappelijk heel relevant is (het roosteren van schoolroosters komt immers voor in het leven van elke leerling), is de maatschappelijke bijdrage niet van significante waarde. Alles wat in dit profielwerkstuk ontwikkeld danwel uitgeprobeerd is, is al eerder gedaan en is dus niks nieuws.

Doordat onvoldoende onderzoek aan de voorhand gedaan is, is er lang geëxperimenteerd met het haalbaar krijgen van doelfuncties die programmeer-technisch onhaalbaar bleken. Dit heeft een hoop tijd gekost, die bij andere onderdelen ten koste van resultaat is gegaan. Als we eerder met experts gesproken hadden hadden we eerder kunnen weten dat we bezig waren met een manier die waarschijnlijk niet het beste was, waardoor we meer tijd hadden gehad om andere methodes uit te proberen. We hebben pas laat gesproken met experts van Zermelo waardoor we veel tijd zijn kwijtgeraakt.

Hoewel de onderzoeksvraag een complete school van 1000 leerlingen als toetsing stelt, zijn er slechts resultaten van een leerling of 250. Tevens waren dit allemaal onderbouwklassen, terwijl de echte complexiteit pas bij een bovenbouw om de hoek komt kijken.

Bij het testen van de algoritmes en het doen van de metingen is maar van 1 onderbouw van een school uitgegaan. Hierdoor kan het zijn dat het lineair geprogrammeerde algoritme niet per se goed is in het inroosteren van elke onderbouw, maar alleen in het inroosteren van een onderbouw die lijkt op die van het OLZ in schooljaar 2020/2021.

Bij het vergaren van de voorkeur van een leerling voor roosters is uit gegaan van de mening van twee bèta-leerlingen. Dit is een erg kleine, tevens ongevarieerde groep leerlingen, waardoor het beeld van wat een leerling zwaar vind wegen wel eens vertekend zou kunnen zijn. In een vervolgonderzoek kan het verstandig zijn om een enquête onder meer leerlingen los te laten.

8.2 Suggesties vervolgonderzoek

In een vervolgonderzoek kan gekeken worden naar bovenbouwklassen met keuzenvakken. Dit is een significant complexer onderdeel van het lesroosteren waar nog een hoop bereikt kan worden. Tevens kunnen er meer variaties in aantal lessen, hoeveelheid uren van de lessen, en grootte van de school geëxperimenteerd worden.

Er is geen rekening gehouden met de beschikbaarheid van lokalen bij het ontwerpen van dit roosteringsalgoritme. Dit zou in het vervolg wel gedaan kunnen worden, opdat de vakspecifieke lokalen ook daadwerkelijk beschikbaar zijn.

In een vervolgonderzoek kan verder gewerkt worden op de doelfuncties die met dit onderzoek ten gronde zijn gebracht. Doordat de basis voor het genereren van een schoolrooster schaalbaar ontwikkeld is, kunnen er modulaire doelfuncties, leerlingen, docenten of zelfs vakken aan worden toegevoegd.

Tevens zou het allicht interessant zijn om te kijken of graventheorie met branch-and-bound gecombineerd kan worden. Dit, omdat graventheorie relatief snel is voor het vinden van onderlinge verbanden en het inroosteren van een initiaalrooster ten opzichte van de andere geëxperimenteerde varianten, terwijl branch-and-bound voor 0-1 integer programmeringen relatief snel is voor het benaderen van optimale antwoorden van doelfuncties.

Tot slot kan ook het gebruik van de kleinste-kwadratenmethode onderzocht worden, mogelijk kan deze methode gebruikt worden in combinatie met de graventheorie.

Hoofdstuk 9

Bibliografie

Bakker, H., Boon, B., Bos, D., Doekes, W., Peereboom, H., Reek, K. v. d., Reijenga, R., Schaberg, G., Sinkeldam, R., Wallien, C. & et al. (2017). *Moderne Wiskunde 6 vwo D* (11de ed.). Noordhoff Uitgevers.

Bazett, D. T. (2021). Intro to Linear Programming and the Simplex Method. Verkregen 2 december 2021, van https://www.youtube.com/watch?v=K7TL5NMlKIk

Casquilho, M. (g.d.). Integer Programming; the Branch and Bound method. *Technico*. http://web.tecnico.ulisboa.pt/mcasquilho/compute/_linpro/TaylorB_module_c.pdf

Dijk, J. v., Staijen, S. & Finkelnberg, H. (2021). Roosterontwerp volgens Zermelo ontwikkelaars.

