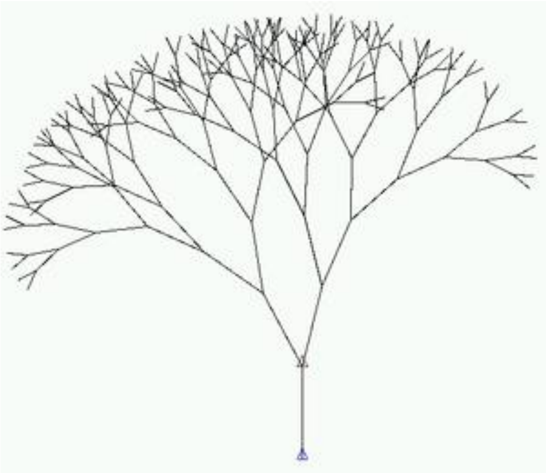
Computational thinking in een praktische opdracht voor 5 vwo wiskunde B



Joris van der Meulen, s1014447
Verslag van Onderzoek van Onderwijs richting wiskunde (10 EC variant)

Samenvatting

Computational thinking laat zich omschrijven als "het denkproces waarmee een probleem en zijn oplossing(en) zo worden geformuleerd dat deze effectief door een informatieverwerkende tussenpersoon – mens of machine – kan worden uitgevoerd". Verschillende bronnen wijzen erop dat computational thinking een belangrijk onderdeel van het toekomstige (wiskunde)onderwijs is. Hulpmiddelen die docenten kunnen ondersteunen bij het onderwijzen hierin zijn momenteel echter nog erg schaars. Computational thinking leek ons daarom een geschikt onderwerp om een praktische opdracht over te ontwikkelen, die invulling zou kunnen geven aan de werkmiddag van 5 vwo wiskunde B op het Carmel College Salland die op 28 maart 2017 plaatsvond. Dit heeft geleid tot de volgende onderzoeksvraag voor dit ontwerponderzoek:

"Hoe kunnen aspecten van computational thinking door middel van een werkmiddag worden aangeleerd aan leerlingen in 5 vwo wiskunde B?"

De voor de werkmiddag ontworpen praktische opdracht heeft het algoritme-ontwerpparadigma Branch and Bound als onderwerp. Branch and Bound is een algoritme-ontwerpparadigma dat vele toepassingen kent in de praktijk en geschikt is om puzzelachtige problemen mee op te lossen. Voor het ontwerpen van de praktische opdracht zijn vijf ontwerpeisen opgesteld die ingaan op de duur en het niveau van de praktische opdracht, de benodigde sturing van de docent, de motivatie van de leerlingen en de te behalen leerdoelen. De principes van geleid heruitvinden en relationeel begrip zijn aangewend als didactische uitgangspunten bij het ontwerpen van de praktische opdracht. Door het algoritme-ontwerpparadigma stukje bij beetje te introduceren en daarbij geen uitgewerkte voorbeelden te geven, is gepoogd de opdracht een zelfontdekkend karakter te geven.

Aan de hand van de vijf opgestelde ontwerpeisen wordt de ontworpen praktische opdracht geanalyseerd. Daarbij wordt gebruik gemaakt van data die zowel tijdens als na de werkmiddag verzameld is. Naast de uitvoering van het ontwerp op het Carmel College Salland, waarin twee 5 vwo wiskunde B klassen de praktische opdracht hebben doorgewerkt, is data verzameld door vijf leerlingen een dag na de werkmiddag te interviewen over de praktische opdracht en door een walkthrough van het ontwerp te doen met een eventuele belanghebbende. Op basis van de analyse wordt geconcludeerd dat er in het gemaakte ontwerp veel verbeteringen mogelijk zijn, maar het desalniettemin een goede basis vormt voor zowel het gebruik in de schoolpraktijk als voor verder onderzoek naar het aanleren van aspecten van computational thinking.

Over het aanleren van aspecten van computational thinking door middel van een werkmiddag aan leerlingen in 5 vwo wiskunde B wordt geconcludeerd dat het introduceren van puzzelachtige problemen en een algoritme-ontwerpparadigma dat in staat is om deze op te lossen in een praktische opdracht hier geschikt voor lijkt te zijn. Algoritmisch denken, computationele complexiteit en heuristieken zijn een aantal aspecten van computational thinking die aan bod kunnen komen. Met name bij het introduceren van heuristieken moet er echter op gelet worden dat er niet te grote stappen gemaakt worden en er een duidelijke uitleg voorhanden is; het is voor leerlingen geen makkelijk concept. Het principe van geleid heruitvinden leent zich over het algemeen goed voor praktische opdrachten, maar omdat de meeste middelbare scholieren over slechts weinig voorkennis beschikken die nuttig is voor het aanleren van aspecten van computational thinking op hun niveau, is het van belang om voldoende sturing vanuit de opdracht of de docent toe te staan. Het onderzoek sluit af met aanbevelingen voor verder onderzoek en een verbeterde versie van het ontwerp.

Inhoudsopgave

Hoofdstuk 1 - Inleiding

In het huidige en toekomstige onderwijs is steeds meer aandacht voor programmeren en ICT-vaardigheden. Nederland loopt op dit gebied flink achter ten opzichte van veel Westerse landen, maar is inmiddels begonnen aan het goedmaken van deze achterstand (Balanskat & Engelhardt, 2014; Van der Graaf, 2015; Platform Onderwijs 2032, 2015). Zo leerden in het schooljaar 2015-2016 800.000 leerlingen van basis- en middelbare scholen te programmeren onder leiding van Neelie Kroes, Startup Delta en enkele technische bedrijven (Platform Onderwijs 2032, 2015). Platform Onderwijs 2032 (2015) geeft zelf echter al aan dat het bij programmeren vaak niet om de vaardigheid van het coderen zelf gaat, maar meer om de digitale vaardigheden en de houding die programmeerlessen met zich meebrengen. Het kunnen doorzien en tot op zekere hoogte zelf kunnen maken van de mechanismen achter ICT-toepassingen door conceptueel en algoritmisch te denken is meer van belang dan kunnen coderen. Hiervoor is het noodzakelijk dat leerlingen in aanraking komen met 'computational thinking': nadenken over de vraag hoe je op algoritmische wijze een probleem kunt oplossen (Van Bruggen, Pijpers, Stiller, & Boeke, 2016). Computational thinking kan worden omschreven als "het denkproces waarmee een probleem en zijn oplossing(en) zo worden geformuleerd dat deze effectief door een informatieverwerkende tussenpersoon – mens of machine – kan worden uitgevoerd" (Wing, 2006). Het omvat abstractie, heuristieken, algoritmes, efficiëntie en complexiteit. Volgens SLO (2015) richt computational thinking zich op de "vaardigheden die essentieel zijn om problemen op te lossen waarbij veel informatie, variabelen en rekenkracht nodig zijn."

1.1 Aanleiding

Sinds een paar jaar is er in verschillende landen veel aandacht voor computational thinking in het onderwijs. In Groot-Brittanië is het sinds kort zelfs verplicht hierin te onderwijzen, maar dit beperkt zich tot het basisschoolcurriculum (Pijpers, Stiller, & Boeke, 2015). Programmeerlessen – die op sommige Nederlandse middelbare scholen al worden gegeven – kunnen bijdragen aan het ontwikkelen van bepaalde aspecten van computational thinking, maar wiskunde is hier ook geschikt voor (Weintrop, et al., 2016). Hulpmiddelen die docenten kunnen ondersteunen bij het onderwijzen in computational thinking zijn voorlopig echter bijzonder schaars. Dat computational thinking ook in Nederland een actueel onderwerp in het (wiskunde)onderwijs is, blijkt wel uit het curriculum van de toekomst van de Stichting Leerplan Ontwikkeling (SLO, 2015) en het onderzoek dat zij (laten) verrichten naar zowel de manieren om op computational thinking gerichte leerdoelen expliciet te verwezenlijken in het wiskundeonderwijs als naar de mening van wiskundedocenten over computational thinking in hun onderwijs (Aalders, Velthuizen, & Tolboom, 2017; Aalders & Velthuizen, 2017).

Alhoewel praktische opdrachten op de middelbare school sinds 2007 niet meer verplicht zijn (Van der Zanden, 2007), is er op veel scholen nog wel ruimte voor. Praktische opdrachten bij het schoolvak wiskunde zijn een leuke manier om leerlingen uit te dagen, hen op een andere manier met wiskunde bezig te laten zijn en/of hen in aanraking te laten komen met wiskundige onderwerpen waar zij gedurende hun middelbareschooltijd niet in aanraking mee zouden komen als zij enkel de stof zouden behandelen die nodig is om het centraal eindexamen door te komen. Omdat computational thinking naar verwachting een erg belangrijk onderdeel zal zijn van het toekomstige (wiskunde)onderwijs, maar hulpmiddelen die docenten kunnen ondersteunen bij het onderwijzen hierin met name op middelbare schoolniveau schaars zijn, is dit een geschikt onderwerp voor zo'n praktische opdracht.

Net als op veel andere scholen, is op het Carmel College Salland (CCS) te Raalte in het programma van toetsing en afsluiting van het schoolvak wiskunde ruimte voor praktische opdrachten. Het is op het CCS gebruikelijk om op ieder niveau in iedere jaarlaag waarin schoolexamens worden afgenomen de leerlingen een praktische opdracht uit te laten voeren aan de hand van een werkblad of opdrachtenboekje. De leerlingen krijgen een dag of middag de tijd om dit werkblad of opdrachtenboekje door te werken waarbij zij begeleid worden door een of meer docenten. Het maken van de praktische opdracht gebeurt over het algemeen in groepjes van twee of vier leerlingen.

Op 28 maart 2017 was op het CCS een werkmiddag gepland voor 5 vwo wiskunde B waar de leerlingen aan een praktische opdracht zouden werken. Het creëren van deze praktische opdracht en derhalve het vormgeven van de werkmiddag, was een aanleiding voor dit onderzoek. Het belang van computational thinking enerzijds en het gebrek aan onderwijsmateriaal op dat gebied anderzijds, was een andere aanleiding voor dit onderzoek. Deze twee aanleidingen hebben ons tot het doel van dit onderzoek gebracht: het ontwerpen van een praktische opdracht die geschikt is voor de werkmiddag op het CCS waarin leerlingen kennis maken met verschillende aspecten van computational thinking.

1.2 Onderzoeksvraag

Naar aanleiding van het hierboven beschreven doel van dit onderzoek hebben wij de volgende onderzoeksvraag geformuleerd:

"Hoe kunnen aspecten van computational thinking door middel van een werkmiddag worden aangeleerd aan leerlingen in 5 vwo wiskunde B?"

Om deze onderzoeksvraag te beantwoorden, hebben wij in dit onderzoek een praktische opdracht voor leerlingen in 5 vwo wiskunde B ontworpen over het onderwerp Branch and Bound. Branch and Bound kent vele toepassingen en is het meestgebruikte algoritme-ontwerpparadigma om complexe problemen in de discrete optimalisatie op te lossen (Clausen, 1999). Onze keuze voor Branch and Bound en de werking van dit algoritme-ontwerpparadigma worden toegelicht in Hoofdstuk 2.

1.3 Leeswijzer

Een ontwerponderzoek in de onderwijspraktijk omvat twee onderzoeksfasen: een vooronderzoek en een innovatiecyclus (Van der Donk & Van Lanen, 2016). In de eerste fase van ons ontwerponderzoek hebben wij richting gegeven aan de praktische opdracht door een geschikt onderwerp te vinden en ontwerpeisen vast te stellen waar ons ontwerp aan moet voldoen. In Hoofdstuk 2 wordt het onderwerp van de praktische opdracht, Branch and Bound, uiteengezet en toegelicht. In dat hoofdstuk is ook aandacht voor de didactische visie waar ons ontwerp bij aansluit. Onze ontwerpeisen worden vervolgens besproken in Hoofdstuk 3.

In de tweede fase van ons ontwerponderzoek hebben wij de praktische opdracht gemaakt, getest en geanalyseerd om ten slotte tot een verbeterde versie te komen. In Hoofdstuk 4 wordt de methode van ons onderzoek toegelicht. De resultaten van ons onderzoek, bestaande uit de ontworpen praktische opdracht en een analyse hiervan, zijn te vinden in Hoofdstuk 5. Deze analyse gebeurt aan de hand van een uitvoering van ons ontwerp in de onderwijspraktijk, afgenomen leerlingeninterviews en een walkthrough van ons ontwerp met een eventuele belanghebbende. In Hoofdstuk 6 sluiten wij dit onderzoek af met conclusies en aanbevelingen voor verder onderzoek. Een verbeterde versie van het ontwerp is onderdeel van onze aanbevelingen.

Hoofdstuk 2 - Theoretisch kader

Dit hoofdstuk geeft de richting van ons ontwerponderzoek weer en hoe dit aansluit op de literatuur. In Sectie 2.1 wordt ingegaan op Branch and Bound, het algoritme-ontwerpparadigma dat het onderwerp van de door ons ontworpen praktische opdracht vormt. Sectie 2.2 beschrijft de didactische visie waar ons ontwerp bij aansluit.

2.1 Branch and Bound

Zoals naar voren is gekomen in de inleiding van dit verslag, was het belang van computational thinking enerzijds en het gebrek aan onderwijsmateriaal op dat gebied anderzijds een aanleiding voor ons onderzoek. Wij wilden een praktische opdracht ontwerpen voor een werkmiddag waarin leerlingen in aanraking in komen met verschillende aspecten van computational thinking zoals abstractie, heuristieken, algoritmes, efficiëntie en complexiteit. Puzzelachtige problemen werken motiverend en helpen studenten op een abstract niveau algoritmisch te laten denken (Levitin & Papalaskari, 2002). Om deze reden hebben wij besloten de leerlingen in onze praktische opdracht kennis te laten maken met een algoritme-ontwerpparadigma dat gebruikt kan worden om verschillende puzzelachtige problemen op te lossen.

Het algoritme-ontwerpparadigma dat wij gekozen hebben is ‘Branch and Bound’ (Land & Doig, 1960). Branch and Bound kent vele toepassingen en is het meestgebruikte algoritme-ontwerpparadigma om complexe discrete optimalisatieproblemen op te lossen (Clausen, 1999). Het idee achter het algoritme-ontwerpparadigma Branch and Bound is om de totale verzameling van mogelijke oplossingen van een optimaliseringsprobleem, die elk overeenkomen met een bepaalde oplossingswaarde, op te delen in steeds kleinere deelverzamelingen en gedurende dit opdelen steeds bepaalde deelverzamelingen uit te sluiten voor verder onderzoek, waardoor maar een deel van alle mogelijke oplossingswaarden uitgerekend hoeft te worden om een optimale oplossing te verkrijgen. Hoe dit opdelen van de totale verzameling van mogelijke oplossingen van een optimaliseringsprobleem in steeds kleinere deelverzamelingen moet gebeuren, is sterk probleemafhankelijk en niet vastgelegd in het algoritme-ontwerpparadigma. Er kunnen voor een bepaald probleem ook verschillende geschikte manieren van opdelen gevonden worden. Gebruikelijk is het om een optimalisatieprobleem om te schrijven naar een (mixed) integer linear programming-probleem voordat Branch and Bound toegepast wordt en dan een bekende methode van opdelen uit de literatuur toe te passen zoals de methode van Balas, de methode van Dakin of het algoritme van Little (Papadimitriou & Steiglitz, 1982; Winston, 2004).

Het uitsluiten van bepaalde deelverzamelingen voor verder onderzoek in Branch and Bound gebeurt op basis van onder- en bovengrenzen. De uitleg in deze alinea is gebaseerd op minimaliseringsproblemen; voor maximaliseringsproblemen werkt het analoog. Elke deelverzameling van mogelijke oplossingen die gecreeërd wordt, komt overeen met een bepaalde ondergrens die zo bepaald wordt dat geen enkele mogelijke oplossing in die deelverzameling overeen kan komen met een lagere (dus betere) oplossingswaarde dan deze ondergrens. De bovengrens van Branch and Bound is de laagste oplossingswaarde die tot dan toe gevonden is of, zolang er nog geen lagere oplossingswaarde gevonden is, gelijk aan een heuristisch gevonden beginwaarde. Als voor een bepaalde deelverzameling geldt dat de met deze deelverzameling overeenkomende ondergrens hoger is dan de Branch and Bound-bovengrens, kan deze deelverzameling uitgesloten worden voor verder onderzoek. Er zal namelijk nooit een betere oplossing gevonden worden in deze deelverzameling dan de oplossing die we al gevonden hebben. Het bepalen van ondergrenzen voor deelverzamelingen is, net als de manier van het opdelen in deelverzamelingen, sterk probleemafhankelijk en niet vastgelegd in het algoritme-ontwerpparadigma. Als een optimaliseringsprobleem omgeschreven is naar een (mixed) integer linear programming-probleem, kunnen de ondergrenzen bepaald worden door het uitrekenen van LP-relaxaties. Voor een efficiënte werking van Branch and Bound is het noodzakelijk dat de ondergrenzen zo hoog mogelijk, maar uiteraard nog wel kloppend, zijn.

Onze keuze voor het algoritme-ontwerpparadigma Branch and Bound brengt een probleem met zich mee. Om Branch and Bound toe te passen op een optimalisatieprobleem, is het namelijk zowel handig als gebruikelijk om dit probleem eerst om te schrijven naar een (mixed) integer linear programming-probleem. Lineair programmeren is echter geen voorkennis voor de meeste leerlingen in 5 vwo wiskunde B en het is evenmin een onderwerp dat de leerlingen in korte tijd bij te brengen is als introductie op de werkmiddag. Dit probleem hebben wij opgelost door voorbeeldtoepassingen van Branch and Bound te selecteren waarop het algoritme-ontwerpparadigma redelijk direct toe te passen is, waardoor het omschrijven van de problemen naar (mixed) integer linear programming-problemen niet per se nodig is. Voorbeeldtoepassingen die ons hiervoor geschikt leken zijn planningsproblemen waarin de maximale vertraging geminimaliseerd moet worden en toewijzingsproblemen (Ross & Soland, 1975; Zdrzaka & Grabowski, 1989; Stein, 2003; Ozgunsur, 2013).

2.2 Didactiek

Voor het ontwikkelen van wiskundig lesmateriaal is het van belang voor de geest te hebben bij welke didactische visie je aan wilt sluiten. Het is namelijk niet zo dat er een eenduidige, algemeen correcte visie bestaat. Honderd jaar geleden was dat al zo, toen Ehrenfest en Dijksterhuis tegenover elkaar stonden. Ehrenfest-Afanassjeewa (1924) pleitte ervoor te focussen op inzicht en intuïtie en om stof aan te laten sluiten op de realiteit. Zij vond dat de constructie van kennis belangrijker is dan reproductie. Dijksterhuis (1924) aan de aan de andere kant was van mening dat intuïtief redeneren het begrip en de correcte toepassing van wiskunde door de leerlingen in de weg zou kunnen staan en pleitte voor “geestelijke tucht, orde en reinheid, die de mathesis nastreeft” (Dijksterhuis, 1924, p. 11).

Er is een parallel te trekken tussen Ehrenfest en Dijksterhuis en de meer recente tegen over elkaar staande visies. De visie van Ehrenfest laat zich vertalen naar het realistisch reken-wiskundeonderwijs, waarin leerlingen problemen moeten oplossen aan de hand van eigen strategieën en inzichten waarbij uitgegaan wordt van voor de leerlingen voorstelbare contextsituaties ter zingeving en als startpunten voor wiskundig redeneren (Heege, 2008). Karakteristieken van realistisch reken-wiskundeonderwijs zijn het gebruik van contexten, het gebruik van modellen, schema’s en symbolen als hulpmiddelen, belang hechten aan eigen constructies en producties van leerlingen, interactiviteit en het verstrengelen van leerstofgebieden (Verbruggen, Frickel, Van Hell, & Boswinkel, 2007). In het realistisch reken-wiskundeonderwijs staat geleid heruitvinden centraal: de leerlingen zelf voor hen nieuwe wiskunde laten ontdekken op basis van reeds bestaande voorkennis waarbij de docent de rol van coach aanneemt (Freudenthal, 1991). Tegenover het realistisch reken-wiskundeonderwijs staat het functioneel reken-wiskundeonderwijs, waarin traditionele rekenalgoritmen centraal staan en er veel geoefend wordt zonder gebruik te maken van context (Van de Craats, 2007). Het functioneel reken-wiskundeonderwijs kan gezien worden als een voortzetting van het traditionele wiskundeonderwijs van Dijksterhuis met als verschil dat er wel ruimte is voor contexten ter verdieping van de aangeleerde stof.

De Koninklijke Nederlandse Akademie van Wetenschappen heeft geconcludeerd dat realistisch leren rekenen ten opzichte van leren rekenen op een traditionele manier geen verschil teweegbrengt in het uiteindelijke niveau van leerlingen, op de bassischool tenminste (KNAW, 2009). De Commissie Toekomst WiskundeOnderwijs heeft echter geconcludeerd dat door de contextrijke aanpak van de afgelopen tijd wiskunde verworden is tot een van de meest populaire schoolvakken en adviseert voor het voortgezet onderwijs een didactische aanpak die aansluit bij het realistisch wiskundeonderwijs (CTWO, 2007). Wij concluderen dat dit in het bijzonder geldt voor praktische opdrachten in het wiskundeonderwijs op de middelbare school, aangezien hierbij vaak een realistische situatie die aanleiding geeft tot een probleemstelling het uitgangspunt is en er vaardigheden aangeleerd worden die los staan van de rest van het wiskundecurriculum op de middelbare school (De Haan, 2001).

Skemp (1977) beschrijft instrumenteel versus relationeel begrip in het wiskundeonderwijs. Instrumenteel begrip komt neer op het kennen en kunnen toepassen van wiskundige regels. Het verschilt van relationeel begrip in de zin dat waar de regels vandaan komen en waarom deze werken niet begrepen hoeft te worden zolang ze maar gebruikt kunnen worden. Voordelen van instrumenteel begrip ten opzichte van relationeel begrip zijn dat het over het algemeen eenvoudiger te begrijpen is, er sneller kortetermijndoelen mee bereikt kunnen worden en dat het zich in de meeste opgaven ertoe leent om makkelijk tot een goed antwoord komen. Voordelen van relationeel begrip ten opzichte van instrumenteel begrip aan de andere kant zijn dat het beter toepasbaar is op nieuwe situaties, het eenvoudiger te onthouden is en dat het leuk kan zijn om iets echt begrijpen en niet alleen toe te kunnen passen.

Voor de praktische opdracht vinden wij de voordelen van relationeel begrip zwaarder wegen dan die van instrumenteel begrip. Leerlingen moeten niet een stappenplan leren toepassen maar echt begrip krijgen van hoe en waarom Branch and Bound werkt en wat het nut ervan is. Om het kweken van relationeel begrip bij leerlingen te maximaliseren, zijn drie aspecten van belang (Sapire & Mays, 2008). Allereerst is het belangrijk hen aan het werk te zetten met wiskundige activiteiten die zich lenen tot zelf onderzoeken en ontdekken. Dit sluit aan bij de visie van realistisch wiskundeonderwijs en het geleid heruitvinden. Daarnaast is het belangrijk dat activerend materiaal beschikbaar is, bijvoorbeeld in de vorm van spelletjes of puzzeltjes, zodat de leerlingen kunnen engageren met de wiskundige taken. Tot slot is voor het maximaliseren van relationeel begrip interactie tussen leerlingen onderling en tussen leerlingen en docenten van belang.

Hoofdstuk 3 - Ontwerpeisen

Ontwerpeisen zijn een belangrijk onderdeel van de eerste onderzoeksfase in een ontwerponderzoek. Goede ontwerpeisen geven inzicht in het doel van het ontwerp, in belangrijke eigenschappen van het ontwerp en in hoe het ontwerp ontwikkeld of uitgevoerd moet worden (Nieveen & Folmer, 2013). Voor ons ontwerponderzoek hebben wij vijf ontwerpeisen opgesteld, die wij hieronder introduceren en toelichten. Een analyse van ons ontwerp waarin wij nagaan of aan deze ontwerpeisen voldaan is, is te vinden in Hoofdstuk 5.

Ontwerpeis 1: De praktische opdracht is in ongeveer drie uur door de leerlingen af te ronden

Bij het ontwerpen van praktische opdrachten is het noodzakelijk duidelijke deadlines op te stellen die voor de leerlingen haalbaar zijn en waar je je als docent ook aan houdt (De Haan, 2001). Voor werkmiddag op het CCS staat drie uur, maar er is eventueel ruimte voor een klein beetje uitloop. De praktische opdracht moet idealiter zo ontworpen worden dat er geen leerlingen zijn die er eigenlijk veel meer dan drie uur voor nodig hebben maar ook geen leerlingen die er veel minder dan drie uur voor nodig hebben om hem af te ronden. Om te testen of aan deze ontwerpeis voldaan is, hebben wij bij het uitvoeren van ons ontwerp genoteerd wanneer de groepjes leerlingen klaar zijn. Daarnaast hebben wij een vraag in de leerlingeninterviews, die wij een dag na de werkmiddag afgenomen hebben, gericht op de duur van de praktische opdracht.

Ontwerpeis 2: De praktische opdracht sluit goed aan bij het niveau van leerlingen in 5 vwo wiskunde B

De praktische opdracht die wij ontwerpen zal door veel leerlingen gemaakt worden en wij willen dat al deze leerlingen door de opdracht uitgedaagd worden maar zich er niet door ontmoedigd voelen. Dit is lastig te meten, maar wat wij ook verstaan onder deze ontwerpeis is dat de gehaalde cijfers een goede weerspiegeling moeten zijn van het niveau van de leerlingen. Bij het nakijken van de opdracht hebben wij getest of dit geslaagd is door uit te rekenen hoe er door de leerlingen gescoord is op de verschillende vragen in de praktische opdracht. Ook hebben wij een vraag in de leerlingeninterviews gericht op het niveau van de praktische opdracht.

Ontwerpeis 3: De praktische opdracht staat op zichzelf, in de zin dat er weinig sturing van de docent nodig is

De praktische opdracht krijgt, zoals gebruikelijk is op het CCS, de vorm van een opdrachtenboekje dat de leerlingen in groepjes door zullen werken. Doordat er veel leerlingen tegelijk met de opdracht aan de gang zullen gaan, is het niet handig als zij hierbij veel sturing van de docent nodig hebben. De uitleg en de vragen in de praktische opdracht moeten dus duidelijk verwoord zijn en de vragen mogen niet dermate lastig zijn dat leerlingen vaak vastlopen. Bij het uitvoeren van ons ontwerp hebben wij getest of aan deze ontwerpeis voldaan is door te letten op de hoeveelheid vragen die de leerlingen stelden.

Ontwerpeis 4: De praktische opdracht is in staat de leerlingen intrinsiek te motiveren

Doordat de resultaten van de werkmiddag onderdeel zijn van het schoolexamencijfer van de leerlingen, zullen zij hiervoor extrinsiek gemotiveerd zijn. Wij willen de leerlingen met de werkmiddag echter ook intrinsiek motiveren, voor de praktische opdracht zelf in het bijzonder en voor het vak wiskunde in een meer algemene zin. Onderzoek wijst uit dat intrinsieke motivatie een belangrijke voorspeller is van succes op school (Taylor et al., 2014) en het belang hiervan moet dus niet onderschat worden. Door te letten op het enthousiasme van de leerlingen tijdens het uitvoeren van ons ontwerp, hebben wij getest of aan deze ontwerpeis voldaan is. Ook hebben wij leerlingen een dag na de werkmiddag in onze interviews gevraagd of zij de werkmiddag leuk vonden.

Ontwerpeis 5: In de praktische opdracht worden de opgestelde leerdoelen behaald

Om ervoor te zorgen dat verschillende aspecten van computational thinking in onze praktische opdracht aan bod komen, hebben wij een drietal leerdoelen opgesteld. Bij het ontwerpen van de praktische opdracht hebben wij rekening gehouden met deze leerdoelen en aan de hand van de leerlingeninterviews hebben wij getest of ze behaald zijn. De drie leerdoelen zijn als volgt.

Leerdoel 1: De leerling heeft een intuïtief begrip van computationele complexiteit en daarmee van het nut van het algoritme-ontwerppardigma Branch and Bound.

Een belangrijk onderdeel van computational thinking is computationele complexiteitstheorie (National Research Council, 2010). In zowel de computationele complexiteitstheorie als in de wiskunde is NP-moeilijkheid en in het bijzonder NP-volledigheid een veelbesproken onderwerp. Of al dan niet geldt dat P = NP is een van de zeven millenniumprijsprobleem en wordt algemeen beschouwd als een van de belangrijkste open problemen in de wiskunde en het belangrijkste probleem in de theoretische informatica (Fortnow, 2009). Het niveau van dit onderwerp is niet geschikt voor leerlingen in 5 vwo wiskunde en geen enkele leek zou er in een werkmiddag een goed begrip van kunnen kweken. Het zou echter wel mooi zijn als de leerlingen een intuïtief begrip krijgen hiervan, en daarmee het nut van het algoritme-ontwerppardigma Branch and Bound inzien. Door een NP-volledig probleem te selecteren en de leerlingen uit te laten vinden hoeveel tijd het kost om zelfs relatief kleine instanties van dat probleem met behulp van brute force uit te rekenen, hebben wij geprobeerd dit leerdoel te behalen. Door in de leerlingeninterviews te vragen naar waarom Branch and Bound werd toegepast op het probleem in de praktische opdracht, hebben wij getest of dit leerdoel behaald is.

Leerdoel 2: De leerling kan omschrijven wat een heuristiek is.