Dijk, J. v., Staijen, S. & Pomstra, K. v. (2021). Roosterontwerp op het OLZ.

Flovik, V. (2021). What is graph theory, and why should you care? Verkregen 2 december 2021, van https://towardsdatascience.com/what-is-graph-theory-and-why-should-you-care-28d6a715a5c2

Gunawan, A., Ng, K. & Poh, K. (2006). A Mathematical Programming Model For A Timetabling Problem., 42–47. https://www.researchgate.net/publication/221143006_A_Mathematical_Programming_Model_For_A_Timetabling_Problem

Katwijk, B. v. (2013). *Een selectie algoritmen voor lineair programmeren* (proefschrift). https://repository.tudelft.nl/islandora/object/uuid:da0ade3f-e05a-41e1-97de-28d672842d78/datastream/OBJ/download

Staijen, S. & Pluut, C. (2021). Roosterontwerp volgens Rector tevens onderwijsdeskundige.

Wikipedia-bijdragers. (2020). *Vierkleurenstelling*. https://nl.wikipedia.org/wiki/Vierkleurenstelling (opgevraagt op: 31.10.2021)

Wikipedia-bijdragers. (2021). *Kleuren van grafen*. https://nl.wikipedia.org/wiki/Kleuren_van_grafen (opgevraagt op: 31.10.2021)

Wikipedia-contributors. (2020). *Polynomiale tijd*. https://nl.wikipedia.org/wiki/Polynomiale_tijd Opgevraagd op: 19.11.2021

Wikipedia-contributors. (2021a). *Computational complexity*. https://en.wikipedia.org/wiki/Computational_complexity (opgevraagt op: 23.11.2021)

Wikipedia-contributors. (2021b). *Time complexity*. https://en.wikipedia.org/wiki/Time_complexity (opgevraagt op: 23.11.2021)

with Mr Orys, M. H. (2019). The Simplex Algorithm - Integer Solutions. Verkregen 21 december 2021, van https://www.youtube.com/watch?v=tsXlclDkzuA

Hoofdstuk 10

Evaluatie

10.1 Persoonlijke evaluatie Sam

De onderlinge communicatie tussen Joep en mij verliep over het algemeen goed. Wij zijn gedurende het gehele profielwerkstuk erg gemotiveerd geweest om een zo goed mogelijk resultaat neer te zetten, en aan het profielwerkstuk te werken.

Echter denk ik dat ik voor ons beiden spreek als we heel goed zijn in het afdwalen naar een zijspoor. Het wil nog wel eens voorkomen dat ik een interessant onderwerp vind wat net niet aansluit op hetgeen waarmee ik bezig ben, waardoor ik waardevolle tijd 'verspil' aan dingen die niet nodig zijn voor het onderzoek waarmee ik bezig ben.

Wij zijn heel goed begonnen met het plannen van het profielwerkstuk, en het werken aan het profielwerkstuk is geen probleem geweest. Echter hebben wij, zoals eerder besproken in de discussie, een tijd lang dingen onderzocht, welke later irrelevant voor dit onderzoek bleken.

Tevens kunnen we volgende keer voor onszelf beter harde deadlines stellen. Nu was het vaak zo dat ondanks dat de voor onszelf weggeschreven tijd reeds bereikt was, dat we toch nog probeerden iets af te maken. Op een gegeven moment moet het besluit genomen worden dat de resultaten die je hebt het maximale haalbare zijn, en anderhalve week voor de eindverslag deadline is hiervoor wat kortdag.

Waar ik wel erg trots op ben is de onderlinge samenwerking op het gebied van kennen- en kunnen. Joep en ik durfden eerlijk naar elkaar te zijn als we iets lastig vonden, of dachten dat de ander iets beter kon dan jijzelf. Zo hebben we op een gegeven moment besloten dat Joep zich meer op het programmeerwerk zou richten, en dat Sam parallel hieraan verder zou werken aan het uitzoeken wat de volgende stap in het proces was (en dit meteen opnam in het verslag), die Joep vervolgens weer kon implementeren. Dit heeft ervoor gezorgd, dat het proces meer meters maakte.

10.2 Persoonlijke evaluatie Joep

Ik denk dat Sam en ik goed hebben samengewerkt tijdens ons profielwerkstuk. We hebben goed en duidelijk met elkaar gecommuniceerd wat het werken een stuk efficiënter maakte.

Jammer is dat we veel tijd hebben besteed aan dingen die uiteindelijk niet nodig of niet nuttig bleken. We hebben bijvoorbeeld een wiskundige methode onderzocht die uiteindelijk niet passend bleek voor ons onderzoek, dat is natuurlijk nodig maar het voelt als verspilde tijd. Hetzelfde hebben we gehad met lineair programmeren, uiteindelijk blijkt het een vrij goeie oplossing te zijn maar niet de beste zoals we in de discussie noemen. We hebben daar heel veel tijd aan besteed waardoor we geen tijd meer over hadden om andere methoden langs te gaan en uit te werken.

Ik ben blij dat we over het algemeen gemotiveerd waren om aan ons profielwerkstuk te werken, hoewel we er soms tegenop zagen omdat we iets saais of ingewikkelds moesten doen hadden we er over het algemeen zin in.

Het was handig geweest om een planning voor het werkstuk te maken. Dan hadden we waarschijnlijk efficiënter en doelgerichter gewerkt wat tijd had bespaard. Dan hadden we doelen gehad om bijvoorbeeld dat hoofdstuk van de theorie voor die dag af te hebben waardoor we concrete plannen hadden in plaats van "maar wat doen".

We hebben goed gebruik gemaakt van het feit dat de één beter is in het ene, en de ander beter is in het andere. Zo heb ik veel gewerkt aan het programmeren, maar heeft Sam weer meer gedaan aan het verslag en hoe we dingen kunnen verwoorden. Zo hebben we van elkaars kwaliteiten gebruik gemaakt.

Tot slot ben ik erg blij met en trots op het resultaat dat we hebben behaald. Als ik van tevoren had geweten dat we zulke lastige dingen zouden moeten begrijpen was ik waarschijnlijk wat bang geworden, maar het is ons uiteindelijk toch nog gelukt.

Bijlage A

Lijst van wiskundige symbolen

Symbool	**Naam/betekenis**	**Voorbeeld**
$\{...\}$	Verzameling	$A = \{3, 6, 9\}$
$\mathbb{Z}$ (Zahlen)	Verzameling van alle gehele getallen	$\mathbb{Z} = \{..., -2, -1, 0, 1, 2, ...\}$
$\in$ (in)	Zit in de verzameling	$-6 \in \mathbb{Z}$
$\subseteq$	Subset	$\{1, 2, 3\} \subseteq \{1, 2, 3\}$ (A is een subset van B)
$\|$	Zo dat	$\mathbb{Z}^+ = \{x \| x \in \mathbb{Z}, x > 0\}$
$\|A\|$	Het aantal elementen in de verzameling A	$\|A\| = 3$
$\sum A$	Som	$\sum A = 3 + 6 + 9 = 19$
$\forall$	Voor alle	$\forall x \in \mathbb{Z}^+, x > 0$
$\lfloor x \rfloor$	afronden naar beneden	$\lfloor 3.5 \rfloor = 3$
$\lceil x \rceil$	afronden naar boven	$\lceil 3.5 \rceil = 4$

Bijlage B

Lessen verbinden

```
# Loop through every combination of two lessons
for (l1, l2) in combinations(lessons, 2):
    # Check if the group or teacher is the same
    if l1.group == l2.group or l1.teacher == l2.teacher:
        # Create edge between lessons
        add_edge(l1.identifier, l2.identifier)
```

Bijlage C

Lineair programmeren - het timmerman probleem

```python
# Linear programming - carpenter problem
# https://github.com/JOeppp/pws/blob/main/mip_examples/carpenter_problem.py
from mip import Model, INTEGER, maximize

# Create the model
model = Model("Timmerman")

# Create the variables x and y of type INTEGER
x = model.add_var(var_type=INTEGER)
y = model.add_var(var_type=INTEGER)

# Add the constraints
model += 5 * x + 4 * y <= 80
model += 10 * x + 20 * y <= 200
model += x >= 0
model += y >= 0

# Set the objective function of the model
model.objective = maximize(180 * x + 200 * y)

# Optimize
model.optimize()

# Print the values
print(f"Found the optimal value on ({x.x}, {y.x}), optimal value: {model.objective_value
```