Heuristieken zijn een ander belangrijk onderdeel van computational thinking (National Research Council, 2010). Onder heuristieken in de informatica worden zoekalgoritmen verstaan die algemene intuïtieve methoden gebruiken om binnen afzienbare tijd een oplossing te vinden voor een probleem. Deze oplossing is over het algemeen geen optimale oplossing van het probleem en hoeft, afhankelijk van de gebruikte heuristiek en de instantie van het probleem dat ermee opgelost wordt, zelfs geen goede oplossing te zijn. Door leerlingen een heuristiek toe te laten passen op een probleem, de zwakte van deze heuristiek te laten ontdekken door het creëren van een specifieke probleeminstantie en zelf een heuristiek te laten bedenken, hebben wij geprobeerd dit leerdoel te behalen. Door in de leerlingeninterviews te vragen naar een voor- en een nadeel van heuristieken, hebben wij getest of dit leerdoel behaald is.

Leerdoel 3: De leerling kan met behulp van Branch and Bound op systematische wijze de optimale oplossing van een voorbeeldprobleem vinden.

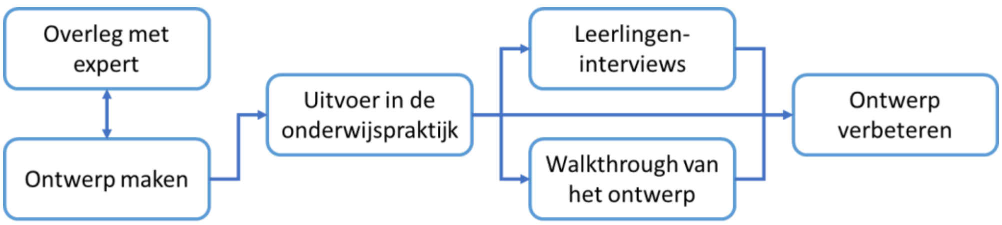
Het onderwerp van de praktische opdracht is Branch and Bound. Wij laten de leerlingen in de praktische opdracht aan de hand van een voorbeeldtoepassing stap voor stap zelf uitvinden hoe Branch and Bound werkt, waartoe zij algoritmisch moeten denken. Als voorbeeldtoepassing hebben wij planningsproblemen waarin de maximale vertraging geminimaliseerd moet worden geselecteerd (Zdrzalka & Grabowski, 1989). De leerlingen vinden in de praktische opdracht stukje bij beetje uit hoe Branch and Bound werkt waarbij zij aangespoord worden zelf te ontdekken hoe de stukken met elkaar samenhangen en waarom het algoritme-ontwerpparadigma werkt. Er worden in de praktische opdracht geen voorbeelden uitgewerkt en de leerlingen oefenen met te weinig probleemsituaties om te automatiseren. Hierdoor zijn wij van mening dat leerlingen duidelijk relationeel begrip van het algoritme hebben opgedaan als zij Branch and Bound een dag na de werkmiddag toe kunnen passen op een voorbeeldprobleem. Wij hebben getest of dit leerdoel behaald is door aan het eind van de leerlingeninterviews de leerlingen te confronteren met een planningsprobleem en hen te vragen de optimale oplossing hiervan te vinden met behulp van Branch and Bound.

Hoofdstuk 4 - Methode

In dit hoofdstuk bespreken wij de onderzoeksmethode van ons ontwerponderzoek. In Sectie 4.1 wordt de onderzoeksopzet besproken. In Sectie 4.2 komen de respondenten aan bod die een rol hebben gespeeld in het testen van ons ontwerp. De manieren waarop data verzameld is worden toegelicht in Sectie 4.3.

4.1 Onderzoeksopzet

In ons ontwerponderzoek hebben wij allereerst het ontwerp, bestaande uit de praktische opdracht en een bijbehorend correctievoorschrift, gemaakt. Dit was een iteratief proces, waarin wij regelmatig een versie voorlegden of ideeën bespraken met een expert op het gebied van ons ontwerp. De praktische opdracht en het bijbehorende correctievoorschrift zijn te zien in Bijlagen A en B. Om vast te stellen of ons ontwerp voldeed aan de ontwerpeisen en om te controleren of de leerdoelen waren behaald, hebben wij ons ontwerp op verschillende manieren getest. Allereerst hebben wij ons ontwerp uitgevoerd in de onderwijspraktijk. Dit komt neer op het begeleiden van de werkmiddag van 5 vwo wiskunde B op het Carmel College Salland en het nakijken van de tijdens die werkmiddag door de leerlingen gemaakte praktische opdrachten. Vervolgens hebben wij, een dag na de werkmiddag, een aantal leerlingen geïnterviewd over de praktische opdracht. Tot slot hebben wij een walkthrough gedaan van ons ontwerp met een eventuele belanghebbende (Van der Donk & Van Lanen, 2016). Deze onderzoeksopzet is schematisch weergegeven in Figuur 1.



Figuur 1: Schematische weergave van de onderzoeksopzet

4.2 Respondenten

In ons onderzoek is er sprake van drie typen respondenten. Allereerst is er de expert die ons geholpen heeft in het ontwerpproces. Deze expert is Mark Timmer, zelf wiskundedocent op een middelbare school en universitair vakdidacticus. Hij heeft veel ervaring met het ontwerpen van leermateriaal voor wiskundelessen op de middelbare school. Dan is er de eventuele belanghebbende bij ons ontwerp met wie de walkthrough is gedaan. Deze eventuele belanghebbende is Léon Klunder, een beginnende wiskundedocent die expert is op het gebied van Branch and Bound. Het belang dat hij heeft bij ons ontwerp is dat hij het wellicht zelf in zijn onderwijspraktijk kan toepassen. Tot slot zijn er de leerlingen die betrokken waren bij de uitvoering van ons ontwerp in de onderwijspraktijk en bij de leerlingeninterviews.

Op 28 maart 2017 was de werkmiddag van 5 vwo wiskunde B op het Carmel College Salland. Twee 5 vwo wiskunde B klassen, bestaande uit in totaal 52 leerlingen, hebben de door ons ontworpen praktische opdracht doorgewerkt. De man/vrouw samenstelling van de twee klassen samen was ongeveer 60/40. Het doorwerken van de praktische opdracht moest in groepjes van twee gebeuren. Door de afwezigheid van drie leerlingen gebeurde dit echter in één groepje van drie en verder in groepjes van twee. De drie afwezige leerlingen hebben de praktische opdracht op een later moment als groepje van drie ingehaald. Hun werk is niet meegenomen in de analyse van ons onderzoek.

Voor de interviews die wij een dag na de praktische opdracht afnamen, hebben wij vijf leerlingen geselecteerd. Om met slechts vijf leerlingen een zo goed mogelijk beeld te krijgen van de gehele klas, hebben wij deze vijf leerlingen geselecteerd op basis van hun wiskundecijfers zodanig dat zij respectievelijk het 10^e, het 30^e, het 50^e, het 70^e en het 90^e percentiel van de klas vormen. Het was mogelijk dit te doen en gelijktijdig recht te doen aan de man/vrouw samenstelling van de klassen. Van de vijf geïnterviewde leerlingen waren er namelijk drie man en twee vrouw.

4.3 Dataverzameling

Dataverzameling heeft in drie stadia van ons onderzoek plaatsgevonden. Allereerst was er de uitvoering van het ontwerp. De uitvoering van het ontwerp bestond uit de werkmiddag op het Carmel College Salland en het nakijken van de gemaakte praktische opdrachten. Tijdens de werkmiddag hebben wij de eindtijden genoteerd waarop de groepjes leerlingen hun werk hebben ingeleverd. Daarnaast hebben wij gelet op de hoeveelheid vragen die de leerlingen stelden door vast te stellen of de docenten het erg rustig, rustig, niet rustig noch druk, druk of zeer druk hadden. Tijdens het nakijken van de gemaakte praktische opdrachten hebben wij de p'-waarde en de r_{it}-waarde van elke vraag berekend. De p'-waarde is een percentage tussen 0 en 100 dat de relatieve score op een vraag en daarmee de moeilijkheidsgraad ervan weergeeft en wordt als volgt berekend: $\mathrm{p}' = \frac{\text{gem.score}}{\text{max.score}} \cdot 100\%$ (Goldebeld, 1992). De r_{it}-waarde is de correlatie tussen de score van een vraag en de totaalscore van een toets, inclusief de score op de vraag zelf (Goldebeld, 1992). Een hoge r_{it}-waarde geeft aan dat dezelfde leerlingen die goed hebben gescoord op de gehele praktische opdracht ook goed hebben gescoord op de vraag kwestie. De eindtijden, de p'-waardes en de r_{it}-waardes gebruiken wij bij de analyse van ons ontwerp in Hoofdstuk 5.

Het tweede stadium van ons onderzoek waarin dataverzameling plaats heeft gevonden, was de afname van de leerlingeninterviews. De interviews die wij af hebben genomen bij de leerlingen zijn semigestructureerde interviews geweest, een veelgebruikte vorm van data-analyse in kwalitatief onderzoek (Edwards & Holland, 2013). Wij hebben voor dit type interview gekozen omdat het ons, ten opzichte van het afnemen van gestructureerde interviews, in staat stelt dieper in het onderwerp te duiken en antwoorden die leerlingen geven goed te begrijpen (Harrell & Bradley, 2009). Een semigestructureerd interview houdt in dat er van tevoren een richtlijn wordt opgesteld met vragen en onderwerpen die sowieso aan bod moeten komen, maar dat doorvragen door de onderzoeker toegestaan is. Of het in semigestructureerde interviews toegestaan is om als onderzoeker de volgorde aan te passen waarin de verschillende vragen en onderwerpen aan bod komen, is een aspect waar de literatuur niet eenduidig over is (Harrell & Bradley, 2009; Edwards & Holland, 2013). Omdat semigestructureerde interviews afnemen toch al een controversiële methode van data verzamelen is (Harrell & Bradley, 2009), hebben wij in onze interviews zo veel mogelijk een vaste volgorde aangehouden. De richtlijn die wij hebben gebruikt bij het afnemen van de semigestructureerde interviews naar aanleiding van onze praktische opdracht is te vinden in Bijlage C.

De interviews die wij bij de leerlingen afnamen, hebben wij opgenomen met een soundrecorder en op een later moment getranscribeerd. De getranscribeerde interviews zijn te vinden in Bijlage D. Deze transcripties nemen wij mee in de analyse van ons ontwerp in Hoofdstuk 5.

Om te voorkomen dat de leerlingen die wij geselecteerd hadden voor de interviews extra hun best zouden doen op de praktische opdracht, kregen zij vooraf niet te weten dat zij geïnterviewd zouden worden. Om ons onderzoek toch uit te voeren conform de ethische richtlijnen voor wetenschappelijk onderzoek, hebben wij alle ouders/verzorgers ruim voor de praktische opdracht plaats zou vinden een brief gestuurd waarin medegedeeld werd dat hun zoon of dochter naar aanleiding van de praktische opdracht mogelijk geïnterviewd kon worden. De brief die wij naar de ouders gestuurd hebben, is te vinden in Bijlage G.

In de brief die wij naar de ouders gestuurd hebben, hebben wij aangegeven dat de interviews met de leerlingen opgenomen worden en dat wat de leerlingen zeggen tijdens de interviews uiteraard geen invloed heeft op hun beoordeling. Verder hebben wij erin vermeld dat de opnamen getranscribeerd worden en dat alles geanonimiseerd en vertrouwelijk behandeld zal worden. De ouders kregen de mogelijkheid bezwaar te maken tegen een eventuele deelname van hun zoon of dochter aan een interview door dat kenbaar te maken via e-mail of door hun zoon of dochter een bij de brief toegevoegd strookje ingevuld aan hun wiskundedocent te laten overhandigen. Wat we ook hebben aangegeven in de brief en wat wij de leerlingen ook duidelijk hebben gemaakt voor aanvang van de interviews, is dat zij op ieder moment zelf mogen besluiten niet (verder) deel te nemen aan het onderzoek. Er waren geen ouders die bezwaar hadden tegen een eventuele deelname van hun zoon of dochter aan het onderzoek en er zijn ook geen leerlingen geweest die besloten om niet deel te nemen aan of zich terugtrokken uit een interview.

Tot slot is er de walkthrough van ons ontwerp met een eventuele belanghebbende. Tijdens dit stadium van ons onderzoek hebben wij data verzameld door de eventuele belanghebbende bij ons ontwerp de praktische opdracht te laten maken en daarbij in het opdrachtenboekje zelf opvallende punten te laten noteren of aanstippen. Vervolgens hebben wij samen met de eventuele belanghebbende bij ons ontwerp deze punten doorgelopen en erover gediscussieerd. Belangrijke punten hebben wij verder uitgewerkt in een verslag. Dit verslag is te vinden in Bijlage H. Wij gebruiken punten uit dit verslag bij het verbeteren van ons ontwerp in Hoofdstuk 6.

Hoofdstuk 5 - Resultaten

In dit hoofdstuk worden de resultaten van ons onderzoek besproken. Allereerst wordt het ontwerp toegelicht. Dit gebeurt in Sectie 5.1. In Sectie 5.2 analyseren wij vervolgens het ontwerp door na te gaan of het aan de ontwerpeisen voldoet die in Hoofdstuk 3 zijn opgesteld.

5.1 Ontwerp

In deze sectie presenteren wij ons ontwerp. Ons ontwerp bestaat uit een praktische opdracht met regels omtrent de afname hiervan en een bijbehorend correctievoorschrift. De regels omtrent de afname van de praktische opdracht hebben wij genoteerd in de inleiding van de opdracht zelf, zodat deze ook voor de leerlingen duidelijk zijn en zij er op ieder moment van de werkmiddag nog eens naar terug kunnen kijken. De ontwikkelde praktische opdracht is te vinden in Bijlage A en het bijbehorende correctievoorschrift is te vinden in Bijlage B.

Zoals omschreven in Hoofdstuk 2 vinden wij geleid heruitvinden en relationeel begrip geschikte uitgangspunten voor de ontwikkeling van een praktische opdracht. Het als leerling zelf ontdekken van voor hen nieuwe wiskunde op basis van bestaande voorkennis staat centraal in de theorie van geleid heruitvinden en is ook belangrijk voor het kweken van relationeel begrip (Skemp, 1977; Freudenthal, 1991). Aangezien leerlingen in 5 vwo wiskunde B over het algemeen echter over slechts weinig voorkennis beschikken die nuttig kan zijn voor het aanleren van aspecten van computational thinking op hun niveau, hebben wij ervoor gekozen een aanzienlijke hoeveelheid sturing in de opdracht toe te voegen. Leerlingen moeten immers wel aan de slag kunnen en leerdoelen moeten behaald worden. Door het algoritme-ontwerpparadigma stukje bij beetje te introduceren en daarbij geen uitgewerkte voorbeelden te geven, hebben wij gepoogd de opdracht toch een zelfontdekkend karakter te geven en zo enigszins aan te sluiten bij de theorie van geleid heruitvinden. Leerlingen worden aangespoord na te denken over hoe de verschillende stukjes met elkaar samenhangen en waarom het algoritme-ontwerpparadigma werkt. Gedurende de praktische opdracht komen de leerlingen bovendien verscheidene puzzelachtige opgaven tegen, die duidelijk vragen om relationeel begrip.

Hieronder lichten wij het ontwerp van de praktische opdracht toe door commentaar toe te voegen in de opdracht zelf. Alle opmerkingen die niet in de praktische opdracht zelf voorkomen maar die tot de toelichting van ons ontwerp behoren, worden in het groen weergegeven. Grotendeels geldt dat het commentaar van toepassing is op de tekst voorafgaand aan dat commentaar. Om de praktische opdracht te zien zonder commentaar, precies zoals de leerlingen deze ook te zien kregen tijdens de uitvoering van ons ontwerp, verwijzen wij de lezer naar Bijlage A.

De praktische opdracht begint met een voorblad waarop de doelgroep, de titel en de afnamedatum worden weergegeven. Verder staat er een grote versie op van de die te zien is op de voorpagina van dit verslag. Aangezien het onderwerp geheel onbekend is voor de leerlingen, zal het hen aanvankelijk niet duidelijk zijn wat het idee achter deze afbeelding is. Wij zorgen ervoor dat dit gedurende de opdracht duidelijk wordt. De volgende pagina van de praktische opdracht bevat een inleiding op de werkmiddag.

Inleiding

Welkom op deze werkmiddag! Je gaat vandaag van 13:15 uur tot 16:00 uur bezig met *Branch and Bound*, een onderwerp op het raakvlak van wiskunde en informatica. Wat Branch and Bound precies inhoudt, wordt verderop in deze opdracht duidelijk.

De werkmiddag op het Carmel College Salland zou tot 16:00 uur duren. Omdat wij schatten dat de leerlingen er zo'n drie uur voor nodig zouden hebben, staan wij hen een kwartier uitloop toe. Dit geven wij verderop in deze inleiding aan en hebben wij ook tegen de leerlingen gezegd voor aanvang van de werkmiddag.

De opdracht maken jullie in groepjes van twee. Gedurende de middag is overleg met andere groepjes niet toegestaan. Als jullie ergens echt niet uitkomen, mag je uiteraard wel de docent om extra uitleg vragen.

Op het Carmel College Salland worden werkmiddagen doorgaans in groepjes van twee of vier doorgewerkt. Om actief meedoen te bevorderen en meeliften te voorkomen, hebben wij gekozen deze praktische opdracht in groepjes van twee door te laten werken. De leerlingen mochten zelf een partner uitzoeken. In het correctievoorschrift (zie Bijlage B) hebben wij opgenomen bij welke vragen de docent de leerlingen extra uitleg moet geven als zij er niet zelf uitkomen. Het is belangrijk deze extra uitleg als docent ook te geven, omdat bepaalde vragen essentieel zijn voor het verdere begrip van de opdracht. Bij de overige vragen moet uitleg tot een minimum beperkt worden zodat de behaalde cijfers zoveel mogelijk overeen komen met het niveau van de leerlingen. Er wordt immers een schoolexamencijfer gehangen aan de uitkomst van de werkmiddag.

Laptops zijn gedurende de middag niet toegestaan en ditzelfde geldt uiteraard voor mobieltjes. Jullie eindproduct is dan ook een handgeschreven verslag waarin de opgaven uit dit boekje op volgorde zijn uitgewerkt. Het verslag leveren jullie vandaag om 16:00 uur (uiterlijk 16:15 uur) in bij de docent.

Om te voorkomen dat leerlingen tijd verspillen door dingen op proberen te zoeken, staan wij geen laptops of mobieltjes toe gedurende de werkmiddag. Dit heeft als bijkomend voordeel dat leerlingen niet met elkaar kunnen overleggen via het internet, iets wat blijkens de ervaringen van wiskundedocenten op het CCS en de eigen ervaring van de onderzoeker bij dit soort toetsvormen vaak een probleem vormt.

De beoordeling vindt in principe per tweetal plaats, tenzij een individuele beoordeling meer op zijn plaats lijkt. Er wordt beoordeeld op (wiskundig) niveau van de aanpak en uitwerkingen van de opgaven. Het is niet nodig om een voorblad, inhoudsopgave of inleiding te maken. Wel is het nodig iedere opgave te voorzien van een uitwerking of toelichting. Antwoorden zonder de vereiste berekeningen of verklaringen leveren geen of weinig punten op!

Omdat wij verwachtten dat de leerlingen al hun tijd nodig zouden hebben om de praktische opdracht door te werken, willen wij niet dat zij tijd verspillen met het maken van een voorblad, inhoudsopgave of inleiding.

Deze opdracht is verdeeld in drie delen. Reken voor het doorwerken van het eerste deel ongeveer 30 – 45 minuten en voor het doorwerken van het tweede deel anderhalf uur. Mochten jullie niet uit een bepaalde opgave komen, maak je dan niet druk. Dit geldt in het bijzonder voor de opgaven in onderdeel 3; wij kunnen ons voorstellen dat deze niet bij ieder groepje helemaal lukken. Probeer wel, zolang de tijd dit toelaat, alle opgaven en schrijf ook over elke opgave iets in jullie verslag (naast het antwoord kan dit bijv. zijn wat jullie geprobeerd hebben, waarom jullie er niet uitkomen, wat voor soort antwoord jullie verwachten dat er uitkomt, etc.).

Hier geven wij een tijdsindicatie per deel van de praktische opdracht. Volgens Harskamp, De Haan, en Van Streun (2000) vermeldt een goede praktische opdracht in het wiskundeonderwijs een schatting van het aantal studielasturen, op basis van ervaringen met leerlingen. Deze ervaringen met leerlingen hadden wij tijdens het ontwerpen niet, daarom hebben wij ervoor gekozen een tijdsindicatie te geven op basis van een gemaakte schatting in de expertsessies. Het weglaten van de tijdsindicatie was geen optie, omdat wij hiermee ervoor wilden zorgen dat de leerlingen allemaal aan deel 3 toekomen, ook al hebben zij wellicht geen tijd meer om dit deel af te maken. Het belang van toekomen aan dit derde deel zit hem erin dat er een mooie samenvatting van het geleerde uit de eerste twee delen in staat.

We raden jullie aan om de opdrachten niet op te delen, dat gaat niet lukken! Samen erdoorheen en telkens overleggen en samen nadenken zal tot de beste antwoorden leiden.

Het is bekend dat veel leerlingen bij groepswerk het werk opdelen om zo snel mogelijk klaar te zijn. Doordat de opgaven in deze praktische opdracht vaak op eerdere opgaven voortbouwen in plaats van op reeds voor de werkmiddag bestaande voorkennis, raden wij de leerlingen aan om deze manier van werken hier vooral niet toe te passen.

Veel succes met deze opdracht!

Dit was de inleiding. De praktische opdracht zelf is in drie delen opgebouwd. Op de pagina na de inleiding begint het eerste deel. In dit deel wordt het probleem geïntroduceerd.

Deel 1: planningsproblemen

In dit deel van de opdracht wordt het type probleem geïntroduceerd waarvan de leerlingen later, in deel 2, met behulp van Branch and Bound verschillende instanties op gaan lossen. Zoals toegelicht wordt in Hoofdstuk 2, missen de leerlingen de voorkennis van lineair programmeren en (mixed) integer lineair programmeren die normalerwijs als voorkennis wordt gebruikt om Branch and Bound te introduceren. Wij hebben planningsproblemen waarin de maximale vertraging geminimaliseerd moet worden geselecteerd als alternatief (Zdrzalka & Grabowski, 1989). Deze problemen hebben als voordeel voor onze opdracht dat een redelijk intuïtieve lower bound handmatig uitgerekend kan worden zonder dat kennis van lineair programmeren nodig is.

In een fabriek staat een machine die verschillende taken moet verrichten. Elk van deze taken heeft een bepaalde duur. Deze *duur* geven we aan met p_i, waarbij i het nummer van de taak is. Zien we ergens staan $p_2 = 3$, dan weten we dus dat de duur van de tweede taak gelijk is aan 3. Naast een bepaalde duur kent iedere taak ook een lanceer- en een vervalmoment. Het *lanceermoment* van een taak geeft aan vanaf wanneer de machine met de desbetreffende taak kan beginnen en het *vervalmoment* van een taak geeft aan wanneer we willen dat de machine met de desbetreffende taak klaar is. Het lanceermoment geven we aan met r_i en het vervalmoment met d_i, waarbij *i* wederom het nummer van de taak is.

NB. p_i , r_i en d_i zijn ontleend aan de Engelse woorden *processing time*, *release date* en *due date.*

We beginnen de opdracht met de introductie van handige en gebruikelijke notatie. Deze notatie is nieuw voor de leerlingen, vandaar dat wij het voorbeeldje van $p_2 = 3$ geven.

Met de notatie die we nu voorhanden hebben, kunnen we overzichtelijk de taken weergeven die een machine moet uitvoeren. Neem bijvoorbeeld de vier taken in Tabel 1 en de drie taken in Tabel 2. Wat wij nu gaan doen, is het systematisch proberen in te plannen van deze taken. Zo systematisch, dat we het met een klein beetje programmeerkennis gemakkelijk door een computer zouden kunnen laten doen, ook als we veel meer dan drie of vier taken moeten inplannen. Met inplannen bedoelen we simpelweg een volgorde kiezen waarop de machine de taken aanpakt. Uiteraard kan de machine pas aan een ingeplande taak beginnen zodra er niet aan een andere taak gewerkt wordt en het lanceermoment van de ingeplande taak bereikt is. Merk op dat het hierdoor zo kan zijn dat een machine soms een tijdje niks aan het doen is.

Het was lastig deze eerste twee stukken theorie te verwoorden in leerlingentaal. Wat hier staat is het uiteindelijke resultaat van verschillende expertsessies.

Tabel 1: vier taken

i	1	2	3	4
p_i	4	2	6	5
r_i	0	1	3	5
d_i	8	12	11	10

Tabel 2: drie taken

i	1	2	3
p_i	2	4	7
r_i	3	8	0
d_i	6	12	14

Vraag 1) Uit Tabel 1 volgt dat de machine nooit alle taken voor hun vervalmomenten uit kan voeren. Leg dit uit.

Door het oplossen van deze vraag krijgen de leerlingen een goed begrip van wat de getallen in de tabel nou eigenlijk voorstellen. Het is waarschijnlijk dat de bovenstaande theorie nog even teruggelezen moet worden, maar aangezien dit belangrijke theorie is voor onze praktische opdracht is dat helemaal niet erg.

We weten nu dat de machine nooit alle taken uit Tabel 1 voor hun vervalmomenten uit kan voeren. Alle taken moeten echter wel vervuld worden. Een mogelijk criterium waar we nu op kunnen letten, is het aantal taken dat voor hun vervalmoment uitgevoerd kan worden. Dit zouden we dan zo groot mogelijk willen hebben. Wij gaan echter letten op het criterium dat iedere taak zo kort mogelijk na zijn vervalmoment afgerond is (of natuurlijk er al voor). Om precies te zijn: we willen het *maximale* aantal tijdseenheden die tussen de afronding en het vervalmoment van een taak zitten, zo klein mogelijk maken. We kijken hierbij alleen naar taken die te laat zijn, aangezien we het niet erg vinden als een taak te vroeg klaar is.

Hier wordt het criterium toegelicht waar wij op gaan letten. Dit criterium hanteren wij omdat het ons een probleem verschaft waarin voor elke deelverzameling van oplossingen een redelijk intuïtieve lower bound handmatig uitgerekend kan worden zonder dat kennis van lineair programmeren of andere voor de leerlingen onbekende voorkennis vereist is. Wederom een lastig stuk om te verwoorden in leerlingentaal, waarbij het resultaat te danken is aan de expertsessies in ons ontwerpproces.

Vraag 2) Als we de taken uit Tabel 1 inplannen in numerieke volgorde, dus in de volgorde 1-2-3-4, wat is dan het maximale aantal tijdseenheden die tussen de afronding en het vervalmoment van een taak zitten?

Vraag 1 was lastig, deze vraag is wat simpeler. Dit is bewust gedaan om geen leerlingen te ontmoedigen terwijl ze pas net aan de praktische opdracht begonnen zijn. Om deze zelfde reden is deze vraag geplaatst tussen twee stukken theorie die eigenlijk elkaar opvolgen, maar erg veel zijn om in één keer te behappen. Leerlingen reageren over het algemeen niet goed op te lange stukken lastige tekst. Een niet al te lastige vraag ter afwisseling kan er voor zorgen dat de aandacht van de leerlingen erbij blijft.

Laten we het aantal tijdseenheden die tussen de afronding en het vervalmoment van taak nummer i zitten, aanduiden met L_i. Gegeven een volgorde voor de taken, kunnen we L_i uitrekenen voor alle waarden van i (oftewel: voor alle taken). Het maximale aantal tijdseenheden die tussen de afronding en het vervalmoment van een taak zitten, duiden we dan aan met $L_{\max}$, de *oplossingswaarde* behorende bij deze oplossing (volgorde van taken). Als $L_{\max}$ zo klein mogelijk is, spreken we van een *optimale planning*.

Deze theorie bouwt voort op het stuk theorie voor vraag 2. Wiskundige notatie wordt geïntroduceerd om tijd en ruimte te besparen in het vervolg van de opdracht.

Vraag 3) Wat is het kleinst mogelijke maximale aantal tijdseenheden die tussen de afronding en het vervalmoment van een taak uit Tabel 1 zitten? Leg uit hoe jullie aan dit antwoord gekomen zijn en geef een optimale planning.

Door deze vraag dwingen we leerlingen te reflecteren op wat ze zojuist in de twee voorgaande theorieblokken gelezen hebben en het criterium dat gebruikt wordt in deze opdracht echt te snappen. Tijdens de uitvoering van ons ontwerp merkten wij dat sommige leerlingen zich snel maar incorrect door de eerste paar opdrachten heen werkten. Wij hebben daarom ter plekke besloten het antwoord op deze vraag op het bord te noteren zodat leerlingen met een foutief antwoord aangespoord werden om te ontdekken wat ze nog niet helemaal begrepen hadden.

Zoals jullie gemerkt hebben, is het niet al te lastig om een optimale oplossing te vinden voor het inplannen van vier taken op één machine. Bij grote bedrijven zullen echter regelmatig veel meer taken optimaal ingepland moeten worden, wellicht op meerdere machines. Jullie kunnen je waarschijnlijk wel voorstellen dat dit zonder hulp van een computer een vrijwel onmogelijke opgave is. Om de hulp van een computer in te kunnen schakelen, moeten we echter 'denken' als een computer (*algoritmisch denken*). Een mogelijkheid waarop een optimale oplossing gevonden kan worden door een computer is door deze L_{max} uit te laten rekenen voor alle mogelijke planningen en dan de planning waarvoor L_{max} het kleinst is als antwoord te laten geven. Deze methode wordt vaak "brute force" genoemd, oftewel "brute kracht".

Door bovenstaand stuk zien leerlingen het belang van een algoritme om dit type problemen aan te pakken. Ze maken kennis met complete enumeratie en het begrip brute force.

Vraag 4) Voor hoeveel mogelijke planningen moet de computer L_{max} uitrekenen in ons voorbeeld als we de hierboven beschreven strategie toepassen? Laat jullie berekening zien.

Vraag 5) Stel dat we in opdracht van een fabrikant een optimale planning willen vinden voor de 20 taken die zijn machine moet uitvoeren. Neem aan dat de computer waarover we beschikken vijftig nanoseconden (een nanoseconde is één-miljardste van een seconde) nodig heeft om L_{max} uit te rekenen voor een mogelijke planning. Hoe lang heeft de computer dan nodig om met de hierboven beschreven strategie een optimale planning voor de 20 taken te vinden? Geef jullie antwoord in een zo groot mogelijke doch logische tijdseenheid.

De bovenstaande twee vragen worden even puzzelen voor de leerlingen. Combinatoriek wordt op de middelbare school namelijk wel behandeld bij wiskunde A, maar niet bij wiskunde B. De leerlingen zijn dus nog niet bekend met het begrip faculteit. Door hen eerst het aantal mogelijke planningen bij vier taken uit te laten rekenen, komen de meeste van hen waarschijnlijk op de manier om dit ook voor twintig taken te doen en het feit dat dit nodig is om vraag 5 op te lossen.

Uit het antwoord dat jullie hebben gegeven bij vraag 5 blijkt als het goed is dat het geen goede strategie is om de computer L_{max} uit te laten rekenen voor alle mogelijke planningen om de optimale planning te vinden. Als je genoegen neemt met een goede planning die niet per se optimaal is maar wel snel gevonden kan worden door een computer, zijn er vele alternatieven mogelijk. Zulke alternatieven worden *heuristieken* genoemd. Een voorbeeld van een heuristiek voor planningsproblemen is om uit de nog niet geplande taken waarvan het lanceermoment bereikt is telkens de taak in te plannen met het vroegste vervalmoment.

Met het oplossen van vraag 5 komen leerlingen erachter dat het zelfs met een supersnelle computer vele jaren (vele eeuwen zelfs) duurt om een optimale oplossing te vinden voor een planningsprobleem met twintig taken indien gebruik wordt gemaakt van complete enumeratie, en dat dit dus geen goede strategie is. Zij maken nu kennis met het begrip heuristiek. Heuristieken zijn een belangrijk onderdeel van computational thinking en een van onze leerdoelen luidt dan ook dat de leerling kan omschrijven wat een heuristiek is. Door de shockfactor van de grote hoeveelheid tijd die resulteert uit vraag 5, hebben wij naar onze verwachting alle aandacht van de leerlingen er bij dit stuk bij, waardoor zij beter zullen onthouden wat verstaan wordt onder het begrip heuristiek. De heuristiek waar de leerlingen hier kennis mee maken is niet willekeurig gekozen. Bij het uitrekenen van lower bounds in deel 2 (en in deel 3) van de opdracht hebben zij deze nodig.

Vraag 6) Pas de hierboven beschreven heuristiek toe op de vier taken uit Tabel 1. Geeft de heuristiek een goede oplossing?

Het antwoord op deze vraag is ja, aangezien het antwoord op deze vraag en het antwoord op vraag 3 overeenkomen. Dit is een mooie inleiding voor de volgende, uitdagende vraag.

Vraag 7) Verzin voor de vier taken uit Tabel 1 nieuwe lanceer- en vervalmomenten zodanig dat de zwakte van deze heuristiek ten opzichte van het nagaan van alle oplossingen duidelijk wordt.

Deze vraag vereist veel inzicht van de leerlingen; wij verwachten daardoor dat slechts een deel hem kan oplossen. Dit sluit aan bij de tweede ontwerpeis, waarvan een onderdeel is dat de gehaalde eindcijfers een goede weerspiegeling zijn van het niveau van de leerlingen.

Vraag 8) Bedenk een andere mogelijke heuristiek die toegepast kan worden op het planningsprobleem en pas deze toe op de vier taken uit Tabel 1.

Om een heuristiek te kunnen bedenken en toe te passen, is het nodig te snappen wat een heuristiek is en dit is precies wat wij willen dat de leerlingen leren. De theorie voor vraag 6 moet hier wellicht nog eens voor teruggelezen worden.

Zoals gezegd, vindt een heuristiek niet altijd de optimale oplossing. Heuristieken worden ontwikkeld om een computer binnen afzienbare tijd met grote kans een goede oplossing te laten vinden. Er zijn echter ook verschillende manieren om met zekerheid de optimale oplossing te vinden voor een planningsprobleem zonder dat je alle mogelijkheden na hoeft te gaan. Eén daarvan is *Branch and Bound*, waar we in deze werkmiddag in meer detail naar gaan kijken.

Hier wordt nog eens de kern herhaald van wat nou eigenlijk een heuristiek is en er wordt tegelijk een mooi bruggetje gelegd naar het volgende deel van de opdracht waarin de leerlingen kennis maken met Branch and Bound. Wat hier (en in de rest van de opdracht) niet expliciet wordt uitgelegd, is het feit dat er geen efficiënt algoritme bekend is om planningsprobleem op te lossen vanwege de NP-volledigheid van dit type problemen. Zoals uitgelegd in Hoofdstuk 3, gaat dit het middelbareschoolniveau ver te boven. In de onderbouwing voor het gebruik van Branch and Bound beperken wij ons in deze opdracht daarom tot de veel te grote tijdspannen van het oplossen van dit soort problemen met complete enumeratie en het gebrek aan zekerheid van een goede oplossing bij het oplossen van dit soort problemen met heuristieken.

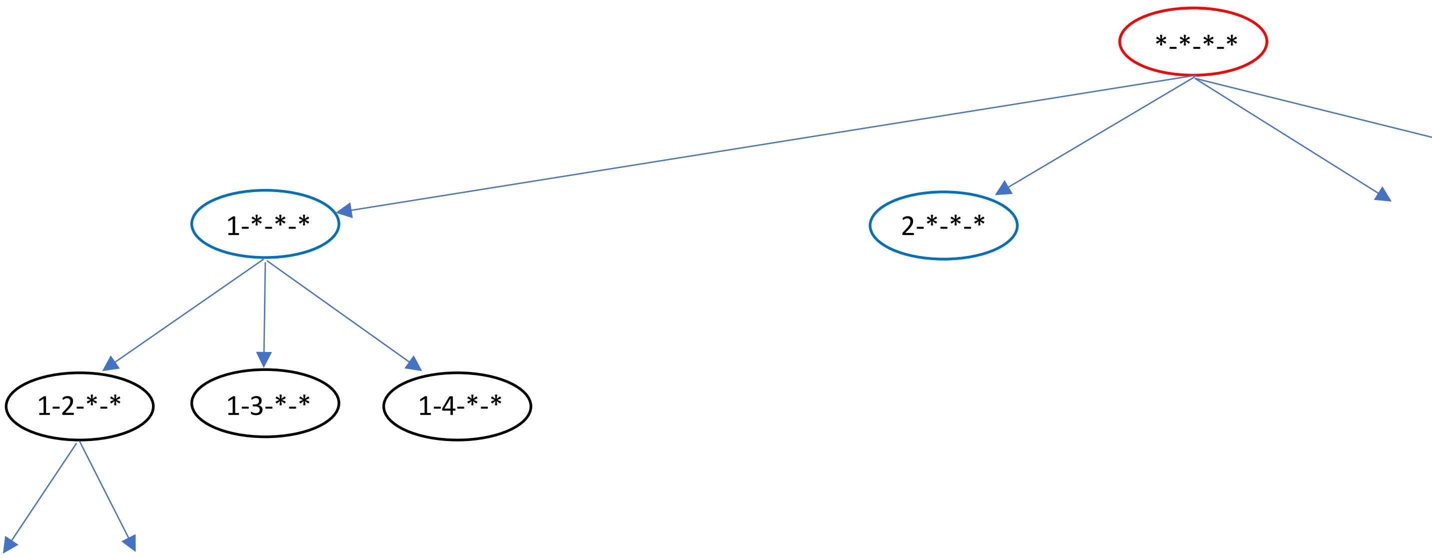
Deel 2: Branch and Bound toepassen op planningsproblemen

In het eerste deel van de opdracht is het probleem geïntroduceerd, evenals twee oplossingsmethoden waarvan de mogelijke nadelen door de leerlingen zijn ontdekt: complete enumeratie, waarbij het erg lang kan duren om tot een optimale oplossing te komen en heuristieken, die niet per se een optimale of zelfs maar goede oplossing vinden. In dit deel van de opdracht maken de leerlingen kennis met het algoritme-ontwerpparadigma dat ook geschikt is om het geïntroduceerde probleem op te lossen, over het algemeen minder last heeft van het nadeel van complete enumeratie en het nadeel van heuristieken niet kent.

In dit deel van de opdracht gaan jullie Branch and Bound toepassen om optimale oplossingen te vinden voor de planningsproblemen uit deel 1. Om dit te kunnen doen, moeten jullie uiteraard eerst leren hoe Branch and Bound in zijn werk gaat. Branch and Bound is een systematische manier om de optimale oplossingen van een probleem te bepalen, zonder dat onnodig werk wordt gedaan. Deze systematische manier van oplossingen bepalen kan wat omslachtig lijken, maar is nuttig als je het bepalen van de optimale oplossing door een computer uit wilt laten voeren.

De motivatie voor het gebruik van Branch and Bound is in het eerste deel van deze opdracht toegelicht. Door de leerlingen in dit deel van de opdracht stap voor stap uit te laten vinden hoe Branch and Bound werkt en hen dit toe te laten passen op voorbeeldsituaties, laten wij hen algoritmisch denken. Wij hebben ervoor gekozen om in de opdracht geen voorbeelden uit te werken om de werking van Branch and Bound te illustreren of in detail uit te leggen waarom Branch and Bound werkt. In combinatie met de puzzelachtige opgaven tussendoor proberen wij op deze manier relationeel begrip van het algoritme-ontwerpparadigma bij de leerlingen te kweken.

Vraag 9) Bepaal op een systematische manier alle mogelijke manieren waarop de taken uit Tabel 1 ingepland kunnen worden. Neem hiertoe onderstaande figuur over in jullie verslag en maak deze af. (Hint 1: Het is waarschijnlijk handig om een of twee hele pagina's te reserveren voor deze opgave en jullie papier in de breedte te gebruiken.) (Hint 2: de kleuren van de ovalen mogen jullie voorlopig negeren.)



Vraag 10) Wat voor overeenkomsten zien jullie tussen jullie antwoord op vraag 4 en de bovenstaande figuur?

Hier zullen leerlingen zien dat uiteindelijk alle mogelijke oplossingen onderaan uitgeschreven staan en dat het aantal ovalen in de onderste generatie dus gelijk is aan het aantal mogelijke oplossingen. Batanero, Navarro-Pelayo en Godino (1997) bepleiten dat het tekenen van een boomdiagram inzicht kan geven in combinatorische aspecten. Doordat leerlingen dat moeten doen in vraag 9 en in vraag

10 zien ze dat het aantal mogelijke oplossingen gelijk is aan het aantal ovalen in de onderste generatie, kunnen zij wellicht zelf de methode bedenken om voor een willekeurig aantal taken het aantal mogelijke oplossingen te bedenken. Voor de leerlingen die niet uit vraag 5 kwamen, is dit dus een mooi moment om nog eens terug te kijken.

Vraag 11) Stel dat er in plaats van vier taken vijf taken ingepland moeten worden. Hoeveel ovalen krijg je dan in totaal als je een figuur tekent zoals die in vraag 9? (Hint: Het is niet nodig om deze figuur te tekenen en dit doen levert geen punten op, je kunt het antwoord beredeneren.)

Dit is net als vraag 7 een vraag die wat meer inzicht van de leerlingen vereist en wij verwachten daardoor dat lang niet alle groepjes deze vraag succesvol zullen oplossen, wat aansluit bij onze eis dat de gehaalde eindcijfers een goede weerspiegeling zijn van het niveau van de leerlingen. Het antwoord op deze vraag al dan niet kunnen vinden heeft geen directe invloed op het verdere succes in de praktische opdracht.

Het Engelse werkwoord *to branch* is te vertalen met *zich vertakken*, zoals een boom dat doet. Nu jullie in vraag 9 alle mogelijke oplossingen op een systematische manier bepaald hebben en de resulterende figuur vergelijken met de figuur op de voorpagina van deze opdracht, zien jullie waar het Branch-gedeelte van Branch and Bound vandaan komt (zeker als je een van de twee figuren op zijn kop houdt). Het Bound-gedeelte van Branch and Bound is erop gericht om het aantal vertakking dat gemaakt moet worden in het Branch-gedeelte te verkleinen zodat maar een (klein) deel van alle mogelijke oplossingen bekeken hoeft te worden en er dus tijd bespaard kan worden zonder dat we de optimale oplossing over het hoofd kunnen zien.

Om te zien hoe het Branch-gedeelte werkt in het voorbeeld van taken inplannen, maken we kennis met *pre-emption (afbreken)*. Als we pre-emption toestaan bij het inplannen van taken, kan een taak op elk moment gewisseld worden met een andere taak (waarvan het lanceermoment is bereikt) ook al is die eerste taak nog niet afgerond. Het werk dat al verricht was gaat niet verloren: als een taak bijvoorbeeld 10 minuten kost en na 4 minuten onderbroken wordt, dan hoeft er later als ie wordt hervat nog maar 6 minuten aan te worden besteed.

Zoals in Hoofdstuk 2 beschreven wordt, is het gebruikelijk om een optimalisatieprobleem om te schrijven naar een (mixed) integer linear programming-probleem voordat Branch and Bound toegepast wordt. Op deze manier kunnen bekende oplossingsmethoden toegepast worden die gebruik maken van LP-relaxaties om tot lower bounds te komen. Omdat lineair programmeren echter geen voorkennis is voor leerlingen in 5 vwo wiskunde B laten wij de leerlingen werken met planningsproblemen waarin de maximale vertraging geminimaliseerd moet worden. Lower bounds voor deelverzamelingen van oplossingen voor dit type problemen kunnen berekend worden door gebruik te maken van pre-emption (Zdrzalka & Grabowski, 1989). Omdat het voor de toepassing van Branch and Bound essentieel is lower bounds voor deelverzamelingen van oplossingen te kunnen berekenen, is het belangrijk dat leerlingen dit correct kunnen doen. Bij vraag 12 en 13 laten wij de leerlingen daarom hun antwoorden checken bij de docent. In deze vragen oefenen zij met pre-emption en dus (zonder dat zij dat in dat stadium van de opdracht weten) met het uitrekenen van lower bounds voor deelverzamelingen van oplossingen van planningsproblemen waarin de maximale vertraging geminimaliseerd moet worden. Leerlingen zullen inzien of gedurende de opdracht uitvinden dat het uitrekenen van lower bounds iets is dat veelvuldig moet gebeuren bij het toepassen van Branch and Bound. Het is dus handig dit op systematische wijze te doen. Om leerlingen onderzoekend en zelfontdekkend bezig te laten zijn, hebben wij hebben ervoor gekozen om geen handige wijze voor het bepalen van lower bounds in de opdracht aan te leveren. Zij zijn hopelijk in staat zelf een systematische manier voor het bepalen van lower bounds te ontwikkelen die voor hen goed werkt.

Vraag 12) Plan de taken uit Tabel 3a zodanig in dat L_{max} zo klein mogelijk is. Sta hierbij pre-emption toe. Als het goed is, vinden jullie $L_1 = L_3 = 1$ en $L_2 = 0$, dus $L_{max} = 1$. Mocht dit niet zo zijn, vraag de docent dan even om jullie te helpen.

Vraag 13) Door vertraging in een ander deel van de fabriek, kan er pas later met taken 1 en 2 begonnen worden. De specificaties van de taken zijn nu zoals in Tabel 3b. Verder is het nu zo dat taak 1 eerst afgerond moet worden, voordat met een andere taak begonnen kan worden (pre-emption is dus pas toegestaan zodra taak 1 is afgerond). Wat is nu de kleinst mogelijk waarde voor L_{max}? Laat jullie antwoord even checken door de docent.

Tabel 3a: drie taken

i	1	2	3
p_i	5	6	5
r_i	0	4	2
d_i	15	11	12

Tabel 3b: drie taken

i	1	2	3
p_i	5	6	5
r_i	1	7	2
d_i	15	11	12

In de praktijk is pre-emption niet vaak toegestaan, taken kunnen gedurende hun uitvoer over het algemeen niet stopgezet worden en later weer gestart. Het uitrekenen van de oplossingswaarde terwijl pre-emption is toegestaan lijkt daarom niet direct nuttig voor het oplossen van het originele planningsprobleem. Het kan echter wiskundig bewezen worden dat als we pre-emption toestaan en de taken inplannen volgens de heuristiek uit vraag 6, we een *lower bound* vinden voor het originele planningsprobleem. Een lower bound voor een probleem is een waarde waar de optimale oplossingswaarde nooit onder kan liggen. Van deze eigenschap maakt Branch and Bound gebruik en het zal gauw duidelijk worden waarom het uitrekenen van de oplossingswaarde terwijl pre-emption is toegestaan toch nuttig is voor het oplossen van het originele planningsprobleem.

Hier wordt uitgelegd dat het zojuist aangeleerde pre-emption gecombineerd kan worden met een heuristiek om zo een lower bound te bepalen. Wij verwachten dat de stapsgewijze opbouw in deze opdracht de meeste leerlingen ertoe in staat stelt lower bounds te berekenen voor deelverzamelingen van oplossingen. Wij laten hen hier in de volgende vraag een keer mee oefenen en daarna knopen we in een stukje theorie het *branchen* en het *bounden* aan elkaar om hen in staat te stellen het planningsprobleem waar zij in deel 1 van de opdracht al mee bezig waren systematisch op te lossen. Het oefenen gebeurt in vraag 14, het toepassen van Branch and Bound op het planningsprobleem waar zij in deel 1 van de opdracht al mee bezig waren gebeurt in vraag 15. Bij zowel vraag 14 als vraag 15 is het de taak van de begeleidende docent om leerlingen te ondersteunen om tot een goed antwoord te komen. De rest van de praktische opdracht bouwt hier namelijk op voort en vraag 15 omvat de kern van dit tweede deel van de opdracht: het kunnen toepassen van Branch and Bound op planningsproblemen waarin de maximale vertraging geminimaliseerd moet worden.

Vraag 14) Pas de heuristiek uit vraag 6 toe op de taken uit Tabel 1. Sta hierbij pre-emption toe. Wat is de waarde van de lower bound die jullie nu gevonden hebben voor dit planningsprobleem?

Bij het uitvoeren van Branch and Bound begint een computer (of in dit geval, beginnen jullie) met het uitrekenen van een lower bound voor het probleem. Dit hebben jullie in vraag 13 gedaan (het is handig deze lower bound alvast te noteren naast het ovaal in jullie verslag dat overeenkomt met het rode ovaal uit vraag 9; een veelgebruikte afkorting voor lower bound is LB). Vervolgens bepaal je lower bounds voor de situaties waarin één taak al gepland is. Dit is te zien als een vertakking vanuit het rode ovaal, waarbij de blauwe ovalen uit vraag 9 gevormd worden.

Bij het bepalen van de lower bounds voor de situaties waarin één taak al gepland is (of meerdere taken al gepland zijn) moet goed op het volgende gelet worden: op de taken die al vast staan in de ovalen wordt pre-emption niet toegepast, maar op de taken die op de sterretjes kunnen komen wel. Neem bijvoorbeeld het ovaal "2***" en vraag je eens af wat dit betekent volgens de systematische notatie die wij gebruiken. Dit ovaal geeft aan dat taak 2 gepland is als eerste taak en dat er nog drie taken niet gepland zijn. Het lanceermoment van taak 2 is 1 en de duur van deze taak is 2. Dat betekent dus dat vanaf tijdstip 3 nog de taken 1, 3 en 4 gepland moeten worden in een nog niet vastgelegde volgorde.

In de volgende stappen van Branch and Bound kies je telkens het ovaal met de laagste lower bound dat nog niet vertakt is. Dit ovaal wordt vertakt en voor de nieuwe ovalen die uit deze vertakking voortkomen bereken je de lower bounds. Als het ovaal dat je kiest niet verder vertakt kan worden (en dus overeenkomt met een mogelijke oplossing) en de lower bound bij dit ovaal lager is die van elk ovaal dat nog niet vertakt is, bereken je de oplossingswaarde behorende bij dit ovaal. Je gaat hiermee door tot een optimale oplossing gevonden is. Dat is het geval als er geen ovalen meer zijn waarvan de lower bound lager is dan de beste gevonden oplossingswaarde tot nu toe.

In de eerste zin van bovenstaand theorieblok switchen we van het gebruik van "de computer" naar het gebruik van "je/jullie". Het gebruik van "de computer" versus het gebruik van "je/jullie" is een aspect dat in de expertsessies in ons ontwerpproces vaak aan de orde is gekomen. Computational thinking kan worden omschreven als "het denkproces waarmee een probleem en zijn oplossing(en) zo worden geformuleerd dat deze effectief door een informatieverwerkende tussenpersoon – mens of machine – kan worden uitgevoerd" (Wing, 2006). Het gaat er dus om problemen aan te kunnen pakken op systematische wijze, niet per se zodanig dat zij direct geprogrammeerd kunnen worden en door een computer kunnen worden opgelost. Dit pleit voor het gebruik van "je/jullie". De motivatie voor het gebruik van Branch and Bound zit er echter in dat een computer je het werk uiteindelijk uit handen kan nemen en wij willen niet dat leerlingen gedemotiveerd raken doordat zij het gevoel krijgen overbodig werk te doen. Het antwoord op vraag 15 bijvoorbeeld hebben zij zelf al in deel 1 van de opdracht gevonden met behulp van logisch nadenken; waarom zouden we nu hetzelfde antwoord dan op zo'n omslachtige wijze gaan zoeken? Dit pleit voor het gebruik van "computer". Uiteindelijk hebben we voor een afwisseling van "computer" en "je/jullie" gekozen en per situatie gekeken wat wij het meest geschikt vonden. Dit is in de gehele praktische opdracht merkbaar, maar vooral in bovenstaand stuk theorie.

Vraag 15) Ga verder met Branch and Bound voor de taken uit Tabel 1 (gebruik jullie resultaat uit vraag 9) totdat een optimale oplossing gevonden is. Bereken en noteer telkens de lower bounds bij de ovalen waarvoor ze uitgerekend worden. Licht toe waarom je weet dat een optimale oplossing gevonden is en je niet verder hoeft te zoeken.

Vraag 16) Voor welk percentage van de ovalen die jullie in vraag 9 getekend hebben hoef je door het toepassen van Branch and Bound in dit probleem geen lower bound of oplossingswaarde te berekenen? En voor welk percentage van de mogelijke oplossingen hoef je door het toepassen van Branch and Bound in dit probleem geen oplossingswaarde te berekenen?

De computationele complexiteit van Branch and Bound is gelijk aan die van brute force. Voor zeer kleine instanties van problemen kan het zelfs zo zijn dat oplossen met Branch and Bound langer kost dan oplossen door complete enumeratie, vanwege de lower bounds die je telkens uit moet rekenen. Voor grotere problemen werkt Branch and Bound over het algemeen echter veel efficiënter dan complete enumeratie doordat een groot deel van de mogelijke oplossingen nooit bekeken hoeft te worden (Kellerer, Pferschy, & Pisinger, 2004; Boyd & Mattingley, 2011). Bovendien kan vaak een manier van lower bounds berekenen gevonden worden die vele malen sneller is dan het uitrekenen van oplossingswaarden en zijn er meer aspecten die Branch and Bound sneller tot een oplossing doen komen die in deze praktische opdracht niet aan bod komen. In het volgende stuk theorie vatten wij dit even samen. Om leerlingen ook zelf door hun werk een idee te laten krijgen van het nut van Branch and Bound hebben wij een planningsprobleem gekozen waaruit de efficiëntie van Branch and Bound blijkt ten opzichte van complete enumeratie ondanks dat het een kleine probleeminstantie betreft. Dit heeft als bijkomend voordeel dat vraag 15 binnen afzienbare tijd met de hand op te lossen is, wat niet het geval zou zijn als vele takken onderzocht moesten worden. Het voordeel van leerlingen ook zelf door hun werk een idee te laten krijgen van het nut van Branch and Bound en het niet alleen in de theorie uit te leggen is dat het waarschijnlijk beter blijft hangen.

De kracht van Branch and Bound is dat oplossingswaarden voor slechts een klein deel van de mogelijke oplossingen berekend hoeven te worden. Zo kan een computer betrekkelijk snel een optimale oplossing vinden voor een probleem en hierdoor kan het omvangrijkere problemen tackelen. Dit is te danken aan de lower bounds, die gebruikt worden om te bepalen dat alle mogelijke vertakkingen van bepaalde ovalen niet meer bekeken hoeven te worden.

NB. Naast de lower bounds bestaan er andere manieren om ovalen en al hun mogelijke vertakkingen vroegtijdig af te schrijven. Gezien de tijd gaan we hier tijdens deze werkmiddag niet verder op in.

Omdat de omvang al redelijk omvangrijk is, hebben wij ervoor gekozen Branch and Bound te beperken tot de kern: het *branchen* en het *bounden*. Aspecten als dominantie laten wij achterwege.

Vraag 17) Stel dat je bezig bent met het toepassen van Branch and Bound op een planningsprobleem en aangekomen bent bij de situatie die wij kunnen weergeven als in de figuur op pagina 10. De volgende stap is om de oplossingswaarde te berekenen voor het ovaal “1-4-3-2”. Wat er na deze stap gebeurt, hangt af van die oplossingswaarde. Beschrijf alle mogelijke scenario’s die zich voor kunnen doen en wat er op basis van die scenario's zal gebeuren (alleen de eerste stap van het vervolg).

Op dit punt van de praktische opdracht aangekomen, hebben leerlingen de mogelijkheid om volledig te snappen hoe het toepassen van Branch and Bound op planningsproblemen waarin de maximale vertraging geminimaliseerd moet worden in zijn werk gaat. In vraag 15 hebben zij dit gedaan op een relatief eenvoudig voorbeeldprobleem. Om hen eens goed na te laten denken over hoe Branch and Bound werkt, hebben wij voor vraag 17 en 18 interessantere voorbeeldproblemen bedacht. Interessante situaties die zich voor kunnen doen in voorbeeldproblemen om precies te zijn. Het voordeel van deze situatieschetsen ten opzichte van gehele voorbeeldproblemen is dat niet allerlei tijdrovend werk (lower bounds en oplossingswaarden berekenen) hoeft te worden verricht om tot de interessante situaties te komen. Bovendien kunnen we de leerlingen hierdoor ook vragen wat de voorgaande stap was om tot een situatie te komen, zonder dat ze stap voor stap zelf die situatie hebben opgebouwd. Dit vereist een goed inzicht in het algoritme. Zoals aangegeven in het correctievoorschrift helpt de docent de leerlingen bij vragen 17 en 18 indien nodig maar moet hij dit niet te snel te doen en, in het geval van vraag 18, enkel bij het eerste deel. De situatieschetsen behorende bij vraag 17 en 18 nemen beide hele pagina's in beslag. Deze geven wij hier niet weer, maar ze zijn te zien is in Bijlage A. De situatieschetsen bevinden zich in de praktische opdracht tussen het eind van deel 2 en het begin van deel 3.

Vraag 18) Stel dat je bezig bent met het toepassen van Branch and Bound op een planningsprobleem en aangekomen bent bij de situatie die wij kunnen weergeven als in de figuur op pagina 11. Wat is nu de volgende stap? En wat is de laatste stap die gedaan is om tot de huidige situatie te komen? Zoals altijd geldt, licht jullie antwoord duidelijk toe.

Vraag 19) Pas Branch and Bound toe op de taken uit Tabel 2 om een optimale planning te vinden. Geef hierbij duidelijk alle stappen en de volgorde van deze stappen aan.

Het planningsprobleem dat weergegeven wordt door Tabel 2 biedt meer uitdagende situaties dan het planningsprobleem dat weergegeven wordt door Tabel 1. Doordat dit probleem een taak minder omvat, is het alsnog binnen afzienbare tijd met de hand op te lossen. Het is een mooie afsluiter van dit deel van de opdracht: al het geleerde komt erin samen en een probleem met een niet direct triviale oplossing kan worden opgelost door op systematische wijze gebruik te maken van een zelfstandig (met enige sturing vanuit de opdracht en indien nodig vanuit de docent) ontdekt algoritme.

Deel 3: Branch and Bound toepassen op andere problemen

Jullie hebben nu geleerd hoe je met behulp van Branch and Bound de optimale oplossing van een planningsprobleem kunt vinden. Het mooie van Branch and Bound is dat deze methode om de optimale oplossing van een probleem te vinden toe te passen is op een breed scala aan problemen. Ook mooi is dat je slechts een klein beetje kennis van programmeren hoeft te hebben om Branch and Bound te implementeren, onafhankelijk van de programmeertaal die je geleerd hebt. Het is zelfs zo dat je helemaal geen kennis van programmeren hoeft te hebben om Branch and Bound te gebruiken, zolang je voor een optimalisatieprobleem de volgende twee aspecten weet te verzinnen:

1. Een manier om te “branchen” of vertakken en zo stap voor stap steeds completere deeloplossingen te creëren.
2. Een manier om lower bounds te bepalen voor de deeloplossingen.

Zodra je deze twee aspecten weet te verzinnen, zijn er voldoende programma’s of programmeurs beschikbaar die de computer met behulp van Branch and Bound een optimale oplossing kunnen laten vinden voor het optimalisatieprobleem. Heel veel problemen kunnen zo geformuleerd worden dat ze met Branch and Bound op te lossen zijn. Voorbeelden hiervan zijn het vinden van optimale locaties voor radio- en telefoonmasten of wifihotspots in de binnenstad en het bepalen van optimale routes voor bijvoorbeeld vrachtschepen, vuilniswagens of fietskoeriers. Hoe efficiënt Branch and Bound werkt ten opzichte van een computer domweg alle mogelijke oplossingen te laten bekijken, hangt allereerst af van het probleem zelf maar is vooral afhankelijk van hoe “scherp” de lower bounds zijn die berekend worden.