Bijlage D

Test input voor algoritmes, onderbouw 2 docenten

```
{
    "subjectInformation": [
        {
            "subject": "Nederlands",
            "year": 1,
            "amount": 4
        },
        {
            "subject": "Engels",
            "year": 1,
            "amount": 4
        },
        {
            "subject": "Wiskunde",
            "year": 1,
            "amount": 4
        },
        {
            "subject": "Biologie",
            "year": 1,
            "amount": 2
        },
        {
            "subject": "BV",
            "year": 1,
            "amount": 2
        },
        {
            "subject": "LO",
            "year": 1,
            "amount": 4
```

```
        },
        {
            "subject": "Frans",
            "year": 1,
            "amount": 3
        },
        {
            "subject": "Geschiedenis",
            "year": 1,
            "amount": 3
        },
        {
            "subject": "NaSk",
            "year": 1,
            "amount": 3
        },
        {
            "subject": "Mentorles",
            "year": 1,
            "amount": 1
        },
        {
            "subject": "Nederlands",
            "year": 2,
            "amount": 4
        },
        {
            "subject": "Engels",
            "year": 2,
            "amount": 4
        },
        {
            "subject": "Wiskunde",
            "year": 2,
            "amount": 4
        },
        {
            "subject": "Biologie",
            "year": 2,
            "amount": 2
        },
        {
            "subject": "BV",
            "year": 2,
            "amount": 2
        },
```

```
        {
            "subject": "LO",
            "year": 2,
            "amount": 4
        },
        {
            "subject": "Frans",
            "year": 2,
            "amount": 3
        },
        {
            "subject": "Geschiedenis",
            "year": 2,
            "amount": 3
        },
        {
            "subject": "NaSk",
            "year": 2,
            "amount": 3
        },
        {
            "subject": "Mentorles",
            "year": 2,
            "amount": 1
        },
        {
            "subject": "Nederlands",
            "year": 3,
            "amount": 4
        },
        {
            "subject": "Engels",
            "year": 3,
            "amount": 4
        },
        {
            "subject": "Wiskunde",
            "year": 3,
            "amount": 4
        },
        {
            "subject": "Biologie",
            "year": 3,
            "amount": 2
        },
        {
```

```
            "subject": "BV",
            "year": 3,
            "amount": 2
        },
        {
            "subject": "LO",
            "year": 3,
            "amount": 4
        },
        {
            "subject": "Frans",
            "year": 3,
            "amount": 3
        },
        {
            "subject": "Geschiedenis",
            "year": 3,
            "amount": 3
        },
        {
            "subject": "NaSk",
            "year": 3,
            "amount": 3
        },
        {
            "subject": "Mentorles",
            "year": 3,
            "amount": 1
        }
    ],
    "groups": [
        {
            "name": "zhk1a",
            "year": 1,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
```

```
        },
        {
            "name": "zhk1b",
            "year": 1,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "zhv1a",
            "year": 1,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "zhv1b",
            "year": 1,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
```

```
            ],
            "lessons": []
        },
        {
            "name": "zhv1c",
            "year": 1,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "za1a",
            "year": 1,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "za1b",
            "year": 1,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
```

```
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "zg1a",
            "year": 1,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "zhv2b",
            "year": 2,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "zhv2c",
            "year": 2,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
```

```
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "za2a",
            "year": 2,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "za2b",
            "year": 2,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "za2c",
            "year": 2,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
```

```
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "za2d",
            "year": 2,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "zg2a",
            "year": 2,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "zh3a",
            "year": 3,
            "subjects": [
                "Nederlands",
```

```
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "zh3b",
            "year": 3,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "zh3c",
            "year": 3,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "za3a",
            "year": 3,
```

```
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "za3b",
            "year": 3,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "za3c",
            "year": 3,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        }
    ],
```

```
    "teacherInfo": [
        {
            "subject": "Nederlands",
            "prefix": "Ned",
            "amount": 2
        },
        {
            "subject": "Engels",
            "prefix": "Eng",
            "amount": 2
        },
        {
            "subject": "Wiskunde",
            "prefix": "Wisk",
            "amount": 2
        },
        {
            "subject": "Biologie",
            "prefix": "Bio",
            "amount": 2
        },
        {
            "subject": "BV",
            "prefix": "BV",
            "amount": 2
        },
        {
            "subject": "LO",
            "prefix": "LO",
            "amount": 2
        },
        {
            "subject": "Frans",
            "prefix": "Fr",
            "amount": 2
        },
        {
            "subject": "Geschiedenis",
            "prefix": "Gs",
            "amount": 2
        },
        {
            "subject": "NaSk",
            "prefix": "Nk",
            "amount": 2
        }
```

```
    ],
    "lessons": [],
    "amountOfDaysAWeek": 5,
    "amountOfHoursADay": 9
}
```