In bovenstaand stuk theorie wordt opgedane kennis uit delen 1 en 2 herhaald door een overzicht te geven van de motivatie voor het gebruik van Branch and Bound en door samen te vatten wat er nodig is om Branch and Bound toe te kunnen passen op een probleem. We geven een aantal voorbeeldtoepassingen die opgelost kunnen worden met behulp van het algoritme-ontwerpparadigma Branch and Bound. Door deze voorbeeldtoepassingen zo te kiezen dat ze aansluiten bij de belevingswereld van de leerlingen motiveren wij hen om hetgeen dat zij geleerd hebben te onthouden en om nog even hun best te doen in dit laatste deel van de opdracht (Ebbens & Ettekoven, 2015). In de volgende vraag moeten de leerlingen laten zien dat zij begrip hebben opgedaan van de werking van Branch and Bound en dat zij dit ook kunnen formuleren.

Vraag 20) Hoe efficiënt Branch and Bound werkt ten opzichte van een computer domweg alle mogelijke oplossingen te laten bekijken, hangt voornamelijk af van hoe “scherp” de lower bounds zijn die berekend worden. Licht toe wat volgens jullie wordt bedoeld met het woordje “scherp” in deze zin.

Als je tegen een optimalisatieprobleem aanloopt dat je niet met de hand op kunt lossen en het een computer veel tijd kost om alle mogelijke oplossingen na te gaan, is het waarschijnlijk de moeite waard om te kijken of je Branch and Bound toe kunt passen. Hiertoe hoef je enkel de eerdergenoemde twee aspecten te bedenken: een manier om te vertakken en een manier om lower bounds te bepalen. Dat is geen grote klus, maar toch vaak lastiger dan het lijkt. In vraag 21 is een begin gemaakt met Branch and Bound om een bepaald probleem op te lossen. Een manier om te vertakken en een manier om lower bounds te bepalen zijn dus al bedacht, anders had dit begin niet gemaakt kunnen worden. Aan jullie de taak om op basis van wat gegeven is te bedenken hoe deze keuzes gemaakt zijn en vervolgens Branch and Bound door te zetten om een optimale oplossing te vinden.

Om niet enkel over de brede toepasbaarheid van Branch and Bound te schrijven maar de leerlingen dit ook enigszins te laten merken en om af te sluiten met een leuke uitdagende opgave, hebben wij gezocht naar andere voorbeeldtoepassingen die met Branch and Bound kunnen worden opgelost zonder dat kennis van lineair programmeren benodigd is. Wij vonden zo’n voorbeeldtoepassing in toewijzingsproblemen (Ross & Soland, 1975). Ozgunsur (2013) voorziet ons van een toewijzingsprobleem dat ons geschikt leek doordat het probleem binnen afzienbare tijd handmatig met behulp van Branch and Bound op te lossen is en doordat wij het probleem zodanig aan konden passen dat het goed aansluit op de belevingswereld van de leerlingen.

Vraag 21) Inspecteer het probleem en de gedeeltelijke uitwerking met Branch and Bound op de volgende pagina. Leg uit welke manier van vertakken en welke manier om lower bounds te bepalen gebruikt zijn en werk Branch and Bound verder uit in jullie verslag. Geef hierbij duidelijk alle stappen en de volgorde van deze stappen aan.

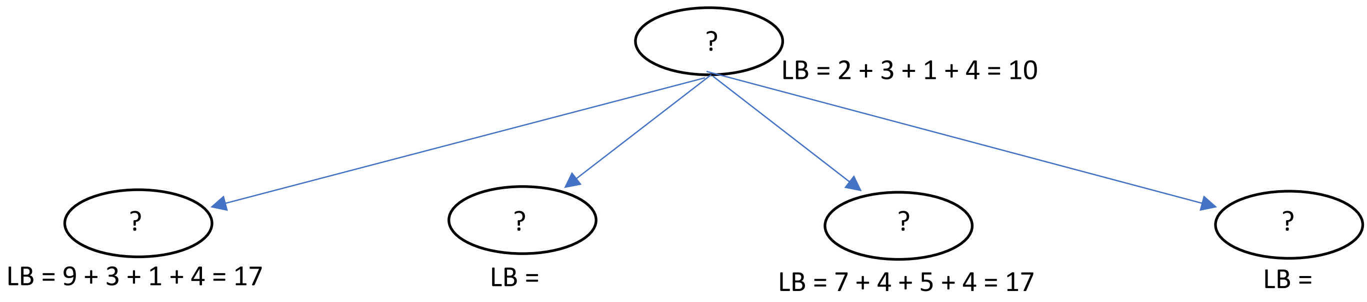
Om deze opdracht uitdagend te maken, geven wij de leerlingen niet de manieren van *branchen* en *bounden* maar laten wij hen deze zelf ontdekken op basis van de incomplete informatie die wij hieronder geven. Dit test het inzicht van de leerlingen en het relationele begrip dat zij van het algoritme-ontwerppardigma hebben opgedaan.

Vier studenten hebben de luxe dat zij onderling vier verschillende functies mogen verdelen bij het bedrijf waar zij voor willen werken. Alle vier de studenten vinden elk van deze functies ongeveer even aantrekkelijk, maar er is een probleem. Om de functies uit te kunnen voeren, zullen zij nog even door moeten studeren. Iets waar zij allen niet zo op zitten te wachten.

Wegens verschillende vooropleidingen zit er nogal een verschil in studietijd die de vier functies van de verschillende studenten vereisen. In Tabel 3 is de vereiste studietijd in jaren per student per functie weergegeven. De studenten besluiten om de vier functies op zo'n manier te verdelen dat hun totale studietijd zo kort mogelijk is. Om er zeker van te zijn dat zij de verdeling van functies vinden die overeenkomt met de kortst mogelijke totale studietijd zonder dat zij alle mogelijke verdelingen hoeven te bekijken, passen de studenten Branch and Bound toe. Het begin van deze procedure is weergegeven onderaan deze pagina; hierbij is een deel weggevallen achter de vraagtekens. Ga om vraag 21 te kunnen beantwoorden allereerst na wat er onder de vraagtekens moet staan. Bedenk vervolgens hoe de lower bounds berekend worden en werk tot slot Branch and Bound verder uit.

Tabel 3: vereiste studiejaren per student per functie

	Functie A	Functie B	Functie C	Functie D
	9	2	7	8
	6	4	3	7
	5	8	1	8
	7	6	9	4



5.2 Analyse

In Hoofdstuk 3 hebben wij vijf ontwerpeisen opgesteld. In deze sectie analyseren wij ons ontwerp door na te gaan of het aan deze ontwerpeisen voldoet. Wij maken in deze analyse gebruik van de leerlingeninterviews die te vinden zijn in Bijlage D. Wij vatten de relevante antwoorden van de leerlingen samen en geven deze weer in een tabel telkens wanneer informatie uit de leerlingeninterviews toepasselijk is voor de analyse van een ontwerpeis.

Ontwerpeis 1: De praktische opdracht is in ongeveer drie uur door de leerlingen af te ronden

Bij het afnemen van de praktische opdracht hebben wij gemerkt dat veel leerlingen te weinig tijd hadden om de opdracht af te ronden. De tijden waarop de groepjes klaar waren, hebben wij bij de uitvoering van ons ontwerp genoteerd. Deze zijn te vinden in Tabel 1. Doordat wij in alle hectiek aan het eind van de werkmiddag de eindtijden vergeten zijn te noteren van twee van de groepjes, kunnen wij niet zeker zijn of deze kloppen. Dit is aangegeven in de tabel. De eindtijd van de werkmiddag was 16.00u. Zoals te zien is in Tabel 1, waren slechts enkele groepjes voor de eindtijd klaar. Voor aanvang hebben wij, in overeenstemming met de inleiding van de praktische opdracht, aangegeven de leerlingen een kwartier uitloop toe te staan. Om 16.20u uiteindelijk moesten de leerlingen die nog bezig waren van ons hun werk toch echt inleveren ook al hadden zij nog niet alle opdrachten afgerond en konden zij nog wel energie opbrengen om dit te doen.

Tabel 1: Eindtijden van de verschillende groepen

Groep	1	2	3	4	5	6	7	8	9	10	11	12
Eindtijd	15:30	15:30	16:00	16:03	16:05	16:05	16:10	16:15	16:15	16:15	16:15	16:15
Groep	13	14	15	16	17	18	19	20	21	22	23	24
Eindtijd	16:15	16:15	16:15*	16:15*	16:20	16:20	16:20	16:20	16:20	16:20	16:25	16:25

*Eindtijd mogelijk niet accuraat

Niet alleen uit de eindtijden, maar ook uit de interviews is gebleken dat de leerlingen zich hebben moeten haasten tijdens de werkmiddag. Zoals te zien is in Tabel 2, gaven vier van de vijf geïnterviewde leerlingen dit aan. Bij het nakijken van de opdracht hebben wij gemerkt dat veel groepjes niet aan het derde deel van de praktische opdracht toegekomen zijn, wat erg jammer is. Allereerst wordt er een duidelijke, korte samenvatting gegeven in dit derde deel en daarnaast zitten er twee goede inzichtsvragen in waar leerlingen mee kunnen puzzelen. Dit derde deel maakt de brede toepasbaarheid van het algoritme-ontwerpparadigma Branch and Bound duidelijk en wij hadden dan ook graag gezien dat vrijwel alle groepjes hier voldoende tijd voor zouden hebben.

Tabel 2: Antwoorden op de vraag of geïnterviewde leerlingen zich moesten haasten

Leerling 1	Ja, veel tijd kwijt aan het begin van de opdracht
Leerling 2	Ja
Leerling 3	Ja, vraag 21 niet af
Leerling 4	Ja, vraag 21 niet af
Leerling 5	Nee, maar dat is te danken aan groepsgenoot

Ons uitgevoerde ontwerp heeft niet voldaan aan de ontwerpeis dat de praktische opdracht in ongeveer drie uur door de leerlingen af te ronden is. Het is lastig om op basis van de nagekeken praktische opdrachten te kwantificeren hoe ver de leerlingen zijn gekomen en dus hoeveel tijd er nodig is voor het gemiddelde groepje leerlingen om de opdracht in zijn geheel af te ronden. Het is in ieder geval duidelijk dat of de opdracht iets korter moet worden of de werkmiddag iets langer moet duren. Op het CCS wordt volgend jaar in ieder geval vier uur tijd uitgetrokken voor de werkmiddag wiskunde van 5 vwo wiskunde B, een tijdbestek waarin wij verwachten dat het voor vrijwel alle leerlingen mogelijk moet zijn de praktische opdracht door te werken.

Ontwerpeis 2: De praktische opdracht sluit goed aan bij het niveau van leerlingen in 5 vwo wiskunde B

In Hoofdstuk 3 hebben wij toegelicht dat wij onder goed aansluiten bij het niveau van de leerlingen verstaan dat de praktische opdracht in staat is de leerlingen uit te dagen zonder hen te ontmoedigen, maar ook dat de gehaalde cijfers een goede weerspiegeling moeten zijn van het niveau van de leerlingen. Om te testen of aan deze ontwerpeis voldaan is, hebben wij de p'-waarden en de r_{it}-waarden van de vragen in de praktische opdracht uitgerekend. Deze p'-waarden en r_{it}-waarden zijn gebaseerd op de resultaten behaald tijdens de 5 vwo wiskunde B werkmiddag op 28 maart 2017 op het CCS. Aan deze werkmiddag deden 24 groepjes leerlingen mee. De p'-waarden en de r_{it}-waarden zijn te vinden in Tabel 3.

Tabel 3: p'-waarden (in procenten) en r_{it}-waarden

Vraag	1	2	3	4	5	6	7	8	9	10	
p'-waarde	70.8	85.4	89.6	100	79.2	64.6	21.9	70.8	95.8	97.9	
r_{it}-waarde	0.37	0.18	-0.22	-	0.52	0.04	0.21	0.45	0.17	-0.26	
Vraag	11	12	13	14	15	16	17	18	19	20	21
p'-waarde	59.4	87.5	87.5	87.5	61.7	54.2	49.0	57.3	36.7	20.8	22.9
rit-waarde	0.52	-0.04	0.33	0.12	0.35	0.38	0.65	0.61	0.69	0.22	0.75

In Tabel 3 is te zien dat er zowel hoge, middelhoge als lage p'-waarden zijn. Richting het eind van de opdracht worden de p'-waarden wel erg laag, maar dit heeft voornamelijk te maken met de vorige ontwerpeis en minder met een te hoog niveau van de vragen. De p'-waarde van vraag 1 is aanzienlijk lager dan de p'-waarden van de volgende vier vragen. Een te lastig begin van een opdracht kan ertoe leiden dat leerlingen ontmoedigd worden voor de rest van de opdracht. Vraag 1 staat dus wellicht niet op de beste plek.

Bij de r_{it}-waarden valt op dat een drietal negatief zijn en een drietal positief maar erg laag (onder 0.2). Dit is over het algemeen geen goed teken, aangezien een negatieve r_{it}-waarde erop wijst dat die vraag slecht gemaakt is door de leerlingen die goed scoren op de praktische opdracht in zijn algemeenheid en vice versa. Een erg lage r_{it}-waarde vertelt ons dat er amper verband is tussen goed scoren op de praktische opdracht in zijn algemeenheid en de betreffende vraag. Een negatieve of erg lage r_{it}-waarde kan erop duiden dat er iets mis is met de vraag. Nadere inspectie van de vragen met negatieve en erg lage r_{it}-waarden toont ons echter dat dit, afgezien van vraag 6, allemaal vragen met een hoge p'-waarde zijn en dat de negatieve en erg lage r_{it}-waarden te verklaren zijn doordat er één groepje dat heel goed heeft gescoord op de opdracht in zijn geheel slecht heeft gescoord op deze vraag. Afgezien van de erg lage waarde behorende bij vraag 6, zijn de negatieve en erg lage r_{it}-waarde dus met toeval te verklaren. Vraag 6 heeft zowel een lage r_{it}-waarde als een relatief lage p'-waarde, waaruit wij concluderen dat wij de vraagstelling in vraag 6 en de uitleg die aan deze vraag vooraf gaat, bij het verbeteren van ons ontwerp in Hoofdstuk 6 goed onder de loep moeten nemen.

Om verder te testen of de praktische opdracht goed aansluit bij het niveau van leerlingen in 5 vwo wiskunde B hebben wij de leerlingen in ons interview gevraagd wat zij lastig vonden aan de opdracht. Achteraf gezien was deze vraag niet de beste vraag om deze ontwerpeis te testen. Een open vraag die niet al impliceert dat (een deel van) de opdracht lastig werd gevonden door de geïnterviewde leerling was beter geweest. Een vraag als "Wat vond je van het niveau van de opdracht?" met de mogelijkheid om door te vragen als het antwoord slechts een korte beschrijving is, is hier een goed voorbeeld van.

In Tabel 4 geven wij een samenvatting van wat leerlingen blijkens onze interviews lastig vonden aan de praktische opdracht. Zowel het begin als het einde van de opdracht zijn door leerlingen lastig genoemd. Alhoewel wij dit bij het einde van de opdracht als een probleem ervaren, vinden wij het wel verontrustend dat het begin als lastig wordt ervaren. Verder werd er genoemd dat er weinig uitleg was en dat je als leerling dingen gelijk toe moet passen. Dit is inderdaad lastig, maar wij denken dat dit ervoor zorgt dat leerlingen zelf ontdekkend bezig gaan wat op zijn beurt weer leidt tot relationeel begrip (Skemp, 1977). Iets anders wat bijdraagt aan relationeel begrip is interactie (Sapire & Mays, 2008). Hierdoor kan het lawaaiig zijn in het lokaal, iets wat door een leerling als lastig aangegeven werd.

Tabel 4: Wat de geïnterviewde leerlingen lastig vonden aan de praktische opdracht

Leerling 1	Deel 3, vooral vraag 21
Leerling 2	Dat er weinig uitleg was en dat je dingen gelijk moest toepassen
Leerling 3	Het begin, onder andere de betekenis van de cijfers in Tabel 1
Leerling 4	Dat er grote lappen tekst in stonden en dat het lawaaiig was in het lokaal
Leerling 5	Niks

Een leerling gaf aan dat het lastig was dat er grote lappen tekst in de praktische opdracht stonden. Het onderbreken van stukken uitleg en de uitleg verduidelijken door bulletpoints toe te voegen lijkt ons een goed plan. Hier zullen wij bij het verbeteren van ons ontwerp in Hoofdstuk 6 naar kijken. Tot slot was er een leerling die aangaf niks lastig gevonden te hebben. Het is mooi dat hij alles blijkbaar zonder moeilijkheden heeft begrepen, maar het kan er ook op duiden dat wij er met onze praktische opdracht niet in zijn geslaagd deze leerling uit te dagen. Een uitdagende bonusopdracht toevoegen zou een mogelijkheid zijn om leerlingen die de praktische opdracht zonder problemen doorlopen toch te kunnen uitdagen als zij zelf ook graag uitgedaagd worden.

In de walkthrough van ons ontwerp is het niveau van de opdracht ook ter sprake gekomen. Hierin zijn veel overeenkomsten met het bovenstaande te vinden. Dat er sterk gestuurd wordt op relationeel begrip is positief en goed gelukt. Aan het eind van de opdracht zijn een paar zeer uitdagende opgaven te vinden die daar goed op hun plek zijn. Het begin van de opdracht is echter te lastig. De eerste vraag is meteen een inzichtsvraag waardoor leerlingen ontmoedigd kunnen raken. Dat deze vraag eigenlijk te lastig is, is ook gebleken uit zijn p'-waarde en uit de leerlingeninterviews. De introductie van voor de leerlingen nieuwe wiskundige notatie die zij de rest van de opdracht nodig hebben aan het begin van de opdracht gaat erg snel. Bij het nakijken van de praktische opdracht is ons ook opgevallen dat niet alle leerlingen deze notatie onder de knie hebben gekregen.

Samenvattend vinden wij dat de praktische opdracht in zijn geheel goed aansluit bij het niveau van de leerlingen en dat er dus aan deze ontwerpeis voldaan is. Er zijn echter verbeteringen mogelijk, met name in het begin van de opdracht.

Ontwerpeis 3: De praktische opdracht staat op zichzelf, in de zin dat er weinig sturing van de docent nodig is

Bij het uitvoeren van de opdracht hebben wij gelet op de hoeveelheid vragen die de leerlingen stelden door vast te stellen of de docenten het erg rustig, rustig, niet rustig noch druk, druk of zeer druk hadden. Het is zonder twijfel het geval geweest dat beide begeleidende docenten het zeer druk hadden. Zij waren gedurende de hele werkmiddag constant bezig met het beantwoorden van vragen. Er is dus wel veel sturing van de docent nodig geweest, waardoor we concluderen dat er niet aan deze ontwerpeis is voldaan.

Twee aanpassingen die bij de analyse van ontwerpeis 2 naar voren zijn gekomen en die wij in Hoofdstuk 6 aan de opdracht doen, kunnen de opdracht verduidelijken en dus werk van de begeleidende docent(en) uit handen nemen. Deze aanpassingen zijn het uitbreiden van de uitleg in het begin van de opdracht zodat alle leerlingen in 5 vwo wiskunde B dit kunnen begrijpen en het toevoegen van bulletpoints in stukken uitleg. Het toevoegen van bulletpoints heeft als voordelen dat het uitleg kan verduidelijken en dat er minder lange stukken in de opdracht door lijken te staan. Hierdoor zien leerlingen minder snel dingen over het hoofd en weten zij beter wat ze nog even terug kunnen lezen voordat zij de docent om hulp vragen.

Wat ook werk uit handen kan nemen van de begeleidende docent(en) is om enkel te helpen bij de vragen die essentieel zijn voor het verdere begrip van de opdracht zoals aangegeven in het correctievoorschrift. In de inleiding van de praktische opdracht aangeven dat de docent enkel hulp biedt bij specifieke vragen kan leerlingen ertoe aansporen minder snel de docent om hulp te vragen. Zij zullen het dan ook accepteren als de docent bij een bepaalde vraag aangeeft niet te willen helpen. Leerlingen enkel helpen bij bepaalde vragen neemt niet alleen werk uit handen van de docent, het zorgt er ook voor dat de gehaalde cijfers een betere weerspiegeling worden van het niveau van de leerlingen. Er kleeft echter ook een nadeel aan het niet helpen van leerlingen bij bepaalde vragen. Dat vragen niet essentieel zijn voor het verdere begrip van de opdracht betekent namelijk niet dat deze vragen niet erg leerzaam zijn. Een mogelijke oplossing van dit nadeel is de werkmiddag, die in eerste instantie een summatief toetsmoment vormt, een sterk formatief karakter te geven door de praktische opdracht op een later moment nog eens uitgebreid te behandelen (William, 2011).

Ontwerpeis 4: De praktische opdracht is in staat de leerlingen intrinsiek te motiveren

Om te controleren of aan deze ontwerpeis voldaan is, hebben wij gelet op het enthousiasme van de leerlingen tijdens het uitvoeren van ons ontwerp, tijdens de werkmiddag dus. Alhoewel dit natuurlijk een subjectieve beoordeling is, leken de leerlingen over het algemeen erg enthousiast aan het begin van de werkmiddag. Wij vermoeden dat dit toe te schrijven is aan de alternatieve werkvorm waar de leerlingen mee bezig gingen. Bij sommige leerlingen was dit enthousiasme in het eerste uur al verdwenen. Dit zal te maken hebben met het feit dat zij het begin te lastig vonden, zoals gebleken is uit de analyse van ontwerpeisen 2 en 3. Veel leerlingen bleven echter enthousiast en aan het eind van de werkmiddag moesten zij echt aangemaand worden om er een eind aan te breien. De sfeer in het lokaal was gedurende de hele werkmiddag goed.

Om verder te controleren of aan deze ontwerpeis voldaan is, hebben wij aan de geïnterviewde leerlingen gevraagd of zij de praktische opdracht leuk vonden, waarop vier van de vijf leerlingen een positief antwoord gaven. De redenen die zij hiervoor gaven zijn te vinden in Tabel 5. Een van de leerlingen gaf aan de opdracht zelf op zich wel leuk te vinden, maar de werkvorm niet omdat de middag zo lang duurde.

Tabel 5: Waarom de geïnterviewde leerlingen de praktische opdracht leuk vonden

Leerling 1	Uitdagend, anders dan standaard wiskundelessen, zet aan tot nadenken
Leerling 2	Interessant, leuk onderwerp, beetje puzzelen
Leerling 3	Anders dan andere (wiskunde)opdrachten, verdieping in onbekend onderwerp
Leerling 4	-
Leerling 5	Anders (dan standaard wiskundelessen), houdt je bezig, verplicht tot nadenken, nieuw onderwerp

De positieve sfeer die in het lokaal hing gedurende de werkmiddag en het feit dat alle vijf de geïnterviewde leerlingen aangaven de opdracht leuk te vinden, waarvan in ieder geval vier leerlingen goed konden onderbouwen waarom, doen ons concluderen dat de praktische opdracht in staat is leerlingen intrinsiek te motiveren en dat er dus aan deze ontwerpeis voldaan is. Twee leerlingen die mee hebben gedaan aan de werkmiddag hebben naar aanleiding ervan zelfs besloten hun profielwerkstuk bij wiskunde te doen. Het zal gaan over het onderwerp lineair programmeren, waarbij zij Branch and Bound gaan gebruiken om integer linear programming problemen op te lossen. Deze twee leerlingen hebben de praktische opdracht erg leuk gevonden (en goed gemaakt) terwijl zij blijkens hun cijfers normaal niet zo veel op hebben met het schoolvak wiskunde. Voor in ieder geval een paar van de leerlingen die mee hebben gedaan aan de werkmiddag geldt dus dat wij er niet alleen in geslaagd zijn hen te motiveren voor de praktische opdracht, maar ook voor het schoolvak wiskunde of wiskunde in zijn algemeenheid.

Ontwerpeis 5: In de praktische opdracht worden de opgestelde leerdoelen behaald

Aan de hand van de leerlingeninterviews controleren wij of de in Hoofdstuk 3 opgestelde leerdoelen behaald zijn en daarmee of er aan deze ontwerpeis voldaan is. Wij maken ook gebruik van de p'-waarden en r_{it}-waarden uit Tabel 3 die horen bij voor de leerdoelen relevante vragen.

Leerdoel 1: De leerling heeft een intuïtief begrip van computationele complexiteit en daarmee van het nut van het algoritme-ontwerppardigma Branch and Bound.

Wij hebben getest of dit leerdoel behaald is door in de leerlingeninterviews te vragen naar waarom Branch and Bound, een best omslachtige methode, werd toegepast op het probleem in de praktische opdracht. Eventueel hebben wij doorgevraagd naar de meest voor de hand liggende methode om een planningsprobleem op te lossen en naar wat het probleem van die methode is. Met deze meest voor de hand liggende methode doelen wij op complete enumeratie en het probleem van deze methode is uiteraard dat het ongelofelijk lang duurt. Branch and Bound toepassen op het probleem zorgt ervoor dat je waarschijnlijk veel sneller een optimale oplossing vindt. Tabel 6 geeft een samenvatting van met betrekking tot dit leerdoel relevante opmerkingen die de geïnterviewde leerlingen hebben gemaakt.

Tabel 6: Waarom volgens de geïnterviewde leerlingen Branch and Bound werd toegepast

Leerling 1	Om een helder beeld voor de leerlingen te krijgen
Leerling 2	Omdat je dan een heel groot deel niet hoeft uit te rekenen, wat heel veel tijd scheelt
Leerling 3	[Na doorvragen en erop wijzen dat alle mogelijke oplossingen nagaan en de beste selecteren de meest voor de hand liggende methode is om een planningsprobleem op te lossen, over die methode:] dat is veel werk, ook een computer is er lang mee bezig
Leerling 4	Om het duidelijk te maken voor de leerlingen, zodat je als leerling er een beeld bij hebt
Leerling 5	Daarmee hoef je niet alles stuk voor stuk uit te rekenen, wat heel veel tijd kost

Wat wij opmaken uit Tabel 6 is dat twee van de vijf geïnterviewde leerlingen duidelijk voor ogen hebben waarom Branch and Bound wordt toegepast, maar drie anderen de link met efficiëntie ten opzichte van brute force niet leggen. Twee van deze drie leerlingen dachten dat in de praktische opdracht Branch and Bound werd toegepast om de leerlingen een helder beeld erbij te geven, wat een cirkelredenering is en niet ingaat op het waarom-aspect van het toepassen ervan. Achteraf gezien hadden wij hier door moeten vragen in de interviews. Vraag 5 in de praktische opdracht, waaruit duidelijk het nadeel van complete enumeratie blijkt, is blijkens Tabel 3 goed gemaakt door de leerlingen. De p'-waarde van vraag 5 is bijna 80%, waardoor wij verwachtten dat vier van de vijf geïnterviewde leerlingen inzien dat complete enumeratie over het algemeen geen handige strategie is in de praktijk. Tijdens de uitvoer hebben de onderzoeker en de reguliere docent door de vragen en opmerkingen rondom vraag 5 ook gemerkt dat de computationele complexiteit van complete enumeratie op veel leerlingen een behoorlijke indruk heeft gemaakt.

Dat Branch and Bound over het algemeen een aanzienlijk snellere oplossingsmethode is dan complete enumeratie, maken wij zoals uitgelegd in Sectie 5.1 in de praktische opdracht aannemelijk in vraag 16. De p'-waarde behorende bij deze vraag is slechts een kleine 55%. Dat twee van de vijf geïnterviewde leerlingen het nut van Branch and Bound aan kunnen geven van Branch and Bound ten opzichte van complete enumeratie is dus lang zo gek nog niet. Zoals uitgelegd in Hoofdstuk 3 is computationele complexiteit een lastig onderwerp. Wij zijn daarom tevreden over het feit dat dit leerdoel door grofweg de helft van de leerlingen behaald lijkt te zijn. Desalniettemin zullen wij bij het verbeteren van ons ontwerp in Hoofdstuk 6 de praktische opdracht dusdanig aanpassen dat dit leerdoel bij een volgende uitvoer hopelijk door een groter deel van de leerlingen behaald wordt.

Leerdoel 2: De leerling kan omschrijven wat een heuristiek is.

Wij hebben getest of dit leerdoel behaald is door de geïnterviewde leerlingen te vragen of zij een voor- en een nadeel van heuristieken op kunnen noemen en door de bij vraag 6, 7 en 8 behorende p'-waarden en r_{it}-waarden te inspecteren. Vraag 6 is de enige vraag met een lage r_{it}-waarde die niet door een hoge p'-waarde en dus door toeval te verklaren is. Vraag 7 heeft een erg lage p'-waarden en vraag 8 heeft dan wel een gemiddelde p'-waarde, maar dit is een vraag waarvoor de p'-waarde eigenlijk de 100% zou moeten naderen. Blijkbaar zijn heuristieken voor de leerlingen een lastig begrip dat wij in de praktische opdracht niet duidelijk genoeg toegelicht hebben. Zoals blijkt uit Tabel 7 konden de meeste geïnterviewde leerlingen wel het voordeel en het nadeel noemen die heuristieken definiëren. Alhoewel het specifieke leerdoel daarmee behaald is door die leerlingen, zou het goed zijn als vraag 6, 7 en 8 zo aangepast worden dat leerlingen niet alleen kunnen omschrijven wat een heuristiek is maar ook een dieper begrip hebben van hoe deze eruitzien in de context van een (plannings)probleem zodat zij relationeel begrip van het concept heuristieken kweken.