Bijlage E

Test input voor algoritmes, onderbouw 3 docenten

```
{
    "subjectInformation": [
        {
            "subject": "Nederlands",
            "year": 1,
            "amount": 4
        },
        {
            "subject": "Engels",
            "year": 1,
            "amount": 4
        },
        {
            "subject": "Wiskunde",
            "year": 1,
            "amount": 4
        },
        {
            "subject": "Biologie",
            "year": 1,
            "amount": 2
        },
        {
            "subject": "BV",
            "year": 1,
            "amount": 2
        },
        {
            "subject": "LO",
            "year": 1,
            "amount": 4
```

```
        },
        {
            "subject": "Frans",
            "year": 1,
            "amount": 3
        },
        {
            "subject": "Geschiedenis",
            "year": 1,
            "amount": 3
        },
        {
            "subject": "NaSk",
            "year": 1,
            "amount": 3
        },
        {
            "subject": "Mentorles",
            "year": 1,
            "amount": 1
        },
        {
            "subject": "Nederlands",
            "year": 2,
            "amount": 4
        },
        {
            "subject": "Engels",
            "year": 2,
            "amount": 4
        },
        {
            "subject": "Wiskunde",
            "year": 2,
            "amount": 4
        },
        {
            "subject": "Biologie",
            "year": 2,
            "amount": 2
        },
        {
            "subject": "BV",
            "year": 2,
            "amount": 2
        },
```

```
        {
            "subject": "LO",
            "year": 2,
            "amount": 4
        },
        {
            "subject": "Frans",
            "year": 2,
            "amount": 3
        },
        {
            "subject": "Geschiedenis",
            "year": 2,
            "amount": 3
        },
        {
            "subject": "NaSk",
            "year": 2,
            "amount": 3
        },
        {
            "subject": "Mentorles",
            "year": 2,
            "amount": 1
        },
        {
            "subject": "Nederlands",
            "year": 3,
            "amount": 4
        },
        {
            "subject": "Engels",
            "year": 3,
            "amount": 4
        },
        {
            "subject": "Wiskunde",
            "year": 3,
            "amount": 4
        },
        {
            "subject": "Biologie",
            "year": 3,
            "amount": 2
        },
        {
```

```
            "subject": "BV",
            "year": 3,
            "amount": 2
        },
        {
            "subject": "LO",
            "year": 3,
            "amount": 4
        },
        {
            "subject": "Frans",
            "year": 3,
            "amount": 3
        },
        {
            "subject": "Geschiedenis",
            "year": 3,
            "amount": 3
        },
        {
            "subject": "NaSk",
            "year": 3,
            "amount": 3
        },
        {
            "subject": "Mentorles",
            "year": 3,
            "amount": 1
        }
    ],
    "groups": [
        {
            "name": "zhk1a",
            "year": 1,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
```

```
        },
        {
            "name": "zhk1b",
            "year": 1,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "zhv1a",
            "year": 1,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "zhv1b",
            "year": 1,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
```

```
            ],
            "lessons": []
        },
        {
            "name": "zhv1c",
            "year": 1,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "za1a",
            "year": 1,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "za1b",
            "year": 1,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
```

```
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "zg1a",
            "year": 1,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "zhv2b",
            "year": 2,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "zhv2c",
            "year": 2,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
```

```
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "za2a",
            "year": 2,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "za2b",
            "year": 2,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "za2c",
            "year": 2,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
```

```
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "za2d",
            "year": 2,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "zg2a",
            "year": 2,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "zh3a",
            "year": 3,
            "subjects": [
                "Nederlands",
```

```
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "zh3b",
            "year": 3,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "zh3c",
            "year": 3,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "za3a",
            "year": 3,
```

```
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "za3b",
            "year": 3,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        },
        {
            "name": "za3c",
            "year": 3,
            "subjects": [
                "Nederlands",
                "Engels",
                "Wiskunde",
                "Biologie",
                "BV",
                "LO",
                "Frans",
                "Geschiedenis",
                "NaSk"
            ],
            "lessons": []
        }
    ],
```

```
    "teacherInfo": [
        {
            "subject": "Nederlands",
            "prefix": "Ned",
            "amount": 3
        },
        {
            "subject": "Engels",
            "prefix": "Eng",
            "amount": 3
        },
        {
            "subject": "Wiskunde",
            "prefix": "Wisk",
            "amount": 3
        },
        {
            "subject": "Biologie",
            "prefix": "Bio",
            "amount": 3
        },
        {
            "subject": "BV",
            "prefix": "BV",
            "amount": 3
        },
        {
            "subject": "LO",
            "prefix": "LO",
            "amount": 3
        },
        {
            "subject": "Frans",
            "prefix": "Fr",
            "amount": 3
        },
        {
            "subject": "Geschiedenis",
            "prefix": "Gs",
            "amount": 3
        },
        {
            "subject": "NaSk",
            "prefix": "Nk",
            "amount": 3
        }
```

```
538         ],
539         "lessons": [],
540         "amountOfDaysAWeek": 5,
541         "amountOfHoursADay": 9
542     }
```