Tabel 7: Een voordeel en een nadeel van heuristieken volgens de geïnterviewde leerlingen

Leerling 1	Voordeel: dat het een stukje sneller gaat dan brute force
Leerling 2	Voordeel: dat je niet alles uit hoeft te rekenen maar dat je denkt, nou, we doen dit maar Nadeel: dat je niet per se de beste oplossing hebt
Leerling 3	Voordeel: dat het sneller gaat dan als je alles gaat uitrekenen Nadeel: dat je alsnog niet de kortste tijd hebt
Leerling 4	Weet niet meer wat een heuristiek is
Leerling 5	Voordeel: Als het niet te gek afwijkt dan kom je op een redelijk goed antwoord uit en je kunt snel een oplossing bepalen Nadeel: [op basis van voorbeeld uit de opdracht dat de leerling zich nog kon herinneren] het is niet helemaal efficiënt

Leerdoel 3: De leerling kan met behulp van Branch and Bound op systematische wijze de optimale oplossing van een voorbeeldprobleem vinden.

Wij hebben getest of dit leerdoel behaald is door in het derde deel van de leerlingeninterviews de leerlingen te confronteren met een planningsprobleem en hen te vragen de optimale oplossing hiervan te vinden met behulp van Branch and Bound. Zoals te zien is in de scans van de aantekeningen die de leerlingen tijdens de interviews gemakt hebben in Bijlage D, zijn vier van de vijf leerlingen erin geslaagd met behulp van Branch and Bound de optimale oplossing voor dit planningsprobleem te vinden. Alhoewel deze vier leerlingen verschillende manieren van notatie hanteerden, pasten zij allen de correcte stappen van Branch and Bound toe. Drie van de vier leerlingen deden dit geheel zelfstandig op systematische wijze en zonder fouten te maken. Op basis hiervan kunnen we concluderen dat dit leerdoel niet door alle leerlingen behaald is maar waarschijnlijk wel door de meesten. Het is echter nog interessant om uit de werkwijze en opmerkingen van de leerlingen bij het oplossen van het voorbeeldprobleem relevante zaken op te maken voor de analyse en mogelijk de verbetering van ons ontwerp. Deze relevante zaken zijn samengevat in Tabel 8.

Wat opvalt bij de leerling die er niet in is geslaagd om met behulp van Branch and Bound de optimale oplossing voor het voorbeeldprobleem te vinden, is dat deze leerling begon met een hele boom te tekenen en vervolgens, na erop gewezen te zijn dat dit niet de bedoeling is omdat we de werking van het algoritme zoals een computer het volgt willen imiteren, niet bovenaan begint met het uitrekenen van lower bounds, waarvan deze leerling overigens aangeeft niet meer te weten hoe dat moet. Dat niet is blijven hangen hoe lower bounds berekend moeten worden, zegt nog niet zoveel over het begrip dat de leerling heeft opgedaan van de werking van Branch and Bound. Dat de leerling denkt dat om Branch and Bound toe te passen eerst de hele boom getekend moet worden en dat de leerling niet bovenaan begint, duidt er wel op dat hij weinig begrip heeft opgedaan van de werking van het algoritme-ontwerpparadigma. Indien de docenten er genoeg tijd voor hebben, is het aan te raden de groepjes die met deel 2 van de opdracht bezig zijn wat gerichte vragen te stellen om te kijken hoe het zit met het begrip van de leerlingen en eventueel wat bij te sturen. Dit kan mooi bij een vraag waar de leerlingen toch al een docent moeten vragen om hun antwoord te controleren.

De andere vier geïnterviewde leerlingen zijn er wel in is geslaagd om met behulp van Branch and Bound de optimale oplossing voor het voorbeeldprobleem te vinden, ook al ging het bij een van hen beduidend minder soepel dan bij de andere drie. Dit lag eraan dat deze leerling niet systematisch te werk ging bij het bepalen van de lower bounds en oplossingswaarden maar hier en daar rondom de

Tabel 8: Interessante aspecten uit de werkwijzen en opmerkingen van de geïnterviewde leerlingen

Leerling 1	- Begint met hele boom te tekenen en vervolgens niet bovenaan met het uitrekenen van lower bounds - Weet niet hoe lower bounds berekend moeten worden - Ziet niet dat je met een deelverzameling van oplossingen verder moet rekenen indien de bijbehorende lower bound lager is dan de laagst gevonden oplossingswaarde
Leerling 2	- Begint met het maken van een tabel met tijdstappen en gebruikt deze om systematisch lower bounds uit te rekenen (sommige worden uit het hoofd uitgerekend) - Merkt op dat een optimale planning gevonden is zodra een oplossingswaarde gelijk is aan de laagste lower bound van deelverzamelingen van oplossingen - Merkt op dat er meerdere optimale planningen kunnen zijn
Leerling 3	- Gaat niet erg systematisch te werk bij het uitrekenen en noteren van lower bounds, waardoor foutjes gemaakt worden - Maakt foutjes, maar merkt op waar deze gemaakt zijn zodra door de onderzoeker erop gewezen wordt dat er iets fout is
Leerling 4	- Maakt geen tabel maar gaat wel systematisch te werk waardoor de lower bounds snel en foutloos uitgerekend worden
Leerling 5	- Merkt op dat in de onderste generatie lower bounds gelijk zijn aan oplossingswaarden - Maakt geen tabel maar gaat wel systematisch te werk waardoor de lower bounds snel en foutloos uitgerekend worden

gegeven tabel wat getalletjes noteerde en daardoor al gauw niet meer wist waar ze naar moest kijken. De andere drie leerlingen gingen wel systematisch te werk, maar op onderscheidende wijzen van elkaar. Om leerlingen onderzoekend en zelfontdekkend bezig te laten zijn, hebben wij er bij het ontwerpen van de praktische opdracht bewust voor gekozen geen handige wijze aan te leveren om de oplossingswaarden of de lower bounds te bepalen. Het is mooi om te zien dat de meeste leerlingen zelf handige en systematische wijzen hebben gevonden. Dit geldt echter niet voor alle leerlingen, terwijl het doorwerken van de opdracht er wel een stuk aangenamer van wordt als je zo'n handige en systematische wijze voor het uitrekenen van de oplossingswaarden en lower bounds weet te bedenken. Zonder voor te kauwen wat een handige manier is en daarmee het zelfontdekkende aspect teniet te doen, kunnen wij in de verbeterde versie van ons ontwerp wel wat opmerken over het belang hiervan, zodat de leerlingen er op gewezen worden dat dit een belangrijke stap is waar ze even goed over na moeten denken.

Tot slot is het leuk om te vermelden dat wij uit de interviews op kunnen maken dat ten minste twee van de leerlingen duidelijk relationeel begrip op hebben gedaan van het algoritme. Leerling 2 heeft opgemerkt dat een optimale planning gevonden is zodra een oplossingswaarde gelijk is aan de laagste lower bound van deelverzamelingen van oplossingen en dat er meerdere optimale planningen kunnen zijn. Leerling 5 heeft opgemerkt dat in de onderste generatie geldt dat lower bounds gelijk zijn aan oplossingswaarden. Deze drie aspecten zijn niet expliciet in de praktische opdracht naar voren gekomen en er is in de interviews op geen enkele wijze naar gehint door de onderzoeker. Dat de leerlingen hier tijdens het oplossen van een voorbeeldprobleem met Branch and Bound zelf mee kwamen, toont dat zij goed doorhebben hoe en waarom het algoritme werkt.

Hoofdstuk 6 - Conclusies en discussie

Met het ontwikkelen van een praktische opdracht is in dit ontwerponderzoek gepoogd de volgende onderzoeksvraag te beantwoorden:

“Hoe kunnen aspecten van computational thinking door middel van een werkmiddag worden aangeleerd aan leerlingen in 5 vwo wiskunde B?”

Wij hebben gepoogd deze onderzoeksvraag te beantwoorden door een praktische opdracht te ontwerpen waarin verschillende aspecten van computational thinking aan bod komen. Het onderwerp van de praktische opdracht is Branch and Bound en de didactische visies die wij in acht genomen hebben bij het ontwerpproces zijn die van geleid heruitvinden en relationeel begrip, zoals toegelicht in Hoofdstuk 2. Hoofdstuk 3 beschrijft vijf ontwerpeisen die wij hebben opgesteld om, volgens de in Hoofdstuk 4 beschreven methode, tot de resultaten in Hoofdstuk 5 te komen.

Onze resultaten bestaan uit het ontwerp, bestaande uit een praktische opdracht en een bijbehorend correctievoorschrift (te vinden zijn in Bijlages A en B), en een analyse van dit ontwerp volgens de in Hoofdstuk 3 opgestelde ontwerpeisen. In Sectie 6.1 worden op basis van deze analyse enkele conclusies getrokken en wordt beschreven hoe die conclusies zich laten vertalen naar de manier waarop aspecten van computational thinking aangeleerd kunnen worden. Op basis van deze conclusies geven wij in Sectie 6.2 aanbevelingen voor de verbetering van ons ontwerp. Hierbij wordt ook gebruik gemaakt van de leerlingeninterviews in Bijlage D, de walkthrough met een eventuele belanghebbende bij ons ontwerp in Bijlage H en op andere wijze opgedane inzichten door de onderzoeker. Op basis van de aanbevelingen in Sectie 6.2 presenteren wij een tweede versie van ons ontwerp in Bijlages E en F. Aanbevelingen voor verder onderzoek komen ook aan bod in Sectie 6.2.

6.1 Conclusies en discussie

Wij hebben in dit onderzoek een praktische opdracht ontworpen voor 5 vwo wiskunde B waarbij wij qua didactiek de principes van geleid heruitvinden (Freudenthal, 1991) en relationeel begrip (Skemp, 1977) in acht hebben genomen. In deze praktische opdracht, die het algoritme-ontwerpparadigma Branch and Bound als onderwerp heeft, komen verschillende aspecten van computational thinking aan bod. De in dit onderzoek ontworpen praktische opdracht is te vinden in Bijlage A en het bijbehorende correctievoorschrift in Bijlage B.

Zoals blijkt uit de analyse, is de ontworpen praktische opdracht in staat geweest de leerlingen intrinsiek te motiveren. De werkvorm, maar ook het onderwerp van de opdracht en de manier waarop de opdracht aanzet tot nadenken hebben hieraan bijgedragen. Voor een drie uur durende werkmiddag is de praktische opdracht echter te omvangrijk. Wij concluderen daaruit dat voor een volgende uitvoering van het ontwerp de praktische opdracht ingekort moet worden of er meer tijd voor de werkmiddag uitgetrokken moet worden. Aangezien het tijd kost om relationeel begrip te kweken van een nieuw onderwerp en er geen overbodige vragen of uitleg in de opdracht staan, is meer tijd uittrekken voor de werkmiddag de logische optie. Dit moet natuurlijk wel mogelijk zijn. Op het CCS wordt volgend jaar in ieder geval vier uur tijd uitgetrokken voor de werkmiddag wiskunde van 5 vwo wiskunde B, een tijdbestek waarin wij verwachten dat het voor vrijwel alle leerlingen mogelijk moet zijn de praktische opdracht door te werken.

Wat verder uit de analyse blijkt, is dat het niveau van de opdracht over het algemeen goed is. Het begin van de opdracht moet echter enigszins uitgebreid en verduidelijkt worden. Het is lang niet alle leerlingen gelukt om de geïntroduceerde notatie direct goed te kunnen begrijpen en te gebruiken. Het is daarom goed om hier iets langer bij stil te staan in de opdracht aangezien het afleidt van de daadwerkelijke inhoud als leerlingen niet goed met de notatie om kunnen gaan. Een aantal vragen vereisen veel inzicht en zijn door slechts een klein deel van de leerlingen correct beantwoord. Door meer tijd uit te trekken voor de opdracht, verwachten wij dat het aandeel leerlingen dat deze inzichtsvragen correct kan beantwoorden aanzienlijk zal stijgen. Zij krijgen immers meer tijd om erover na te denken. Het is echter geen slecht plan om alsnog de praktische opdracht eens kritisch door te nemen en uitleg en vragen waar mogelijk te verduidelijken. Hiermee bereiken wij meteen dat er minder sturing van de docent nodig zal zijn tijdens de werkmiddag. Om het kweken van en berusten op instrumenteel begrip te bemoeilijken en daarmee het kweken van relationeel begrip bij de leerlingen uit te lokken, zijn wij tegen het aanwenden van voorbeelden ter verduidelijking van vragen of uitleg in de praktische opdracht. In plaats daarvan kan gebruik worden gemaakt van nuttige herhaling en het aanwenden van bulletpoints.

Het blijkt uit de analyse dat een aanzienlijk deel van de leerlingen aan de hand van onze praktische opdracht een intuïtief begrip van computationele complexiteit op kan doen. Het principe van geleid heruitvinden is een effectief middel om dit te bewerkstelligen. Leerlingen zelf op basis van enkele gegevens te laten ontdekken hoe ongelofelijk lang een snelle computer erover doet om een op het eerste gezicht behapbaar lijkende probleeminstantie met behulp van complete enumeratie op te lossen, maakt veel indruk op hen. De link leggen tussen de inefficiëntie van complete enumeratie en het nut van een algoritme als Branch and Bound blijkt een minder logische stap te zijn voor veel van de leerlingen. Hier moet meer tijd aan besteed worden in een volgende versie van onze praktische opdracht of toekomstig ander onderwijsmateriaal op het gebied van computational thinking. Waar ook meer tijd aan besteed moet worden is aan heuristieken. Alhoewel het nut van heuristieken ten opzichte van complete enumeratie/Branch and Bound voor veel leerlingen bekend lijkt te zijn na het maken van onze praktische opdracht en evenzo het feit dat heuristieken niet altijd een optimale oplossingen vinden, blijkt dat leerlingen het lastig vinden om ermee te werken. Een gegeven heuristiek precies toepassen, zelf een heuristiek bedenken en een voorbeeld bedenken dat de zwakte van een gegeven heuristiek aangeeft, lukte lang niet alle leerlingen. Het verduidelijken van de uitleg en het langzamer opbouwen van de vragen omtrent heuristieken in de ontworpen praktische opdracht kan eraan bijdragen dat meer leerlingen hier een beter begrip van krijgen en er ook mee leren te werken.

Het blijkt uit de analyse dat het op systematische wijze oplossen van een voorbeeldprobleem met behulp van Branch and Bound een dag na de werkmiddag voor de meeste leerlingen geen problemen oplevert. Dat zij met zo weinig oefening hiertoe in staat zijn, getuigt ervan dat zij relationeel begrip van het algoritme-ontwerpparadigma hebben opgedaan. Al met al blijkt dat het stap voor stap introduceren van het algoritme-ontwerpparadigma Branch and Bound waarbij zij zelf ontdekken waarom het algoritme-ontwerpparadigma werkt bij de meeste leerlingen in 5 vwo wiskunde B succesvol te werken. Er zijn echter leerlingen waarvoor dit wellicht niet de meest geschikte methode is. Om deze leerlingen te ondersteunen, kunnen meer controlemomenten in de opdracht toegevoegd worden zodat de leerlingen weten dat ze op de goede weg zijn of de begeleidende docent kunnen vragen hen in de goede richting te coachen. Het is bij het toevoegen van deze controlemomenten wel belangrijk dat zij niet het proces van zelfontdekking in de weg zitten voor de leerlingen die ze eigenlijk niet nodig hebben. De begeleidende docent(en) gerichte vragen laten stellen zodra leerlingen op een bepaald punt in de opdracht aangekomen zijn om te kijken hoe het zit met het begrip van de leerlingen en eventueel wat bij te sturen is een mogelijke oplossing hiervoor. In het correctievoorschrift kunnen meer specifieke aanwijzingen worden toegevoegd voor de begeleidende docent(en) om dit te bewerkstelligen. Uit de analyse blijkt echter dat begeleidende docenten bij de praktische opdracht in zijn huidige vorm hier waarschijnlijk geen tijd voor hebben. Het is daarom zaak deze meer op zichzelf te laten staan en aan zowel de leerlingen als de begeleidende docent(en) duidelijk te maken dat er bij bepaalde vragen, die niet belangrijk zijn voor het verdere verloop van de opdracht, geen hulp van de docent mogelijk is. Dit zorgt er meteen voor dat aan de hand van deze vragen het niveauverschil tussen leerlingen duidelijk gemaakt kan worden.

Terugkomend op de onderzoeksvraag concluderen wij dat een praktische opdracht over het algoritme-ontwerpparadigma Branch and Bound zoals te vinden is in Bijlage A geschikt kan zijn om aspecten van computational thinking door middel van een werkmiddag aan te leren aan leerlingen in 5 vwo wiskunde B. In ons ontwerp zijn veel verbeteringen mogelijk, maar het vormt een goede basis voor zowel het gebruik in de schoolpraktijk als voor verder onderzoek naar het aanleren van aspecten van computational thinking. De belangrijkste generaliserende conclusies van ons onderzoek zijn als volgt.

- Levitin en Papalaskari (2002) beargumenteren dat het algoritmisch denkvermogen bij (informatica)studenten vaak niet goed ontwikkeld is doordat zij zo goed kunnen programmeren. Deze studenten zijn zo gewend om in een programmeertaal te “denken”, dat zij het lastig vinden een abstractiestap te nemen. Er wordt geconcludeerd dat puzzelachtige problemen deze studenten in staat kunnen stellen de programmeertaal in hun hoofd “uit te zetten” en hen algoritmisch denken aan te leren. In dit onderzoek hebben wij laten zien dat het introduceren van puzzelachtige problemen en een algoritme-ontwerpparadigma dat in staat is om deze op te lossen ook geschikt is op middelbareschoolniveau bij het kweken van algoritmisch denkvermogen en het aanleren van andere aspecten van computional thinking aan leerlingen in 5 vwo wiskunde B.

- Twee belangrijke aspecten van computational thinking zijn computationele complexiteit en heuristieken. Een intuïtief begrip kweken van computationele complexiteit door leerlingen zelf te laten ontdekken hoe veel tijd het een snelle computer wel niet kost om een zeker probleem met behulp van complete enumeratie op te lossen lijkt goed te werken bij leerlingen in 5 vwo wiskunde B, ondanks dat combinatoriek geen voorkennis van hen is. Heuristieken blijken een lastig concept voor de leerlingen te zijn. Om leerlingen te leren zelf een heuristiek te bedenken of een voorbeeld te bedenken dat de zwakte (of kracht) van een gegeven heuristiek aangeeft, is het is van belang niet te grote stappen te maken en te zorgen voor een duidelijke uitleg.

- De didactische visie van het realistisch reken-wiskundeonderwijs en specifiek het principe van geleid heruitvinden leent zich bijzonder goed voor praktische opdrachten. Echter, aangezien leerlingen in 5 vwo wiskunde B echter over het algemeen over slechts weinig voorkennis beschikken die nuttig kan zijn voor het aanleren van aspecten van computational thinking op hun niveau, is het lastig om naadloos bij deze visie aan te sluiten. Als alleen een probleem zou worden geïntroduceerd en leerlingen gevraagd werd om zelf een oplossingsmethode te bedenken die dit probleem op algoritmische wijze kan oplossen, dan zouden er waarschijnlijk door slechts een zeer beperkt aantal leerlingen resultaten worden geboekt of leerdoelen behaald. Er is meer sturing vanuit de opdracht en/of de docent nodig dan de theorie van geleid heruitvinden voorschrijft. Het stap voor stap introduceren van de oplossingsmethode waarbij geen uitgewerkte voorbeelden gegeven worden kan ervoor zorgen dat leerlingen toch zelfontdekkend bezig zijn terwijl wel een groot deel van hen in staat wordt gesteld om resultaten te boeken en leerdoelen te behalen. Het is goed om hierbij controlemomenten in te bouwen zodanig dat leerlingen die dat absoluut nodig hebben meer op weg geholpen kunnen worden maar het zelfontdekkende proces van de andere leerlingen niet in de weg gezeten wordt.

6.2 Aanbevelingen

Wij hebben geconcludeerd dat de door ons ontworpen praktische opdracht een geschikte basis vormt om aspecten van computational thinking aan te leren aan leerlingen in 5 vwo wiskunde B. Aanbevelingen voor verder onderzoek zijn het uitproberen van ons ontwerp in andere leerjaren en/of bij wiskunde A en/of op de havo, om te zien wat daar de leeropbrengsten zijn en er dus iets gezegd kan worden over de restricties waarbinnen ons ontwerp werkt. Het ombouwen van ons ontwerp naar een lessenserie is een andere aanbeveling voor verder onderzoek. Het verloop van de uitvoer en de leeropbrengsten van de werkmiddag kunnen dan vergeleken worden met die van een overeenkomende lessenserie. Een derde aanbeveling is het creëren van een alternatief ontwerp waarbij uitgegaan wordt van andere didactische visies zoals het functionele reken-wiskundeonderwijs in plaats van het realistisch reken-wiskundeonderwijs en/of instrumenteel begrip in plaats van relationeel begrip. Er kan dan een vergelijkend onderzoek plaatsvinden waaruit blijkt op welke wijze in het tijdsbestek van een werkmiddag meer aan computational thinking gerelateerde leerdoelen behaald kunnen worden.

Nog een aanbeveling voor verder onderzoek is het optimaliseren van het in dit onderzoek gepresenteerde ontwerp door meerdere iteraties van uitvoeren, analyseren en verbeteren te doorlopen. Door hieronder een lijst met aanbevelingen ter verbetering van ons ontwerp te presenteren en op basis van deze aanbevelingen een tweede versie van ons ontwerp te maken, voltooien wij met dit onderzoek de eerste iteratie van dit proces. Deze tweede versie van ons ontwerp is te vinden in Bijlages E en F. Op basis van volgende iteraties van het ontwerpproces, kunnen ongetwijfeld meer verbeterpunten gevonden worden die niet in onderstaande lijst voorkomen waardoor de praktische opdracht nog verder verbeterd kan worden en er daardoor nog beter toe in staat wordt om aspecten van computational thinking aan te leren in een werkmiddag. Dit geldt met name voor het laatste deel van de opdracht, waar in onze uitvoer maar weinig leerlingen aan toe zijn gekomen en waar wij dus weinig informatie over hebben die kan leiden tot verbeterpunten. Hieronder de lijst met aanbevelingen ter verbetering van ons ontwerp.

- Op het CCS wordt volgend jaar vier uur tijd uitgetrokken voor de werkmiddag wiskunde van 5 vwo wiskunde B, een tijdbestek waarin wij verwachten dat het voor vrijwel alle leerlingen mogelijk moet zijn de praktische opdracht door te werken. In de inleiding van de praktische opdracht staat een tijdsindicatie die op basis van deze verandering aangepast moet worden.
- Als leerlingen de notatie niet goed begrijpen, leidt dit af van de daadwerkelijke inhoud van de opdracht. Het begin van de opdracht moet daarom uitgebreid en verduidelijkt worden.

- Op basis van de lage p'-waarde van vraag 1 en de walkthrough van ons ontwerp, raden wij aan de opdracht te beginnen met een makkelijkere vraag dan vraag 1. Wij willen immers niet dat leerlingen gedemotiveerd raken doordat ze meteen al niet uit de eerste vraag komen.

- In de walkthrough van ons ontwerp wordt gesuggereerd om de context van het planningsprobleem aan het begin van de opdracht uit te breiden door een fabrieksnaam, een soort machine en enkele specifieke taken te benoemen. Betekenisvolle context aan het probleem geven sluit goed aan bij de in Hoofdstuk 2 beschreven didactische visie. Hoewel er in de huidige vorm van de opdracht al een praktisch en goed voorstelbaar probleem wordt geïntroduceerd, kan met name het optimalisatiecriterium dat op pagina 3 wordt geïntroduceerd op deze manier realistischer worden en meer betekenis krijgen voor de leerlingen.

- Tabel 2 staat in de praktische opdracht naast Tabel 1, maar komt pas aan het einde van deel 2 voor het eerst aan bod. Het zou dus logisch zijn deze tabel te verplaatsen.

- Gelet op de principes van geleid heruitvinden en relationeel begrip, hebben wij er bij het ontwerpen van de praktische opdracht bewust voor gekozen geen handige wijze aan te leveren om de oplossingswaarden of de lower bounds te bepalen. Dit willen wij zo laten. Op basis van onze conclusies raden wij echter wel aan in de verbeterde versie van ons ontwerp wat op merken over het belang van het vinden van een handige, systematische methode zonder voor te kauwen hoe zo'n methode eruit zou kunnen zien.

- Tijdens de uitvoering van ons ontwerp hebben wij gemerkt dat sommige leerlingen zich snel maar incorrect door de eerste paar opdrachten heen werkten. Wij hebben daarom ter plekke besloten het antwoord op vraag 3 op het bord te noteren zodat leerlingen met een foutief antwoord aangespoord werden om te ontdekken wat ze nog niet helemaal begrepen hadden. Wij raden aan dit eventueel noodzakelijke ingrijpen van de docent overbodig te maken door hier een controlemoment in te voegen. Het antwoord door de begeleidende docent te laten checken kan een controlemoment zijn. Dit heeft weliswaar als voordelen dat het zelfontdekkende proces van de leerlingen minimaal in de weg gezeten kan worden en dat het begrip van de leerlingen beter gecontroleerd en eventueel bijgestuurd kan worden, maar als nadeel dat het de docent veel tijd kost en leerlingen die geen moeite hadden met de vraag wellicht gaan zitten niksen totdat de docent is langsgeweest terwijl ze eigenlijk graag verder hadden willen en kunnen gaan. Omdat deze vraag zo vroeg in de opdracht zit en er dus niet veel tijd zal zitten tussen de momenten waarop de verschillende groepjes hieraan toekomen, raden wij aan om een controlemoment in te voegen door middel van een alternatieve vraagstelling.

- Met oog op de leerlingen die moeite hadden het begin van de opdracht door te komen, raden wij aan de zinsnede "niet al te lastig" op pagina 4 te vervangen door "het is mogelijk".

- Omdat er tijdens de uitvoering van ons ontwerp meerdere vragen kwamen over de betekenis van dit voor de leerlingen blijkbaar vreemde woord, raden wij aan "doch" op pagina 4 te vervangen door "maar".

- In het laatste stuk van deel 1 van de praktische opdracht wordt het begrip heuristiek aan de leerlingen geïntroduceerd. Vragen 6, 7 en 8 gaan over heuristieken. Op basis van onze conclusies raden wij aan de uitleg voorafgaand aan vraag 6 te verduidelijken en de vragen omtrent heuristieken langzamer op te bouwen. Door het toevoegen van bulletpoints wordt de uitleg een stuk duidelijker en zien leerlingen minder snel iets over het hoofd. Dit heeft hier zeker toegevoegde waarde. Wij raden aan ook in de rest van de opdracht te kijken of uitleg met behulp van bulletpoints zo kan worden aangepast dat het er duidelijker van wordt of ervoor kan zorgen dat leerlingen minder snel iets belangrijks over het hoofd zien.

- De heuristiek die in vraag 6 aangewend wordt, is niet geheel eenduidig. Mochten er meerdere taken zijn met hetzelfde vroegste vervalmoment is het niet duidelijk wat er moet gebeuren. Dit kan makkelijk aangepast worden door “het” te vervangen door “een” in de praktische opdracht, maar wij raden aan een vraag toe te voegen die leerlingen na laat denken over dit gebrek aan eenduidigheid waardoor zij eens te meer worden gewezen op het belang van precisie (in de wiskunde).

- Op basis van de walkthrough en het interview met leerling 5 raden wij aan vraag 9 aan te passen. Het tekenen van de boom is veel werk, wat voor leerlingen die al een goed beeld in hun hoofd hebben van hoe hij eruit gaat zien behoorlijk overbodig kan voelen. Alhoewel het tekenen van een boomdiagram inzicht kan geven in combinatorische aspecten (Batanero, Navarro-Pelayo, & Godino, 1997), raden wij aan de vraag zo te veranderen dat leerlingen de boom niet helemaal hoeven te tekenen maar slechts een gedeelte ervan.

- Op pagina 7 wordt gebruik gemaakt van het “ie” als persoonlijk voornaamwoord. Dit is erg populair taalgebruik. Het kan vervangen worden door “hij”. Wij raden om de gehele praktische opdracht nog eens kritisch door te nemen op populair of incorrect taalgebruik en op correcte stukken tekst die zodanig anders geformuleerd kunnen worden zodat zij hetzelfde zeggen maar beter te begrijpen zijn of mogelijke onduidelijkheden vermijden.

- Wij raden ook aan om de gehele praktische opdracht nog eens kritisch door te nemen op technische onduidelijk- of onjuistheden en wiskundig taalgebruik. Zo duiden wij het nagaan van alle mogelijke oplossingen en daarvan de beste selecteren aan met “brute force”, wat, hoewel correct, niet handig is. Branch and Bound is technisch gezien namelijk ook een brute force methode. Brute force in de praktische opdracht vervangen door complete enumeratie maakt in dit geval korte metten met eventuele onduidelijkheid.

- Verder raden wij aan de opmaak van ons ontwerp nog eens kritisch te bekijken en waar nodig aan te passen. Zo volgt Tabel 3 in de praktische opdracht Tabel 3b op. Tabel 3 had dus Tabel 4 moeten zijn.

- In vraag 12 en 13 zijn controlemomenten ingebouwd. In vraag 12 is dit gedaan door de vraagstelling en, indien nodig, door het antwoord te laten checken door de docent. In vraag 13 is dit gedaan door het antwoord te laten checken door de docent. Dit zorgt ervoor dat de begeleidende docent of begeleidende docenten bij ieder groepje dat met deel 2 van de opdracht bezig is in ieder geval een keer langskomt. Dit is een mooi moment om wat gerichte vragen te stellen om te kijken hoe het zit met het begrip van de leerlingen en eventueel wat bij te sturen. Zoals blijkt uit onze conclusies, kan dit in het geval van sommige leerlingen van grote toegevoegde waarde zijn voor het behalen van de leerdoelen. Dit kost de docent(en) natuurlijk wel veel tijd. Wij raden daarom aan in het correctievoorschrift op te nemen bij welke vragen de docenten de leerlingen geen hulp mogen bieden. Het is zaak dat de docenten zich hieraan houden zodat ze tijd over houden om de leerlingen die dat absoluut nodig hebben door de vragen heen te helpen die essentieel zijn voor hun begrip van de opdracht. Duidelijk in de praktische opdracht vermelden dat leerlingen bij bepaalde vragen geen kans maken om informatie van de docent te ontgoochelen kan eventueel gezeur voorkomen.

- In de praktische opdracht maken wij over het algemeen gebruik van Nederlandse vertalingen van Engelstalige terminologie. Een uitzondering hierop is “lower bound”, terwijl dit zich toch goed laat vertalen door het Nederlandse woord ondergrens. Wij raden aan het woord ondergrens in de praktische opdracht te gebruiken in plaats van lower bound. Ter introductie van de afkorting LB is het wel handig de term lower bound ook te noemen.

- Om meer leerlingen het nut van Branch and Bound in te laten zien, raden wij aan rondom vraag 16 een terugkoppeling te maken naar complete enumeratie en naar heuristieken. Op basis van de walkthrough raden wij aan rondom deze terugkoppeling te noemen dat bij grotere problemen de efficiëntie van Branch and Bound over het algemeen toeneemt.

- Een wellicht problematische met-before voor een kleine groep leerlingen is dat er voor bepaalde types (polynomiaal oplosbare) problemen efficiënte algoritmes bestaan die in de hiërarchie van complete enumeratie, Branch and Bound en heuristieken tussen Branch and Bound en heuristieken horen te staan. Indien Branch and Bound toepast kan worden is het waarschijnlijk nog steeds een snellere oplossingsmethode dan complete enumeratie, maar vergeleken met het efficiënte algoritme is het waarschijnlijk een vreselijk langzame methode om een optimale oplossing te verkrijgen. Zoals toegelicht in Hoofdstuk 3, is NP-volledigheid een te lastig concept voor leerlingen in 5 vwo. Om de potentieel problematische met-before desalniettemin te kunnen vermijden raden wij aan een opmerking in de praktische opdracht te plaatsen die aangeeft dat er in tegenstelling tot sommige andere problemen geen “slimme oplossingsmethoden” zijn voor planningsprobleem die gegarandeerd sneller zijn dan complete enumeratie en dat daarom heuristieken veel gebruikt worden voor het oplossen ervan. Dat Branch and Bound niet gegarandeerd sneller is dan complete enumeratie, kunnen leerlingen zien aan het probleem dat weergegeven wordt door Tabel 2 in de praktische opdracht. Dat Branch and Bound voor veel probleeminstanties echter wel vele malen efficiënter is dan complete enumeratie, wordt gepoogd in de praktische opdracht aannemelijk te maken.

Hieronder nog twee aanbevelingen die wij, in tegenstelling tot alle aanbevelingen in bovenstaande lijst, niet doorvoeren in de tweede versie van ons ontwerp. De reden hiervoor is dat de manier waarop deze aanbevelingen het beste doorgevoerd kunnen worden sterk afhankelijk is van de wensen van de docent en de mogelijkheden binnen school.

- Op basis van de analyse van ontwerpeis 3 raden wij aan de werkmiddag, die in eerste instantie een summatief toetsmoment vormt, een sterk formatief karakter te geven door de praktische opdracht op een later moment nog eens uitgebreid te behandelen. Dit moet uiteraard wel binnen de mogelijkheden van de school en het lesplan van de docent passen.

- Op basis van de analyse van ontwerpeis 2 raden wij aan een bonusopgave toe te voegen aan het eind van de praktische opdracht. Deze bonusopgave kan een nieuw type probleem introduceren waarbij leerlingen geheel zelfstandig moeten bedenken hoe dat type probleem opgelost kan worden met Branch and Bound (welke manier van vertakken en hoe lower bounds uit te rekenen). Clausen (1999) geeft mogelijke geschikte problemen voor dit doeleinde. Een ander idee, resulterend uit de walkthrough, is om de leerlingen zelf Branch and Bound voor het oplossen van planningsproblemen te laten programmeren. Hiertoe moeten zij uiteraard van een opzetje voorzien worden als zij nog geen of weinig programmeerervaring hebben. Als voor deze laatste optie gekozen wordt, is een wat prikkelender stukje tekst omtrent het programmeren van Branch and Bound in het samenvattende stuk theorie aan het begin van deel 3 op zijn plek.

Tot slot geven wij nog enkele methodologische aanbevelingen. Deze kunnen nuttig zijn bij de analyse van het ontwerp in een eventuele volgende iteratie van het ontwerponderzoek.

- Bij het uitvoeren van het ontwerp kunnen de leerlingen die de praktische opdracht maken aan het eind van de werkmiddag gevraagd worden een korte reflectie op de opdracht te schrijven of een vragenlijst over de praktische opdracht in te vullen. Zo hoef je geen kleine steekproef te nemen, wat het geval is bij het afnemen van interviews zoals wij in dit onderzoek hebben gedaan, maar krijg je input van alle leerlingen.

- Wij raden aan uitgebreid na te denken over de interviews voordat deze afgenomen worden. In ons onderzoek hebben wij vragen gesteld die gesloten waren of bepaalde aannames veronderstelden van de leerlingen. Achteraf gezien hadden een aantal van onze vragen beter geformuleerd kunnen worden waardoor er waarschijnlijk meer nuttige informatie uit de interviews te halen zou zijn geweest voor de analyse van ons ontwerp. Goed nadenken over vragen waarmee je eventueel door mag/wilt vragen kan ook erg nuttig zijn. Dit is vooral het geval als je wilt testen of de leerlingen relationeel begrip op hebben gedaan van de onderwerpen in de leerdoelen. Over het algemeen geldt namelijk dat je niet kunt zien of een leerling relationeel of instrumenteel begrip heeft van een procedure zoals Branch and Bound als deze leerling enkel in staat is om het uit te voeren. Om daarachter te komen moet je als interviewer doorvragen naar waarom de leerlingen bepaalde stappen neemt. Een goed voorbeeld van een vraag die gesteld kan worden om erachter te komen of een leerling relationeel begrip op heeft gedaan van het algoritme-ontwerpparadigma is waarom hij of zij de ondergrens voor de allereerste ovaal uitrekent. Dit is namelijk eigenlijk helemaal niet nodig maar wordt in de opdracht wel gedaan om het algoritme-ontwerpparadigma stapsgewijs te introduceren.

- Het gemaakte werk van de leerlingen kan eventueel ook gebruikt worden bij de analyse en verbetering van het ontwerp. Wij hebben ons in dit onderzoek beperkt tot de leerlingeninterviews en cijfermatige resultaten omdat wij al veel tijd kwijt waren aan het maken van het ontwerp, maar hebben tijdens het nakijken van de praktische opdracht wel gemerkt dat er waarschijnlijk nuttige conclusies uit een analyse van het gemaakte werk getrokken kunnen worden.

Referenties

Aalders, B., & Velthuizen, V. (2017). Leeswijzer voor de map. In *Computational Thinking in de Wiskunde.*

Aalders, B., Velthuizen, V., & Tolboom, J. (2017, 2 juli). Wiskunde en 'computational thinking'. *WiskundE-brief, 782*.

Balanskat, A., & Engelhardt, K. (2014). *Computing our future - Computer programming and coding - Priorities, school curricula and initiatives across Europe.* Brussel: European Schoolnet.

Batanero, C., Navarro-Pelayo, V., & Godino, J. D. (1997). Effect of the implicit combinatorial model on combinatorial reasoning in secondary school pupils. *Educational Studies in Mathematics, 32*, 181-199.

Boyd, S., & Mattingley, J. (2011). *Branch and Bound Methods.* Stanford, CA: Stanford University.

Clausen, J. (1999). *Branch and Bound Algorithms - Principles and Examples.* Kopenhagen: University of Copenhagen.

Coenen, T., Timmer, M., & Verhoef, N. (2016, 22 december). Combinatoriek en Kansrekening. Enschede: Universiteit Twente.

CTWO (2007). *Rijk aan betekenis: Visie op vernieuwd wiskundeonderwijs.* Utrecht: Commissie Toekomst Wiskundeonderwijs.

De Haan, D. (2001). Praktische opdrachten bij wiskunde: verslag van een onderzoek. *Nieuwe Wiskrant, 20* (3), 17-23.

Dijksterhuis, E. J. (1924). Moet het meetkunde-onderwijs gewijzigd worden? *Bijvoegsel op het Nieuw Tijdschrift voor Wiskunde, 1*, 1-26.

Drijvers, P. (2012). Wat bedoelen ze toch met... symbol sense? *Nieuwe Wiskrant, 31*(3), 39-42.

Ebbens, S., & Ettekoven, S. (2015). *Effectief leren basisboek.* Groningen/Houten: Noordhoff Uitgevers.

Edwards, R., & Holland, J. (2013). *What is Qualitative Interviewing?* Londen/New York: Bloomsbury Academic.

Ehrenfest-Afanassjeewa, T. (1924). Wat kan en moet het Meetkunde-onderwijs aan een niet-wiskundige geven. *Paedagogiese Voordrachten*, 3-30.

Fortnow, L. (2009). The Status of the P versus NP Problem. *Communications of the ACM, 52*(9), 78–86.

Freudenthal, H. (1991). *Revisiting Mathematics Education.* Dordrecht: Kluwer Academic Publishers.

Goldebeld, P. (1992). *Toets- en itemanalyse met TIA.* Arnhem: CITO.

Harrell, M. C., & Bradley, M. A. (2009). *Data Collection Methods.* Santa Monica, CA: RAND Corporation.

Harskamp, E., De Haan, D., & Van Streun, A. (2000). *Praktijkbrochure Praktische opdrachten wiskunde.* Groningen: GION.

Heege, J. T. (2008). Wat is realistisch reken-wiskundeonderwijs? Een voordracht van Koeno Gravemeijer. *Reken-wiskundeonderwijs: onderzoek, ontwikkeling, praktijk, 27*(3/4), 3-7.

Kellerer, H., Pferschy, U., & Pisinger, D. (2004). *Kanpsack problems.* Berlin: Springer Berlin Heidelberg.

KNAW (2009). *Rekenonderwijs op de basisschool. Analyse en sleutels tot verbetering.* Alkmaar: Bejo druk & print.

Land, A. H., & Doig, A. G. (1960). An Automatic Method of Solving Discrete Programming Problems. *Econometrica, 28*(3), 497-520.

Levitin, A., & Papalaskari, M. (2002). Using puzzles in teaching algorithms. *Proceedings of the 33rd SIGCSE technical symposium on Computer science education*, (pp. 292-296). Covington, Kentucky, USA.

National Research Council. (2010). *Report of a Workshop on the Scope and Nature of Computational Thinking.* Washington, DC: The National Academies Press.

Nieveen, N., & Folmer, E. (2013). Formative Evaluation in Educational Design Research. In T. Plomp, & N. Nieveen, *Educational Design Research Part A: An Introduction* (pp. 152-159). Enschede: SLO.

Ozgunsur, E. (2013, 13 augustus). *Branch and Bound - An Assignment Problem*. Opgehaald van https://www.youtube.com/watch?v=F4vI_qc_u0Q

Papadimitriou, C. H., & Steiglitz, K. (1982). *Combinatorial optimization: algorithms and complexity.* Upper Saddle River, NJ: Prentice-Hall.

Pijpers, R., Stiller, L., & Boeke, H. (2015). *Computing-onderwijs (Wat kunnen we leren van de Britten?).* Kennisnet.

Platform Onderwijs 2032. (2015). *Analyse dialoog Onderwijs2032.* Den Haag: Platform Onderwijs2032.

Ross, G. T., & Soland, R. M. (1975). A branch and bound algorithm for the generalized assignment problem. *Mathematical Programming, 8*(1), 91–103.

Sapire, I., & Mays, T. (2008). *Developing Understanding in Mathematics.* Braamfontein, Zuid-Afrika: SAIDE.

Skemp, R. (1977). Relational understanding and instrumental understanding. *Mathematics Teaching, 77*, 20-26.

SLO. (2015, 22 december). *Computational thinking*. Opgehaald van Curriculum van de toekomst: http://curriculumvandetoekomst.slo.nl/21e-eeuwse-vaardigheden/digitale-geletterdheid/computational-thinking

Stein, C. (2003). *Branch and Bound.* New York: Colombia University.

Taylor, G., Jungert, T., Mageau, G. A., Schattke, K., Dedic, H., Rosenfield, S., & Koestner, R. (2014). A self-determination theory approach to predicting school achievement over time: The unique role of intrinsic motivation. *Contemporary Educational Psychology, 39*, 342–358.

Van Bruggen, W., Pijpers, R., Stiller, L., & Boeke, H. (2016). *Computational thinking in het Nederlandse onderwijs.* Zoetermeer: Kennisnet.

Van de Craats, J. (2007). Waarom Daan en Sanne niet kunnen rekenen. *Nieuw Archief voor Wiskunde*, 132-136.

Van der Donk, C., & Van Lanen, B. (2016). *Praktijkonderzoek in de school.* Bussum: Uitgeverij Coutinho.

Van der Graaf, A. (2015). *Programmeren, het lezen en schrijven van de toekomst?* Schooljournaal.

Van der Zanden, W. J. (2007). *Een nadere beschouwing van de praktische opdracht in het onderwijs.* Eindhoven: Technische Universiteit Eindhoven.

Verbruggen, I., Frickel, M., Van Hell, J., & Boswinkel, N. (2007). Realistisch reken-wiskundeonderwijs in het speciaal basisonderwijs - attitude van leerkrachten en lesgedrag in de klas. *Reken-wiskundeonderwijs: onderzoek, ontwikkeling, praktijk, 26*(3), 37-46.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127-147.

William, D. (2011). What is assessment for learning? *Studies in Educational Evaluation, 37*, 3-14.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33-35.

Winston, W. L. (2004). *Operations Research - Applications and Algorithms.* Belmond, CA: Thomson Learning.

Zdrzalka, S., & Grabowski, J. (1989). An algorithm for single machine sequencing with release dates to minimize maximum cost. *Discrete Applied Mathematics, 23*(1), 73-89.

Ziegler, G., & Loos, A. (2014). Teaching and Learning “What is Mathematics”. *Proceedings of the International Congress of Mathematicians*, (pp. 1201–1215). Seoul, Korea: Kyung Moon Books.

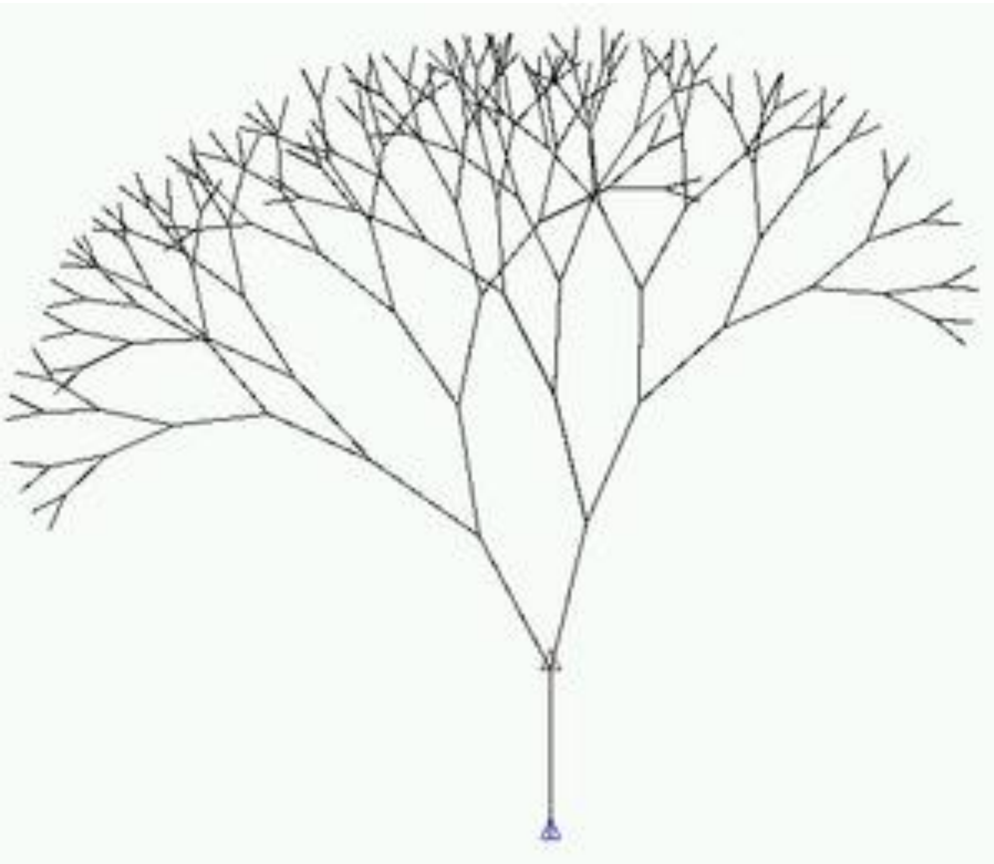
Bijlage A: Praktische opdracht

De komende dertien pagina's geven de praktische opdracht weer die de 5 vwo wiskunde B klassen hebben gemaakt tijdens de werkmiddag van 28 maart 2017 op het Carmel College Salland. Dit is de versie van de praktische opdracht die besproken wordt in Hoofdstuk 4. Het correctievoorschrift behorende bij deze versie van de praktische opdracht is te vinden in Bijlage B.

Praktische Opdracht

5 VWO Wiskunde B

Branch and Bound



28 maart 2017

Inleiding

Welkom op deze werkmiddag! Je gaat vandaag van 13:15 uur tot 16:00 uur bezig met *Branch and Bound*, een onderwerp op het raakvlak van wiskunde en informatica. Wat Branch and Bound precies inhoudt, wordt verderop in deze opdracht duidelijk.

De opdracht maken jullie in groepjes van twee. Gedurende de middag is overleg met andere groepjes niet toegestaan. Als jullie ergens echt niet uitkomen, mag je uiteraard wel de docent om extra uitleg vragen.

Laptops zijn gedurende de middag niet toegestaan en ditzelfde geldt uiteraard voor mobieltjes. Jullie eindproduct is dan ook een handgeschreven verslag waarin de opgaven uit dit boekje op volgorde zijn uitgewerkt. Het verslag leveren jullie vandaag om 16:00 uur (uiterlijk 16:15 uur) in bij de docent.

De beoordeling vindt in principe per tweetal plaats, tenzij een individuele beoordeling meer op zijn plaats lijkt. Er wordt beoordeeld op (wiskundig) niveau van de aanpak en uitwerkingen van de opgaven. Het is niet nodig om een voorblad, inhoudsopgave of inleiding te maken. Wel is het nodig iedere opgave te voorzien van een uitwerking of toelichting. Antwoorden zonder de vereiste berekeningen of verklaringen leveren geen of weinig punten op!

Deze opdracht is verdeeld in drie delen. Reken voor het doorwerken van het eerste deel ongeveer 30 – 45 minuten en voor het doorwerken van het tweede deel anderhalf uur. Mochten jullie niet uit een bepaalde opgave komen, maak je dan niet druk. Dit geldt in het bijzonder voor de opgaven in onderdeel 3; wij kunnen ons voorstellen dat deze niet bij ieder groepje helemaal lukken. Probeer wel, zolang de tijd dit toelaat, alle opgaven en schrijf ook over elke opgave iets in jullie verslag (naast het antwoord kan dit bijv. zijn wat jullie geprobeerd hebben, waarom jullie er niet uitkomen, wat voor soort antwoord jullie verwachten dat er uitkomt, etc.).

We raden jullie aan om de opdrachten niet op te delen, dat gaat niet lukken! Samen erdoorheen en telkens overleggen en samen nadenken zal tot de beste antwoorden leiden.

Veel succes met deze opdracht!

Deel 1: planningsproblemen

In een fabriek staat een machine die verschillende taken moet verrichten. Elk van deze taken heeft een bepaalde duur. Deze *duur* geven we aan met p_i, waarbij i het nummer van de taak is. Zien we ergens staan $p_2 = 3$, dan weten we dus dat de duur van de tweede taak gelijk is aan 3. Naast een bepaalde duur kent iedere taak ook een lanceer- en een vervalmoment. Het *lanceermoment* van een taak geeft aan vanaf wanneer de machine met de desbetreffende taak kan beginnen en het *vervalmoment* van een taak geeft aan wanneer we willen dat de machine met de desbetreffende taak klaar is. Het lanceermoment geven we aan met r_i en het vervalmoment met d_i, waarbij i wederom het nummer van de taak is.

NB. p_i , r_i en d_i zijn ontleend aan de Engelse woorden *processing time*, *release date* en *due date.*

Met de notatie die we nu voorhanden hebben, kunnen we overzichtelijk de taken weergeven die een machine moet uitvoeren. Neem bijvoorbeeld de vier taken in Tabel 1 en de drie taken in Tabel 2. Wat wij nu gaan doen, is het systematisch proberen in te plannen van deze taken. Zo systematisch, dat we het met een klein beetje programmeerkennis gemakkelijk door een computer zouden kunnen laten doen, ook als we veel meer dan drie of vier taken moeten inplannen. Met inplannen bedoelen we simpelweg een volgorde kiezen waarop de machine de taken aanpakt. Uiteraard kan de machine pas aan een ingeplande taak beginnen zodra er niet aan een andere taak gewerkt wordt en het lanceermoment van de ingeplande taak bereikt is. Merk op dat het hierdoor zo kan zijn dat een machine soms een tijdje niks aan het doen is.

Tabel 1: vier taken

i	1	2	3	4
p_i	4	2	6	5
r_i	0	1	3	5
d_i	8	12	11	10

Tabel 2: drie taken

i	1	2	3
p_i	2	4	7
r_i	3	8	0
d_i	6	12	14

Vraag 1) Uit Tabel 1 volgt dat de machine nooit alle taken voor hun vervalmomenten uit kan voeren. Leg dit uit.

We weten nu dat de machine nooit alle taken uit Tabel 1 voor hun vervalmomenten uit kan voeren. Alle taken moeten echter wel vervuld worden. Een mogelijk criterium waar we nu op kunnen letten, is het aantal taken dat voor hun vervalmoment uitgevoerd kan worden. Dit zouden we dan zo groot mogelijk willen hebben. Wij gaan echter letten op het criterium dat iedere taak zo kort mogelijk na zijn vervalmoment afgerond is (of natuurlijk er al voor). Om precies te zijn: we willen het *maximale* aantal tijdseenheden die tussen de afronding en het vervalmoment van een taak zitten, zo klein mogelijk maken. We kijken hierbij alleen naar taken die te laat zijn, aangezien we het niet erg vinden als een taak te vroeg klaar is.

Vraag 2) Als we de taken uit Tabel 1 inplannen in numerieke volgorde, dus in de volgorde 1-2-3-4, wat is dan het maximale aantal tijdseenheden die tussen de afronding en het vervalmoment van een taak zitten?

Laten we het aantal tijdseenheden die tussen de afronding en het vervalmoment van taak nummer i zitten, aanduiden met L_i. Gegeven een volgorde voor de taken, kunnen we L_i uitrekenen voor alle waarden van i (oftewel: voor alle taken). Het maximale aantal tijdseenheden die tussen de afronding en het vervalmoment van een taak zitten, duiden we dan aan met L_{max}, de *oplossingswaarde* behorende bij deze oplossing (volgorde van taken). Als L_{max} zo klein mogelijk is, spreken we van een *optimale planning*.

Vraag 3) Wat is het kleinst mogelijke maximale aantal tijdseenheden die tussen de afronding en het vervalmoment van een taak uit Tabel 1 zitten? Leg uit hoe jullie aan dit antwoord gekomen zijn en geef een optimale planning.

Zoals jullie gemerkt hebben, is het niet al te lastig om een optimale oplossing te vinden voor het inplannen van vier taken op één machine. Bij grote bedrijven zullen echter regelmatig veel meer taken optimaal ingepland moeten worden, wellicht op meerdere machines. Jullie kunnen je waarschijnlijk wel voorstellen dat dit zonder hulp van een computer een vrijwel onmogelijke opgave is. Om de hulp van een computer in te kunnen schakelen, moeten we echter 'denken' als een computer (*algoritmisch denken*). Een mogelijkheid waarop een optimale oplossing gevonden kan worden door een computer is door deze L_{max} uit te laten rekenen voor alle mogelijke planningen en dan de planning waarvoor L_{max} het kleinst is als antwoord te laten geven. Deze methode wordt vaak "brute force" genoemd, oftewel "brute kracht".

Vraag 4) Voor hoeveel mogelijke planningen moet de computer L_{max} uitrekenen in ons voorbeeld als we de hierboven beschreven strategie toepassen? Laat jullie berekening zien.

Vraag 5) Stel dat we in opdracht van een fabrikant een optimale planning willen vinden voor de 20 taken die zijn machine moet uitvoeren. Neem aan dat de computer waarover we beschikken vijftig nanoseconden (een nanoseconde is één-miljardste van een seconde) nodig heeft om L_{max} uit te rekenen voor een mogelijke planning. Hoe lang heeft de computer dan nodig om met de hierboven beschreven strategie een optimale planning voor de 20 taken te vinden? Geef jullie antwoord in een zo groot mogelijke doch logische tijdseenheid.

Uit het antwoord dat jullie hebben gegeven bij vraag 5 blijkt als het goed is dat het geen goede strategie is om de computer L_{max} uit te laten rekenen voor alle mogelijke planningen om de optimale planning te vinden. Als je genoegen neemt met een goede planning die niet per se optimaal is maar wel snel gevonden kan worden door een computer, zijn er vele alternatieven mogelijk. Zulke alternatieven worden *heuristieken* genoemd. Een voorbeeld van een heuristiek voor planningsproblemen is om uit de nog niet geplande taken waarvan het lanceermoment bereikt is telkens de taak in te plannen met het vroegste vervalmoment.

Vraag 6) Pas de hierboven beschreven heuristiek toe op de vier taken uit Tabel 1. Geeft de heuristiek een goede oplossing?

Vraag 7) Verzin voor de vier taken uit Tabel 1 nieuwe lanceer- en vervalmomenten zodanig dat de zwakte van deze heuristiek ten opzichte van het nagaan van alle oplossingen duidelijk wordt.

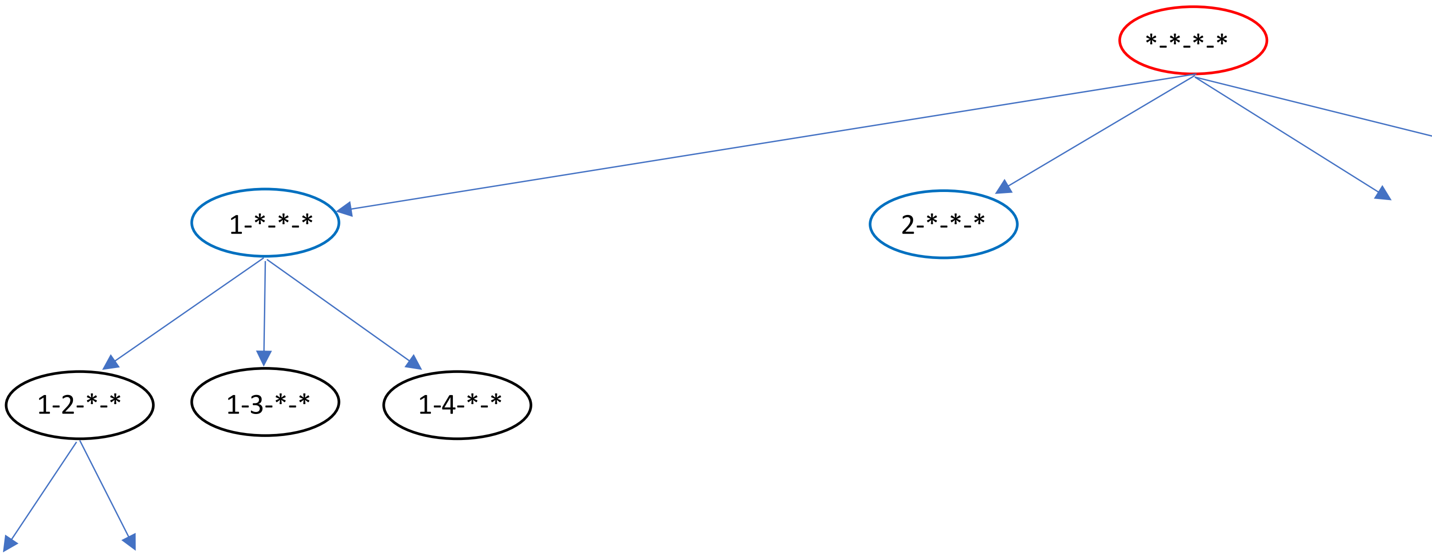
Vraag 8) Bedenk een andere mogelijke heuristiek die toegepast kan worden op het planningsprobleem en pas deze toe op de vier taken uit Tabel 1.

Zoals gezegd, vindt een heuristiek niet altijd de optimale oplossing. Heuristieken worden ontwikkeld om een computer binnen afzienbare tijd met grote kans een goede oplossing te laten vinden. Er zijn echter ook verschillende manieren om met zekerheid de optimale oplossing te vinden voor een planningsprobleem zonder dat je alle mogelijkheden na hoeft te gaan. Eén daarvan is *Branch and Bound*, waar we in deze werkmiddag in meer detail naar gaan kijken.

Deel 2: Branch and Bound toepassen op planningsproblemen

In dit deel van de opdracht gaan jullie Branch and Bound toepassen om optimale oplossingen te vinden voor de planningsproblemen uit deel 1. Om dit te kunnen doen, moeten jullie uiteraard eerst leren hoe Branch and Bound in zijn werk gaat. Branch and Bound is een systematische manier om de optimale oplossingen van een probleem te bepalen, zonder dat onnodig werk wordt gedaan. Deze systematische manier van oplossingen bepalen kan wat omslachtig lijken, maar is nuttig als je het bepalen van de optimale oplossing door een computer uit wilt laten voeren.