Bijlage F

Code

Alle code voor dit project is te vinden op de GitHub repository van dit project: https://github.com/J0eppp/pws.

Bijlage G

Logboek

G.1 Logboek Joep

Wat	**Wanneer**	**Aantal uur**
Inlezen	Voor de zomervakantie	17,5
Begin verslag	Voor de zomervakantie	4
Theorie LP lezen	Tijdens de zomervakantie	5
Begin programma in Go	dinsdag 29 juni 2021	2
Begin Python model	donderdag 1 juli 2021	3
Resultaten weergeven in tabel	vrijdag 2 juli 2021	0,5
Tests voor docent klasse toevoegen	maandag 5 juli 2021	1
Tests voor klassen klasse toegevoegd	dinsdag 6 juli 2021	0,5
Begin van ILP model	donderdag 2 september 2021	4
Begin LP solver	donderdag 23 september 2021	3
Tussenuren teller maken	maandag 27 september 2021	2
Werken aan constraints	dinsdag 28 september 2021	3
JSON parser schrijven	donderdag 30 september 2021	1
Schrijf feasible solution functie	maandag 4 oktober 2021	0,5
Bug fixes	dinsdag 5 oktober 2021	2
Geautomatiseerde tests schrijven	dinsdag 5 oktober 2021	3
Meer tests schrijven	donderdag 7 oktober 2021	1
Werken aan de LP solver	donderdag 7 oktober 2021	2
Verder werken aan de LP solver	maandag 11 oktober 2021	1,5
Bug fixes van de LP solver	zaterdag 16 oktober 2021	5
Bug fixes van de LP solver	maandag 18 oktober 2021	2
Contact ervaringsdeskundige (informatica student)	maandag 18 oktober 2021	1,5
Bug fixes van de LP solver	donderdag 21 oktober 2021	2
Lesdag Zermelo	woensdag 27 oktober 2021	7
Grafentheorie toepassen (programmeren)	zaterdag 30 oktober 2021	2
Resultaat grafentheorie opslaan	zondag 31 oktober 2021	1
Begin aan de GUI	dinsdag 2 november 2021	2
Verder werken aan de GUI	vrijdag 12 november 2021	1,5
Log lines toevoegen aan solver	woensdag 1 december 2021	0,5
Timmermanprobleem toevoegen als voorbeeld	donderdag 2 december 2021	1
Roosters opslaan als Excel bestand	vrijdag 3 december 2021	2
Random solver toevoegen	vrijdag 3 december 2021	1

G.2 Logboek Sam

Wat	**Wanneer**	**Aantal uur**
inlezen voor zomervakantie	voor zomervakantie	17,5
Inlezen Theorie	voor zomervakantie	6,5
opstellen onderzoeksvragen	voor zomervakantie	2
schrijven inleiding	voor zomervakantie	2
herschrijven theorie LP	3-dec	2
inlezen BnB	4-sep	4
kijken youtube-video's BnB	5-sep	2
schrijven theorie BnB	10-sep	2
schrijven conclusie	19-dec	3
schrijven discussie	19-dec	4
schrijven werkwijze	voor zomervakantie	2
belafspraak Coralie	13-dec	2
lesdag bij Zermelo	27-okt	7
schrijven theorie complexitiet	3-nov	3
inlezen complexiteit	12-nov	3
herschrijven theorie BnB	9-dec	2
schrijven kleinste kwadratenmethode	8-okt	2
schrijven probleemstelling	15-sep	4
opmaak verslag	21-dec	1
ik-zinnen passificeren	21-dec	1
Schrijven evaluatie	21-dec	1,5
computer gereedmaken voor berekeningen	5-dec	1,5
toevoegen bijlagen	21-dec	1
mailcontact diversen	divers	2
uitleg krijgen vakstudenkt	divers	2
Totaal		**80**

Tabel G.2: Logboek Sam