Vraag 9) Bepaal op een systematische manier alle mogelijke manieren waarop de taken uit Tabel 1 ingepland kunnen worden. Neem hiertoe onderstaande figuur over in jullie verslag en maak deze af. (Hint 1: Het is waarschijnlijk handig om een of twee hele pagina's te reserveren voor deze opgave en jullie papier in de breedte te gebruiken.) (Hint 2: de kleuren van de ovalen mogen jullie voorlopig negeren.)



Vraag 10) Wat voor overeenkomsten zien jullie tussen jullie antwoord op vraag 4 en de bovenstaande figuur?

Vraag 11) Stel dat er in plaats van vier taken vijf taken ingepland moeten worden. Hoeveel ovalen krijg je dan in totaal als je een figuur tekent zoals die in vraag 9? (Hint: Het is niet nodig om deze figuur te tekenen en dit doen levert geen punten op, je kunt het antwoord beredeneren.)

Het Engelse werkwoord *to branch* is te vertalen met *zich vertakken*, zoals een boom dat doet. Nu jullie in vraag 9 alle mogelijke oplossingen op een systematische manier bepaald hebben en de resulterende figuur vergelijken met de figuur op de voorpagina van deze opdracht, zien jullie waar het Branch-gedeelte van Branch and Bound vandaan komt (zeker als je een van de twee figuren op zijn kop houdt). Het Bound-gedeelte van Branch and Bound is erop gericht om het aantal vertakking dat gemaakt moet worden in het Branch-gedeelte te verkleinen zodat maar een (klein) deel van alle mogelijke oplossingen bekeken hoeft te worden en er dus tijd bespaard kan worden zonder dat we de optimale oplossing over het hoofd kunnen zien.

Om te zien hoe het Branch-gedeelte werkt in het voorbeeld van taken inplannen, maken we kennis met *pre-emption (afbreken)*. Als we pre-emption toestaan bij het inplannen van taken, kan een taak op elk moment gewisseld worden met een andere taak (waarvan het lanceermoment is bereikt) ook al is die eerste taak nog niet afgerond. Het werk dat al verricht was gaat niet verloren: als een taak bijvoorbeeld 10 minuten kost en na 4 minuten onderbroken wordt, dan hoeft er later als ie wordt hervat nog maar 6 minuten aan te worden besteed.

Vraag 12) Plan de taken uit Tabel 3a zodanig in dat $L_{\max}$ zo klein mogelijk is. Sta hierbij pre-emption toe. Als het goed is, vinden jullie $L_1 = L_3 = 1$ en $L_2 = 0$, dus $L_{\max} = 1$. Mocht dit niet zo zijn, vraag de docent dan even om jullie te helpen.

Vraag 13) Door vertraging in een ander deel van de fabriek, kan er pas later met taken 1 en 2 begonnen worden. De specificaties van de taken zijn nu zoals in Tabel 3b. Verder is het nu zo dat taak 1 eerst afgerond moet worden, voordat met een andere taak begonnen kan worden (pre-emption is dus pas toegestaan zodra taak 1 is afgerond). Wat is nu de kleinst mogelijk waarde voor $L_{\max}$? Laat jullie antwoord even checken door de docent.

Tabel 3a: drie taken

i	1	2	3
p_i	5	6	5
r_i	0	4	2
d_i	15	11	12

Tabel 3b: drie taken

i	1	2	3
p_i	5	6	5
r_i	1	7	2
d_i	15	11	12

In de praktijk is pre-emption niet vaak toegestaan, taken kunnen gedurende hun uitvoer over het algemeen niet stopgezet worden en later weer gestart. Het uitrekenen van de oplossingswaarde terwijl pre-emption is toegestaan lijkt daarom niet direct nuttig voor het oplossen van het originele planningsprobleem. Het kan echter wiskundig bewezen worden dat als we pre-emption toestaan en de taken inplannen volgens de heuristiek uit vraag 6, we een *lower bound* vinden voor het originele planningsprobleem. Een lower bound voor een probleem is een waarde waar de optimale oplossingswaarde nooit onder kan liggen. Van deze eigenschap maakt Branch and Bound gebruik en het zal gauw duidelijk worden waarom het uitrekenen van de oplossingswaarde terwijl pre-emption is toegestaan toch nuttig is voor het oplossen van het originele planningsprobleem.

Vraag 14) Pas de heuristiek uit vraag 6 toe op de taken uit Tabel 1. Sta hierbij pre-emption toe. Wat is de waarde van de lower bound die jullie nu gevonden hebben voor dit planningsprobleem?

Bij het uitvoeren van Branch and Bound begint een computer (of in dit geval, beginnen jullie) met het uitrekenen van een lower bound voor het probleem. Dit hebben jullie in vraag 13 gedaan (het is handig deze lower bound alvast te noteren naast het ovaal in jullie verslag dat overeenkomt met het rode ovaal uit vraag 9; een veelgebruikte afkorting voor lower bound is LB). Vervolgens bepaal je lower bounds voor de situaties waarin één taak al gepland is. Dit is te zien als een vertakking vanuit het rode ovaal, waarbij de blauwe ovalen uit vraag 9 gevormd worden.

Bij het bepalen van de lower bounds voor de situaties waarin één taak al gepland is (of meerdere taken al gepland zijn) moet goed op het volgende gelet worden: op de taken die al vast staan in de ovalen wordt pre-emption <u>niet</u> toegepast, maar op de taken die op de sterretjes kunnen komen <u>wel</u>. Neem bijvoorbeeld het ovaal “2***” en vraag je eens af wat dit betekent volgens de systematische notatie die wij gebruiken. Dit ovaal geeft aan dat taak 2 gepland is als eerste taak en dat er nog drie taken niet gepland zijn. Het lanceermoment van taak 2 is 1 en de duur van deze taak is 2. Dat betekent dus dat vanaf tijdstip 3 nog de taken 1, 3 en 4 gepland moeten worden in een nog niet vastgelegde volgorde.

In de volgende stappen van Branch and Bound kies je telkens het ovaal met de laagste lower bound dat nog niet vertakt is. Dit ovaal wordt vertakt en voor de nieuwe ovalen die uit deze vertakking voortkomen bereken je de lower bounds. Als het ovaal dat je kiest niet verder vertakt kan worden (en dus overeenkomt met een mogelijke oplossing) en de lower bound bij dit ovaal lager is die van elk ovaal dat nog niet vertakt is, bereken je de oplossingswaarde behorende bij dit ovaal. Je gaat hiermee door tot een optimale oplossing gevonden is. Dat is het geval als er geen ovalen meer zijn waarvan de lower bound lager is dan de beste gevonden oplossingswaarde tot nu toe.

Vraag 15) Ga verder met Branch and Bound voor de taken uit Tabel 1 (gebruik jullie resultaat uit vraag 9) totdat een optimale oplossing gevonden is. Bereken en noteer telkens de lower bounds bij de ovalen waarvoor ze uitgerekend worden. Licht toe waarom je weet dat een optimale oplossing gevonden is en je niet verder hoeft te zoeken.

Vraag 16) Voor welk percentage van de ovalen die jullie in vraag 9 getekend hebben hoef je door het toepassen van Branch and Bound in dit probleem geen lower bound of oplossingswaarde te berekenen? En voor welk percentage van de mogelijke oplossingen hoef je door het toepassen van Branch and Bound in dit probleem geen oplossingswaarde te berekenen?

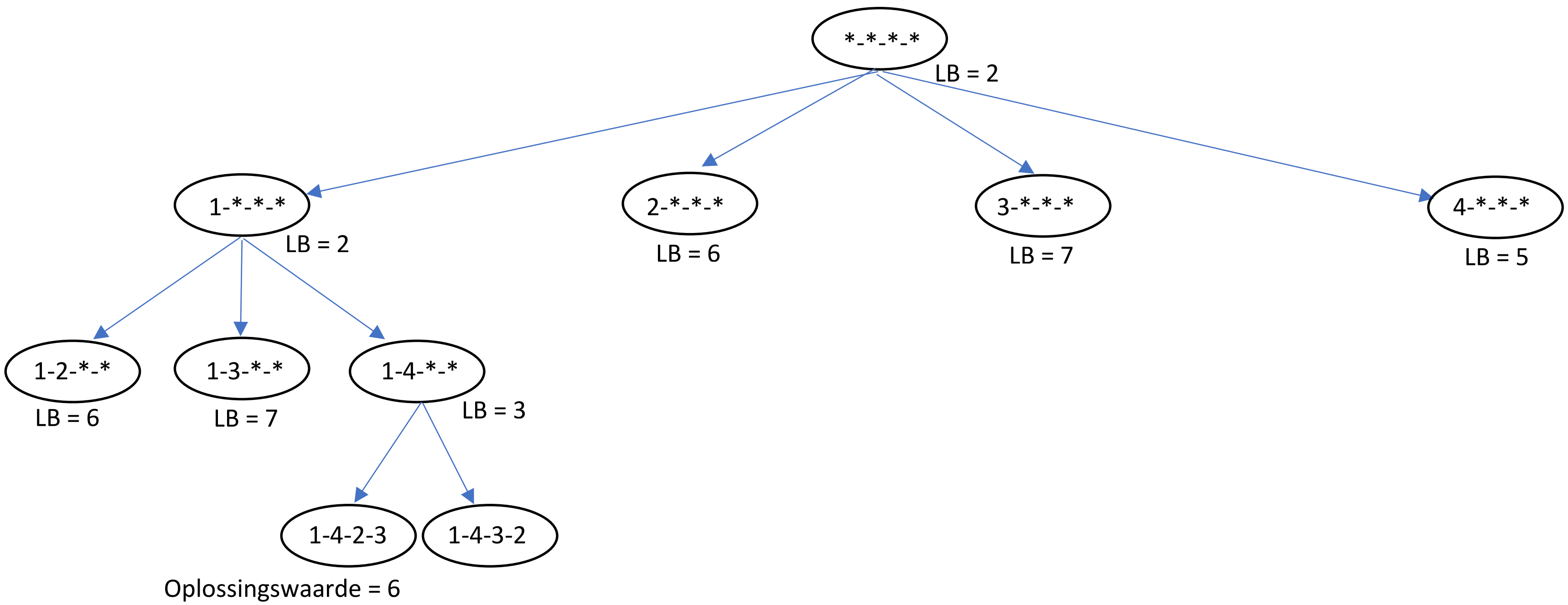
De kracht van Branch and Bound is dat oplossingswaarden voor slechts een klein deel van de mogelijke oplossingen berekend hoeven te worden. Zo kan een computer betrekkelijk snel een optimale oplossing vinden voor een probleem en hierdoor kan het omvangrijkere problemen tackelen. Dit is te danken aan de lower bounds, die gebruikt worden om te bepalen dat alle mogelijke vertakkingen van bepaalde ovalen niet meer bekeken hoeven te worden.

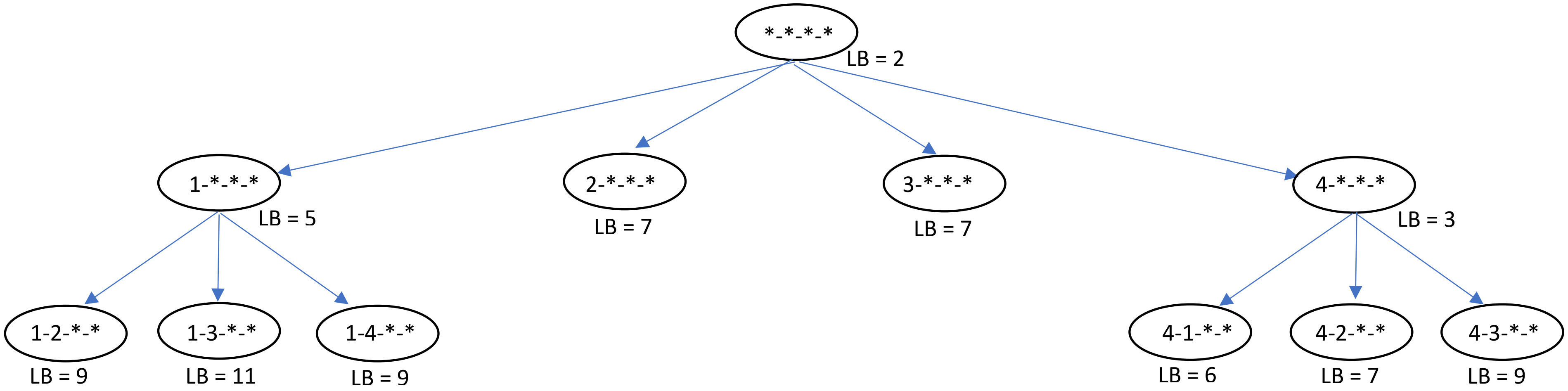
NB. Naast de lower bounds bestaan er andere manieren om ovalen en al hun mogelijke vertakkingen vroegtijdig af te schrijven. Gezien de tijd gaan we hier tijdens deze werkmiddag niet verder op in.

Vraag 17) Stel dat je bezig bent met het toepassen van Branch and Bound op een planningsprobleem en aangekomen bent bij de situatie die wij kunnen weergeven als in de figuur op pagina 10. De volgende stap is om de oplossingswaarde te berekenen voor het ovaal “1-4-3-2”. Wat er na deze stap gebeurt, hangt af van die oplossingswaarde. Beschrijf alle mogelijke scenario’s die zich voor kunnen doen en wat er op basis van die scenario’s zal gebeuren (alleen de eerste stap van het vervolg).

Vraag 18) Stel dat je bezig bent met het toepassen van Branch and Bound op een planningsprobleem en aangekomen bent bij de situatie die wij kunnen weergeven als in de figuur op pagina 11. Wat is nu de volgende stap? En wat is de laatste stap die gedaan is om tot de huidige situatie te komen? Zoals altijd geldt, licht jullie antwoord duidelijk toe.

Vraag 19) Pas Branch and Bound toe op de taken uit Tabel 2 om een optimale planning te vinden. Geef hierbij duidelijk alle stappen en de volgorde van deze stappen aan.


--*-*
LB = 2
1-*-*-*
LB = 2
2-*-*-*
LB = 6
3-*-*-*
LB = 7
4-*-*-*
LB = 5
1-2-*-*
LB = 6
1-3-*-*
LB = 7
1-4-*-*
LB = 3
1-4-2-3
1-4-3-2
Oplossingswaarde = 6


--*-*
LB = 2
1-*-*-*
LB = 5
2-*-*-*
LB = 7
3-*-*-*
LB = 7
4-*-*-*
LB = 3
1-2-*-*
LB = 9
1-3-*-*
LB = 11
1-4-*-*
LB = 9
4-1-*-*
LB = 6
4-2-*-*
LB = 7
4-3-*-*
LB = 9

Deel 3: Branch and Bound toepassen op andere problemen

Jullie hebben nu geleerd hoe je met behulp van Branch and Bound de optimale oplossing van een planningsprobleem kunt vinden. Het mooie van Branch and Bound is dat deze methode om de optimale oplossing van een probleem te vinden toe te passen is op een breed scala aan problemen. Ook mooi is dat je slechts een klein beetje kennis van programmeren hoeft te hebben om Branch and Bound te implementeren, onafhankelijk van de programmeertaal die je geleerd hebt. Het is zelfs zo dat je helemaal geen kennis van programmeren hoeft te hebben om Branch and Bound te gebruiken, zolang je voor een optimalisatieprobleem de volgende twee aspecten weet te verzinnen:

1. Een manier om te “branchen” of vertakken en zo stap voor stap steeds completere deeloplossingen te creëren.
2. Een manier om lower bounds te bepalen voor de deeloplossingen.

Zodra je deze twee aspecten weet te verzinnen, zijn er voldoende programma’s of programmeurs beschikbaar die de computer met behulp van Branch and Bound een optimale oplossing kunnen laten vinden voor het optimalisatieprobleem. Heel veel problemen kunnen zo geformuleerd worden dat ze met Branch and Bound op te lossen zijn. Voorbeelden hiervan zijn het vinden van optimale locaties voor radio- en telefoonmasten of wifihotspots in de binnenstad en het bepalen van optimale routes voor bijvoorbeeld vrachtschepen, vuilniswagens of fietskoeriers. Hoe efficiënt Branch and Bound werkt ten opzichte van een computer domweg alle mogelijke oplossingen te laten bekijken, hangt allereerst af van het probleem zelf maar is vooral afhankelijk van hoe “scherp” de lower bounds zijn die berekend worden.

Vraag 20) Hoe efficiënt Branch and Bound werkt ten opzichte van een computer domweg alle mogelijke oplossingen te laten bekijken, hangt voornamelijk af van hoe “scherp” de lower bounds zijn die berekend worden. Licht toe wat volgens jullie wordt bedoeld met het woordje “scherp” in deze zin.

Als je tegen een optimalisatieprobleem aanloopt dat je niet met de hand op kunt lossen en het een computer veel tijd kost om alle mogelijke oplossingen na te gaan, is het waarschijnlijk de moeite waard om te kijken of je Branch and Bound toe kunt passen. Hiertoe hoef je enkel de eerdergenoemde twee aspecten te bedenken: een manier om te vertakken en een manier om lower bounds te bepalen. Dat is geen grote klus, maar toch vaak lastiger dan het lijkt. In vraag 21 is een begin gemaakt met Branch and Bound om een bepaald probleem op te lossen. Een manier om te vertakken en een manier om lower bounds te bepalen zijn dus al bedacht, anders had dit begin niet gemaakt kunnen worden. Aan jullie de taak om op basis van wat gegeven is te bedenken hoe deze keuzes gemaakt zijn en vervolgens Branch and Bound door te zetten om een optimale oplossing te vinden.

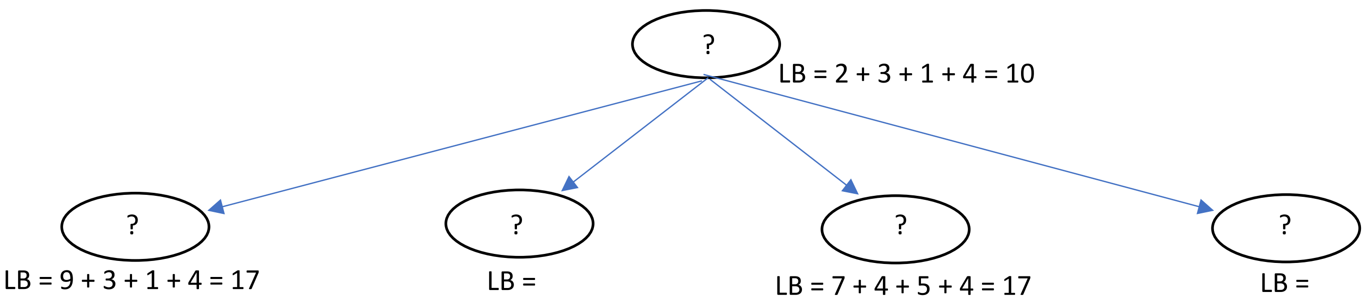
Vraag 21) Inspecteer het probleem en de gedeeltelijke uitwerking met Branch and Bound op de volgende pagina. Leg uit welke manier van vertakken en welke manier om lower bounds te bepalen gebruikt zijn en werk Branch and Bound verder uit in jullie verslag. Geef hierbij duidelijk alle stappen en de volgorde van deze stappen aan.

Vier studenten hebben de luxe dat zij onderling vier verschillende functies mogen verdelen bij het bedrijf waar zij voor willen werken. Alle vier de studenten vinden elk van deze functies ongeveer even aantrekkelijk, maar er is een probleem. Om de functies uit te kunnen voeren, zullen zij nog even door moeten studeren. Iets waar zij allen niet zo op zitten te wachten.

Wegens verschillende vooropleidingen zit er nogal een verschil in studietijd die de vier functies van de verschillende studenten vereisen. In Tabel 3 is de vereiste studietijd in jaren per student per functie weergegeven. De studenten besluiten om de vier functies op zo'n manier te verdelen dat hun totale studietijd zo kort mogelijk is. Om er zeker van te zijn dat zij de verdeling van functies vinden die overeenkomt met de kortst mogelijke totale studietijd zonder dat zij alle mogelijke verdelingen hoeven te bekijken, passen de studenten Branch and Bound toe. Het begin van deze procedure is weergegeven onderaan deze pagina; hierbij is een deel weggevallen achter de vraagtekens. Ga om vraag 21 te kunnen beantwoorden allereerst na wat er onder de vraagtekens moet staan. Bedenk vervolgens hoe de lower bounds berekend worden en werk tot slot Branch and Bound verder uit.

Tabel 3: vereiste studiejaren per student per functie

	Functie A	Functie B	Functie C	Functie D
	9	2	7	8
	6	4	3	7
	5	8	1	8
	7	6	9	4



Bijlage B: Correctievoorschrift

Een aantal opmerkingen vooraf:

- Waar om een correcte berekening/toelichting, hoeft het niet zo gedaan te zijn als in dit antwoordmodel, zolang duidelijk wordt hoe de leerlingen op hun antwoord zijn gekomen.
- Vragen of delen van vragen die niet erg lastig zijn en/of waarvan de antwoorden op deze vragen later in de opdracht sowieso nog uitgelegd worden, zijn slechts 1 punt waard.
- Vragen of delen van vragen waarbij van de docent verwacht wordt dat hij leerlingen door deze vragen loodst met extra uitleg als zij er niet uit komen, zijn eveneens slechts 1 punt waard. Hieronder een lijst van de vragen waar de docent de leerlingen doorheen moet loodsen (omdat dit essentieel is voor het verdere begrip van de opdracht) als zij er niet zelf uitkomen:
    - Vraag 1
    - Vraag 2
    - Vraag 3
    - Vraag 9
    - Vraag 12 -> antwoord in vraag gegeven, laten zien dat het klopt
    - Vraag 13 -> aangegeven in vraag: check antwoord bij docent
    - Vraag 14
    - Vraag 15
    - Vraag 17 (niet te snel helpen, eerst leerlingen aansporen nog eens goed terug te lezen)
    - Vraag 18 (alleen deel 1 van de vraag) (niet te snel helpen, eerst leerlingen aansporen nog eens goed terug te lezen)

Vraag 1) Uit Tabel 1 volgt dat de machine nooit alle taken voor hun vervalmomenten uit kan voeren. Leg dit uit.

Vraag 1: 1 punt voor een correcte uitleg

De gezamenlijke duur van de taken is 4 + 2 + 6 + 5 = 17 terwijl de vervalmomenten van alle taken kleiner zijn.

Vraag 2) Als we de taken uit Tabel 1 inplannen in numerieke volgorde, dus in de volgorde 1-2-3-4, wat is dan het maximale aantal tijdseenheden die tussen de afronding en het vervalmoment van een taak zitten?

Vraag 2: 1 punt voor het correcte antwoord met een correcte berekening/toelichting

Het max. aantal tijdseenheden die tussen de afronding en het vervalmoment van een taak zitten is 17 - 10 = 7, zie onderstaande tabel met planning.

Taak	Afrondmoment	Vervalmoment	Aantal tijdseenheden tussen afronding en vervalmoment
1	4	8	nvt
2	6	12	nvt
3	12	11	1
4	17	10	7

Vraag 3) Wat is het kleinst mogelijke maximale aantal tijdseenheden die tussen de afronding en het vervalmoment van een taak uit Tabel 1 zitten? Leg uit hoe jullie aan dit antwoord gekomen zijn en geef een optimale planning.

Vraag 3: 1 punt voor het correcte antwoord voorzien van enige toelichting

Max. aantal = 5 en de optimale planning is 1-3-4-2. Hieraan kun je komen door verschillende mogelijkheden na te gaan.

NB. Geen punten toekennen voor 1-4-3-2, dit is niet optimaal vanwege het vervalmoment van taak 4

NB. Wel punten toekennen als verderop in het verslag geconcludeerd wordt dat het antwoord op vraag 3 fout is, het correcte antwoord te vinden is in de Branch and Bound uitwerking verderop en daarbij terugverwezen wordt naar deze vraag.

Vraag 4) Voor hoeveel mogelijke planningen moet de computer L_{max} uitrekenen in ons voorbeeld als we de hierboven beschreven strategie toepassen? Laat jullie berekening zien.

Vraag 4: 1 punt voor het correcte antwoord met een correcte berekening

Aantal verschillende manieren waarop de taken ingepland kunnen worden is 4! = 24.

Vraag 5) Stel dat we in opdracht van een fabrikant een optimale planning willen vinden voor de 20 taken die zijn machine moet uitvoeren. Neem aan dat de computer waarover we beschikken vijftig nanoseconden (een nanoseconde is één-miljardste van een seconde) nodig heeft om L_{max} uit te rekenen voor een mogelijke planning. Hoe lang heeft de computer dan nodig om met de hierboven beschreven strategie een optimale planning voor de 20 taken te vinden? Geef jullie antwoord in een zo groot mogelijke doch logische tijdseenheid.

Vraag 5: 2 punten voor het correcte antwoord met een correcte berekening

Aantal verschillende manieren waarop de taken ingepland kunnen worden is 20!. 20! x 50 = 121645100408832000000 nanoseconds, oftewel 3857 jaar (3855 Gregoriaanse jaren).

NB. 1 punt aftrek voor een correct antwoord in een andere tijdseenheid dan jaren (decennia, eeuwen en millennia mogen wel, mits correct afgerond).

NB. ½ punt aftrek voor een nagenoeg correcte berekening maar een fout antwoord, bijv. door een verkeerd aantal nullen achter de komma bij het rekenen met nanoseconden.

NB. Geen punten aftrek voor een iets preciezere of iets minder precies antwoord bijv. door schrikkeljaren mee te nemen in de berekening.

Vraag 6) Pas de hierboven beschreven heuristiek toe op de vier taken uit Tabel 1. Geeft de heuristiek een goede oplossing?

Vraag 6: 1 punt voor het correcte antwoord met een correcte berekening/toelichting

Ja, aangezien we met deze heuristiek de optimale oplossing uit vraag 3 vinden.

Vraag 7) Verzin voor de vier taken uit Tabel 1 nieuwe lanceer- en vervalmomenten zodanig dat de zwakte van deze heuristiek ten opzichte van het nagaan van alle oplossingen duidelijk wordt.

Vraag 7: 2 punten voor een correct antwoord (liefst voorzien van toelichting)

Veel mogelijkheden, per antwoord checken of het klopt. Voorbeeld van een correct antwoord:

i	1	2	3	4
p_i	4	2	6	5
r_i	1	0	0	0
d_i	5	20	20	20

Vraag 8) Bedenk een andere mogelijke heuristiek die toegepast kan worden op het planningsprobleem en pas deze toe op de vier taken uit Tabel 1.

Vraag 8: 2 punten als volgt toegekend:

Heuristiek duidelijk uitgelegd (1 punt).

Heuristiek correct toegepast op de vier taken uit Tabel 1 (1 punt).

Vraag 9) Bepaal op een systematische manier alle mogelijke manieren waarop de taken uit Tabel 1 ingepland kunnen worden. Neem hiertoe onderstaande figuur over in jullie verslag en maak deze af. (Hint 1: Het is waarschijnlijk handig om een of twee hele pagina's te reserveren voor deze opgave en jullie papier in de breedte te gebruiken.) (Hint 2: de kleuren van de ovalen mogen jullie voorlopig negeren.)

Vraag 9: 1 punt voor enigszins nette en overzichtelijke, maar vooral correcte tekening

Zie de tekening(en) bij vraag 15.

NB. Geen punten aftrekken indien ovalen in correcte generatie zitten maar in een andere volgorde staan (1234 en 1243 verwisseld bijvoorbeeld).

Vraag 10) Wat voor overeenkomsten zien jullie tussen jullie antwoord op vraag 4 en de bovenstaande figuur?

Vraag 10: 1 punt voor een correct antwoord.

Het aantal ovalen in de onderste generatie is gelijk aan het aantal mogelijke oplossingen.

Vraag 11) Stel dat er in plaats van vier taken vijf taken ingepland moeten worden. Hoeveel ovalen krijg je dan als je een figuur tekent zoals die in vraag 9? (Hint: Het is niet nodig om deze figuur te tekenen en dit doen levert geen punten op, je kunt het antwoord beredeneren.)

Vraag 11: 2 punten als volgt toegekend:

In de bovenste generatie bevindt zich 1 ovaal.

In de volgende generatie bevinden zich 5 ovalen.

In de volgende generatie bevinden zich 5 x 4 ovalen. (1 punt)

In de volgende generatie bevinden zich 5 x 4 x 3 ovalen.

In de onderste generatie bevinden zich 5 x 4 x 3 x 2 ovalen.

Het totaal aantal ovalen is dan 1 + 5 + 5 x 4 + 5 x 4 x 3 + 5 x 4 x 3 x 2 = 206

Vraag 12) Plan de taken uit Tabel 3a zodanig in dat L_{max} zo klein mogelijk is. Sta hierbij pre-emption toe. Als het goed is, vinden jullie $L_1 = L_3 = 1$ en $L_2 = 0$, dus $L_{max} = 1$. Mocht dit niet zo zijn, vraag de docent dan even om jullie te helpen.

Vraag 12: 1 punt voor een correcte uitwerking waaruit blijkt dat $L_1 = L_3 = 1$ en $L_2 = 0$.

NB. Leerlingen op weg helpen kan bij deze vragen door hen een tabel te laten tekenen en daarin aan te geven van welke taak er op elk minimaal tijdsinterval (0-1, 1-2, etc.) een “stukje” ingepland wordt.

Vraag 13) Door vertraging in een ander deel van de fabriek, kan er pas later met taken 1 en 2 begonnen worden. De specificaties van de taken zijn nu zoals in Tabel 3b. Verder is het nu zo dat taak 1 eerst afgerond moet worden, voordat met een andere taak begonnen kan worden (pre-emption is dus pas toegestaan zodra taak 1 is afgerond). Wat is nu de kleinst mogelijk waarde voor L_{max}? Laat jullie antwoord even checken door de docent.

Vraag 13: 1 punt voor een correcte uitwerking waaruit blijkt dat $L_{max} = 5$ ($L_1 = 0$, $L_2 = 2$ en $L_3 = 5$).

NB. Leerlingen op weg helpen kan bij deze vragen door hen een tabel te laten tekenen en daarin aan te geven van welke taak er op elk minimaal tijdsinterval (0-1, 1-2, etc.) een “stukje” ingepland wordt.

Vraag 14) Pas de heuristiek uit vraag 6 toe op de taken uit Tabel 1. Sta hierbij pre-emption toe. Wat is de waarde van de lower bound die jullie nu gevonden hebben voor dit planningsprobleem?

Vraag 14: 1 punt voor het correcte antwoord met een correcte berekening/toelichting

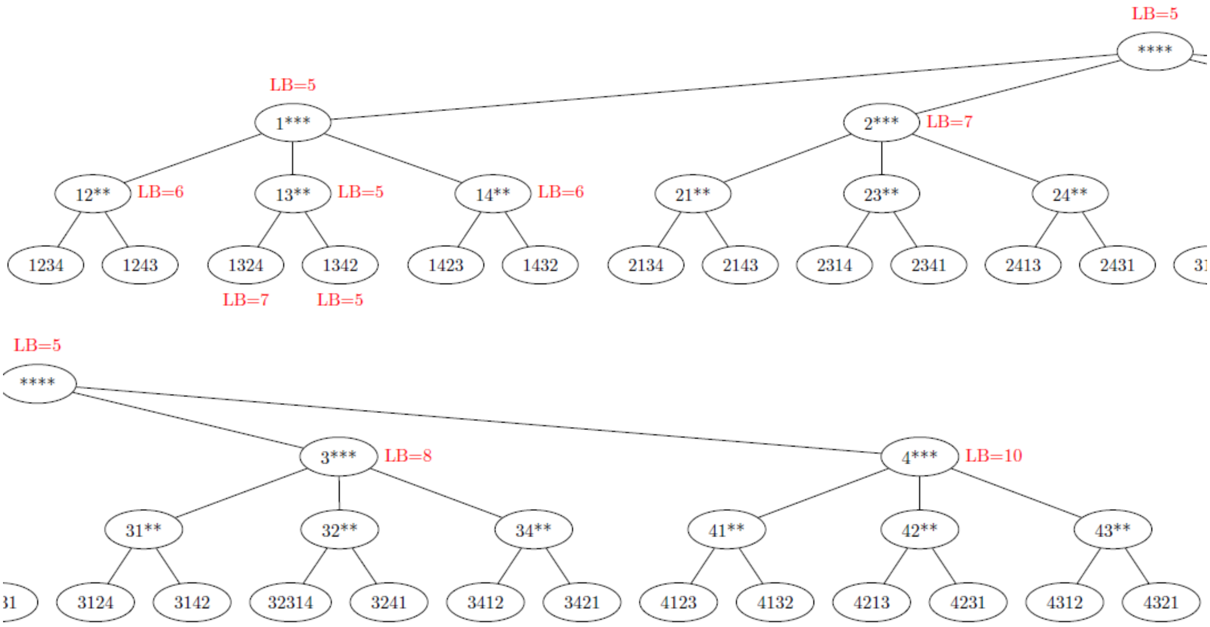
LB = 5.

Vraag 15) Ga verder met Branch and Bound voor de taken uit Tabel 1 (gebruik jullie resultaat uit vraag 9) totdat een optimale oplossing gevonden is. Bereken en noteer telkens de lower bounds bij *de ovalen waarvoor ze uitgerekend worden. Licht toe waarom de computer 'weet' dat een optimale oplossing gevonden is en hij niet verder hoeft te zoeken.*

Vraag 15: 2.5 punten toegekend als volgt:

Er zijn geen ontbrekende of overbodige vertakkingen en de LB's zijn uitgerekend voor de correcte ovalen, zie de tekening (1 punt).

De LB's zijn allemaal correct uitgerekend, zie de tekening (gesplitst in een linker- en rechtergedeelte) hieronder (1 punt).



De computer 'weet' dat een optimale oplossing gevonden is doordat er geen (onvertakte) ovalen meer zijn waarvoor de LB lager is dan de oplossingswaarde in het ovaal 1-3-4-2 (1/2 punt).

NB. Per ontbrekende of overbodige vertakking of per fout gekozen LB om uit te rekenen -1/2 met een max. van -1

NB. Per fout uitgerekende LB -1/2 met een max. van -1

Vraag 16) Welk percentage van de ovalen die jullie in vraag 9 getekend hebben hoeft de computer door het toepassen van Branch and Bound in dit probleem niet te bekijken? En voor welk percentage van de mogelijke oplossingen hoeft de computer door het toepassen van Branch and Bound in dit probleem geen oplossingswaarde te berekenen?

Vraag 16: 1 punt toegekend als volgt:

½ punt voor het correcte antwoord met een correcte berekening/toelichting van deelvraag 1: 31 van de 41 ovalen hoeft de computer niet te bekijken, dit is ongeveer 75,6%.

½ punt voor het correcte antwoord met een correcte berekening/toelichting van deelvraag 2: 22 van de 24 oplossingswaarden hoeft de computer niet te berekenen, dit is ongeveer 91,7%.

Vraag 17) Stel dat een computer bezig is met het toepassen van Branch and Bound op een planningsprobleem en aangekomen is bij de situatie die wij kunnen weergeven als in de figuur op pagina 10. De volgende stap is om de oplossingswaarde te berekenen voor het ovaal "1-4-3-2". Wat er na deze stap gebeurt, hangt af van die oplossingswaarde. Beschrijf alle mogelijke scenario's die zich voor kunnen doen en wat er op basis van die scenario's zal gebeuren (alleen de eerste stap van het vervolg).

Vraag 17: 2 punten toegekend als volgt:

Mogelijke situatie 1: De oplossingswaarde is 3, 4 of 5. In dat geval is de planning 1-4-3-2 een optimale planning en is Branch and Bound afgerond (1 punt).

Mogelijke situatie 2: De oplossingswaarde is hoger dan 5. In dat geval is de volgende stap het vertakken van ovaal 4-*-*-* (1 punt).

NB. 1 punt aftrekken als er een derde situatie bij is gekomen.

Vraag 18) Stel dat een computer bezig is met het toepassen van Branch and Bound op een planningsprobleem en aangekomen is bij de situatie die wij kunnen weergeven als in de figuur op pagina 11. Wat is nu de volgende stap? En wat is de laatste stap die gedaan is om tot de huidige situatie te komen? Zoals altijd geldt, licht jullie antwoord duidelijk toe.

Vraag 18: 2 punten toegekend als volgt:

De volgende stap is het vertakken van het ovaal "4-1-*-*" (1 punt).

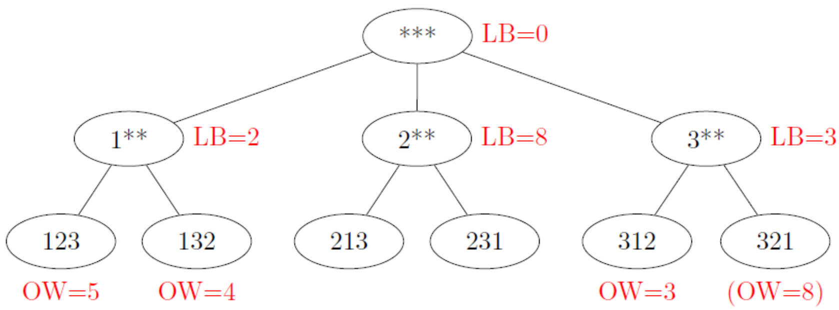
De voorgaande stap was het vertakken van ovaal "1-*-*-*" (1 punt).

Vraag 19) Pas Branch and Bound toe op de taken uit Tabel 2 om een optimale planning te vinden. Geef hierbij duidelijk alle stappen en de volgorde van deze stappen aan.

Vraag 19: 2 punten toegekend als volgt:

Er zijn geen ontbrekende of overbodige vertakkingen en de LB's zijn uitgerekend voor de correcte ovalen, zie de tekening (1 punt).

De LB's zijn allemaal correct uitgerekend, zie de tekening hieronder (1 punt).



NB. Per ontbrekende of overbodige vertakking of per fout gekozen LB om uit te rekenen -1/2 met een max. van -1

NB. Per fout uitgerekende LB -1/2 met een max. van -1

NB. Begrip van hoe een computer denkt komt naar voren: +0.5 punten

Vraag 20) Hoe efficiënt Branch and Bound werkt ten opzichte van een computer domweg alle mogelijke oplossingen te laten bekijken, hangt voornamelijk af van hoe "scherp" de lower bounds zijn die berekend worden. Licht toe wat volgens jullie wordt bedoeld met het woordje "scherp" in deze zin.

Vraag 20: 1 punt voor een duidelijke en correcte uitleg

Lower bounds moeten zo hoog mogelijk zijn (maar nog wel de eigenschap van bound bewaren in die zin dat een optimale oplossing nooit lager kan zijn dan een lower bound).

Vraag 21) Inspecteer het probleem en de gedeeltelijke uitwerking met Branch and Bound op de volgende pagina. Leg uit welke manier van vertakken en welke manier om lower bounds te bepalen gebruikt zijn. Werk vervolgens Branch and Bound verder uit in jullie verslag. Geef hierbij duidelijk alle stappen en de volgorde van deze stappen aan.

Vraag 21: 4 punten toegekend als volgt:

De manier van vertakken is correct en duidelijk uitgelegd: in de eerste vertakking wordt persoon 1 toegewezen aan een functie, in de tweede vertakking wordt persoon 2 toegewezen aan een nog niet bezette functie, etc. (1 punt).

De manier van lower bounds bepalen is correct en duidelijk uitgelegd: aan iedere student wordt de functie die van hem of haar het minste aantal studiejaren vereist toegewezen, mits deze functie nog beschikbaar is (1 punt).

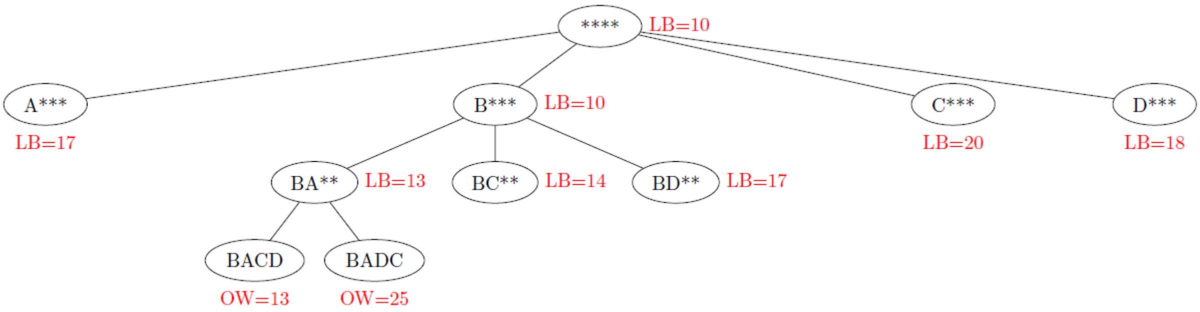
Er zijn geen ontbrekende of overbodige vertakkingen en de LB's zijn uitgerekend voor de correcte ovalen, zie de tekening hieronder (1 punt).

De LB's zijn allemaal correct uitgerekend, zie de tekening op de volgende pagina (1 punt).

NB. In de houdt de positie van de letter verband met de positie van de student in Tabel 3 uit de praktische opdracht. Zo staat BA** voor de situatie waarin persoon 1 functie B toegewezen krijgt, persoon 2 functie A toegewezen krijgt en personen 3 en 4 nog geen functie toegewezen hebben gekregen.

NB. Per ontbrekende of overbodige vertakking of per fout gekozen LB om uit te rekenen -1/2 met een max. van -1

NB. Per fout uitgerekende LB -1/2 met een max. van -1


****
LB=10
A***
LB=17
B***
LB=10
C***
LB=20
D***
LB=18
BA**
LB=13
BC**
LB=14
BD**
LB=17
BACD
OW=13
BADC
OW=25

Bijlage C: Richtlijn voor de interviews

Opmerkingen vooraf

- Dit gesprek heeft geen invloed op je beoordeling. Ik zal zelfs het verslag van jouw groepje niet nakijken, dit doet meneer Timmer. Je kan dus gewoon alles zeggen wat je vindt.
- Het interview wordt opgenomen, achteraf uitgeschreven en anoniem opgeslagen.
- Het interview zal ongeveer een kwartier tot een half uur duren, je mag het ook afbreken wanneer je wilt.
- De interviews zijn erop gericht de opdracht goed te kunnen evalueren. Aan de hand daarvan kunnen wij bepalen of de opdracht geschikt is om vaker te geven en, mocht dit laatste het geval zijn, op welke punten de opdracht eventueel verbeterd kan worden.

Vragen en onderwerpen in het interview

Eerst heb ik een paar algemene vragen over de opdracht, vervolgens komen er aantal vragen die iets dieper op het onderwerp van de opdracht ingaan.

- Vond je het een leuke opdracht? -> je mag eerlijk zijn, graag zelfs
- Moest je je haasten om de opdracht af te krijgen?
- Wat vond je het lastigste aan de opdracht?
- Waarom snapte je bepaalde opdrachten niet (eventueel de opdracht erbij houden en doorbladeren tijdens deze vraag)?

Dan komen nu de vragen over het onderwerp zelf.

- Welk probleem wilden we oplossen (in de eerste twee delen van de opdracht)?
- Waarom gebruiken we Branch and Bound, een best wel omslachtige methode, om dit probleem op te lossen?
    - Wat is de meest voor de hand liggende methode om zo’n probleem op te lossen?
    - Wat is er mis met deze meest voor de hand liggende methode? (Computers zijn toch heel snel, hoezo werkt die meest voor de hand liggende methode dan niet?)
- Heuristieken zijn tijdens de opdracht ook aan bod gekomen. Kun je een voor- en een nadeel noemen van heuristieken?

- Welke twee aspecten heb je nodig om Branch and Bound toe te kunnen passen op een probleem?

Tot slot wil ik je vragen het volgende planningsprobleem met Branch and Bound op te lossen. Spreek daarbij alle gedachten die je hebt en stappen die je doet hardop uit:

i	1	2	3
p_i	5	6	5
r_i	1	7	2
d_i	15	11	12

Bijlage D: Transcripties van de interviews

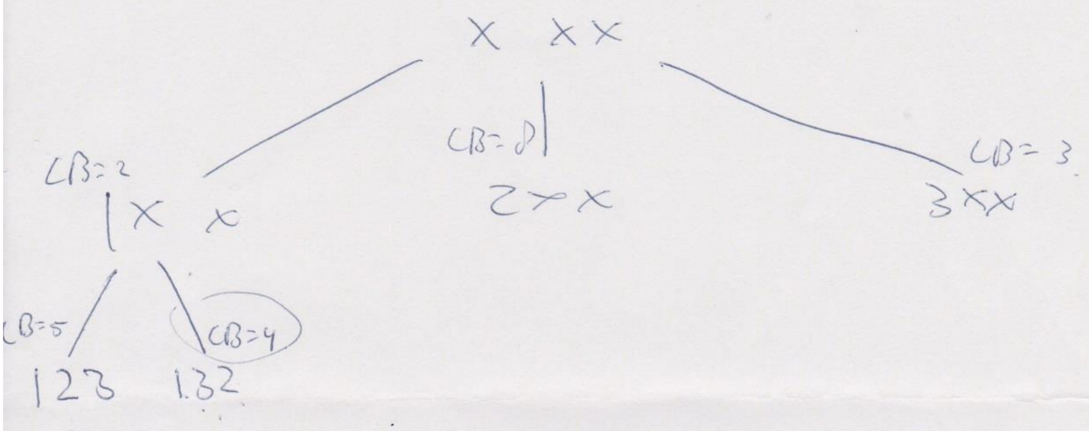
Interview met leerling 1

ONDERZOEKER	Vond je het een leuke opdracht?
LEERLING	Ja, er zat wel uitdaging in en het was eens een keer wat anders dan het normale vergelijkingen oplossen. Je moest vooral even nadenken.
ONDERZOEKER	Moesten jullie je haasten om de opdracht af te krijgen?
LEERLING	Ja, in het begin snapten we het nog niet helemaal. Daar hebben we wel even wat tijd aan besteed. Dus aan het eind moesten we binnen een korte tijd een groot aantal opdrachten maken.
ONDERZOEKER	Maar jullie hebben het wel afgekregen?
LEERLING	Ja.
ONDERZOEKER	Wat vonden jullie het lastigste aan de opdracht?
LEERLING	Ik denk deel 3, vooral vraag 21. Daar snapten we niet zo heel veel van.
ONDERZOEKER	Zou je nog enigszins kunnen toelichten waarom je dat niet snapte?
LEERLING	Ik denk dat het toch wel een stapje moeilijker was dan de voorgaande opdrachten. Aan het begin werd er gegeven wat er in de ovalen moest staan en dat moest je nu zelf gaan verzinnen. De eerste keer is altijd het moeilijkst en als je dat dan ook nog eens na twee uur in een warm lokaal moet doen, is het een beetje lastig.
ONDERZOEKER	Dat waren de algemene vragen. Nu komen er een aantal vragen over het onderwerp zelf. Welk probleem waren we aan het oplossen in de eerste twee delen van de opdrachten?
LEERLING	Hoe je zo snel mogelijk taken kan oplossen, hoe je de optimale snelheid kunt vinden hoe je iets op moet lossen.
ONDERZOEKER	Zou je 'optimale snelheid' nog iets verder kunnen specificeren?
LEERLING	Nee.
ONDERZOEKER	We gebruikten in de opdracht Branch and Bound om planningsproblemen aan te pakken. Branch and Bound is best wel een omslachtige methode, hoezo gebruikten we die eigenlijk?
LEERLING	Ik denk om een helder beeld voor de leerlingen te krijgen.
ONDERZOEKER	Wat is de meest voor de hand liggende methode om zo'n planningsprobleem te optimaliseren?
LEERLING	In een computer invoeren.
ONDERZOEKER	En wat laat je de computer dan doen?
LEERLING	De optimale snelheid uitrekenen.
ONDERZOEKER	Zou je kunnen toelichten hoe de computer dat dan zou doen? Iemand moet de computer immers programmeren. Als die persoon een zo mogelijk mogelijk programma wil schrijven, wat laat ie de computer dan doen?
LEERLING	Gewoon iets toepassen om de tijden uit te rekenen.
ONDERZOEKER	Van elke mogelijke oplossing gaat hij dan de oplossingswaarde berekenen en vervolgens de laagste zoeken. Welk nadeel kleeft er aan deze methode van alle oplossingen bekijken en vervolgens de beste selecteren?
LEERLING	Het duurt nogal lang.

ONDERZOEKER	Ja, en dat is de reden waarom we Branch and Bound toepassen. Kun je een voor- en een nadeel noemen van heuristieken, als je nog weet wat dat zijn?
LEERLING	Een heuristiek was toch de kortste tijd uitrekenen, of zit ik nou fout?
ONDERZOEKER	Kun je dat iets specificeren?
LEERLING	Gewoon L_{max} uitrekenen toch?
ONDERZOEKER	Daarvoor gebruikten we inderdaad soms een heuristiek. Heuristiek is echter een breder begrip. Kun je je dat nog herinneren en daar een voordeel van noemen ten opzichte van optimale oplossingen berekenen?
LEERLING	Dat het een stukje sneller gaat. Dat je er niet 3000 jaar mee bezig bent.
ONDERZOEKER	Kun je ook een nadeel bedenken van heuristieken?
LEERLING	Nee.
ONDERZOEKER	Als je een heuristiek toepast, dan weet je niet zeker of je de optimale oplossing vindt. Maar als je een heuristiek gebruikt samen met Branch and Bound, dan weet je zeker dat je een optimale oplossing vindt en dat dit sneller gaat dan alle mogelijkheden nagaan. Weet je nog welke twee aspecten je nodig hebt om Branch and Bound toe te kunnen passen?
LEERLING	De L_{max} en de [lange pauze].
ONDERZOEKER	Laten we eens één stapje teruggaan. L_{max} zat in de goede richting. We hebben Branch and Bound toegepast op planningsproblemen, maar Branch and Bound kun je toepassen op veel verschillende problemen. Dan heb je dus geen L_{max}, die was namelijk echt gericht op de vertraging van een taak op een machine. Zou je dat kunnen generaliseren naar andere problemen? Voor een probleem heb je allereerst nodig dat je takjes kunt maken [*beeldt uit hoe takjes gemaakt worden], dat je kunt branchen, en wat is dan het andere dat je nodig hebt?
LEERLING	Ja, [pauze] verschillende mogelijkheden [*beeldt uit hoe takjes gemaakt worden].
ONDERZOEKER	Ja, verschillende mogelijkheden voor dat branchen. En dan moet je iets uitrekenen voor die nieuwe ovalen zeg maar, die L_{max}. Dat is de lower bound, waardoor je weet op een gegeven moment, oh, hier hoeven we niet verder te zoeken.
LEERLING	Ja ja ja.
ONDERZOEKER	Ok, dat was het tweede deel van het interview. Als laatste wil ik jou vragen om, hier staat een planningsprobleem zoals je het gister gezien hebt, zou je kunnen proberen dit met Branch and Boun op te lossen?
LEERLING	Ik ga mijn best doen.
ONDERZOEKER	En tijdens dat je dat doet, zo veel mogelijk de stappen die je zet en de gedachten die je hebt hardop uitspreken zodat achteraf te zien is hoe dat gaat in je hoofd?
LEERLING	Ja. Eerst gewoon een plaatsje voor dat hoofdding [*tekent eerste ovaal]. Dan in drieën splitsen [*tekent drie lijnen en drie ovalen], branchen. En dan ga je dus hier verder met [onverstaanbaar gemompel] [*tekent twee lijnen en schrijft de twee planningen 123 en 132 op]. [*tekent twee lijnen en wil de planningen 213 en 231 opschrijven] Nou, dan gewoon gelijk zo.
ONDERZOEKER	Ja, je maakt nu die hele tekening, maar stel nou dat jij een computer was die dit probleem aan zou pakken en het probleem was ook veel groter dan drie takken. Dan begin je niet met eerst die hele tekening te maken.

LEERLING	Nee, dan begin je gewoon met ergens de lower bound uit te rekenen [*wijst naar tweede rij ovalen].
ONDERZOEKER	Ja, laten we dat doen. Laten we net doen alsof we een computer zijn en dit probleem oplossen, dus we beginnen inderdaad bovenaan.
LEERLING	Ja, ja, ik weet niet meer hoe dat moet.
ONDERZOEKER	Ok, dat is niet erg. Dan voeren we het niet uit. Maar je zei net al, de volgende stap is die drie uitrekenen [*wijst naar tweede rij ovalen], wat ga je dan doen voor die drie?
LEERLING	Ja, de lower bounds uitrekenen. Kijken bij welke tak die het laagste is en dan ga je met die tak gewoon verder.
ONDERZOEKER	Ja, inderdaad. Ik laat je zien wat er uit zou komen. We hebben voor die drie [*wijst naar tweede rij ovalen] uitgerekend dat de lower bound hier 2 is, hier 8 en hier 3.
LEERLING	[*Schrijft lower bounds bij de ovalen]
ONDERZOEKER	Goed dat je het erbij schrijft.
LEERLING	Dan ga je dus met deze verder [*wijst naar eerste ovaal in tweede rij].
ONDERZOEKER	Ja.
LEERLING	Dan heb je dus bij deze tak [*wijst naar eerste ovaal in tweede rij] twee mogelijkheden en dan ga je dus bij deze [*wijst naar planningen 123 en 132] nog de lower bound uitrekenen en waar die dan het laagst is daar is dan de optimale [onverstaanbaar gemompel] [*wijst naar planning 123].
ONDERZOEKER	Ok, dat heb ik ook. Hier [*wijst naar planning 123] is ie 5 en hier [*wijst naar planning 132] 4. Dat is dan de lower bound inderdaad. Kun je hier nog iets meer over zeggen, over die lower bound in dit geval?
LEERLING	Dat dat dan de goede volgorde is om de taken uit te voeren.
ONDERZOEKER	Ja, de lower bound is hier [*wijst naar planningen 123 en 132] altijd gelijk aan de oplossingswaarde. Jij zegt dat hier [*wijst naar planning 123] de beste is, maar zou er niet nog een betere oplossing hier [*wijst naar gebied onder tweede ovaal in tweede rij] of hier hier [*wijst naar gebied onder derde ovaal in tweede rij] kunnen zitten?
LEERLING	Nee, dat denk ik niet. Want je moet sowieso met tak 1 beginnen omdat als je met tak 2 begint dan heb je in ieder geval hier [*wijst naar tabel, naar de release dates] een veel te lange tijdsduur en hier begin je gelijk op tijdstap 1.
ONDERZOEKER	Ok. Dit was de laatste vraag van het interview.

i	1	2	3
p_i	5	6	5
r_i	1	7	2
d_i	15	11	12



Interview met leerling 2

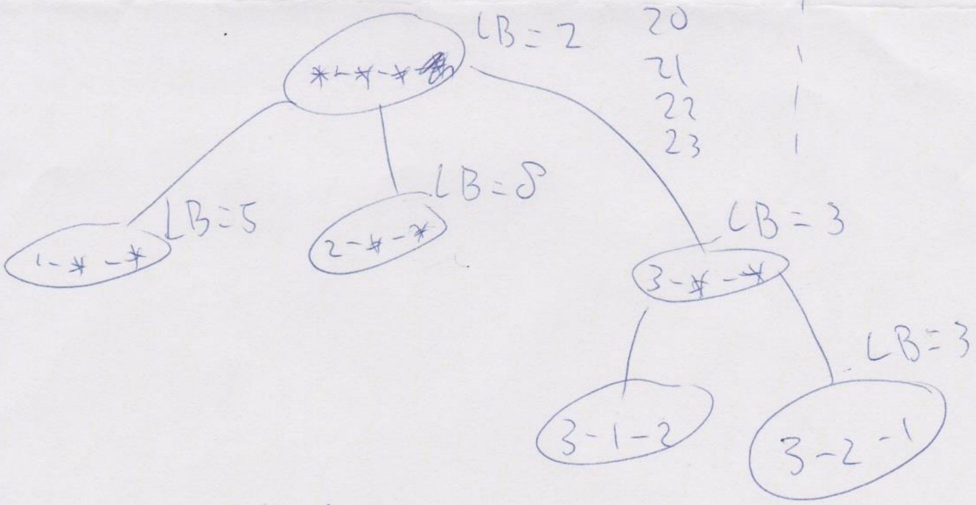
ONDERZOEKER	Vond je het een leuke opdracht?
LEERLING	Ja, ik vond het sowieso wel een leuke opdracht. Ik vond het wel interessant zeg maar, gewoon een leuk onderwerp om iets nieuws te leren.
ONDERZOEKER	Vergeleken met gewone wiskundelessen?
LEERLING	Ja, dat was wel leuk. Van die aparte opdrachten, een beetje puzzelen, daar houd ik wel van.
ONDERZOEKER	Moesten jullie je haasten om de opdracht af te krijgen?
LEERLING	Ja, dat sowieso. Tenminste, als je gewoon rustig de tijd nam, dat het ook wel lukte, maar ik had wel het gevoel dat je je echt moest haasten en dat had denk ik iedereen wel een beetje.
ONDERZOEKER	Wat vonden jullie het lastigste aan de opdracht?
LEERLING	Het lastigste vond ik vooral dat je geen uitleg kreeg. Je zat al een beetje met de tijdsdruk en dat je dan iets moest lezen, meteen leren, en dat je zonder iets uitgelegd te krijgen het ook meteen moest toepassen, dat vond ik denk ik lastig. Het was niet dat het onderwerp lastig was, maar dat je meteen zonder dat het uitgelegd wordt, het meteen al moet toepassen, dat vond ik het lastigste. Dat je eerst even de tijd moet nemen, terwijl je het gevoel hebt dat je die tijd niet hebt en dat je het dan meteen moet maken.
ONDERZOEKER	Er stond op zich wel wat uitleg in het boekje, denk je dat dat voldoende was?
LEERLING	Oh ja, ik denk dat het wel voldoende was, maar meer dat je, en ik denk dat heel veel mensen dat hadden, ook een beetje stress hadden dat ze niet binnen de tijd zaten en dan is het moeilijk om even rustig de tijd te nemen om te lezen zeg maar, dus ik denk dat de uitleg die daar stond op zich wel goed was, maar dan moest je wel even weten dat je je kan concentreren zeg maar en de meeste mensen kunnen dat niet echt onder stress. Ik denk, de uitleg zelf was inderdaad wel goed.
ONDERZOEKER	Ok. Was er ook nog iets uit het verslag waarvan je dacht, dat is eigenlijk wel makkelijk of wel logisch?
LEERLING	Mmm…
ONDERZOEKER	Iets waarvan je dacht, dat had niet per se in de opdracht gehoeven of dat heeft achteraf gezien niet zoveel toegevoegd?
LEERLING	Nee, dat weet ik niet, denk het niet echt. Ja, ik vond alle vragen wel goed. Ik kan nog wel even kijken [*bladert door opdrachtenboekje]. Ik kan me niet herinneren dat er een vraag was waarvan ik dacht ja, die vind ik wel makkelijk. Ik vond zelf die vraag dat je moest uitrekenen hoeveel mogelijke planningen er waren, vond ik heel makkelijk, maar ik hoorde om me heen dat sommige mensen wel vier tot de macht vier hadden gedaan, dus ja. Dus dat is denk ik ook wel goed, nee, ik denk niet dat er echt dingen waren.
ONDERZOEKER	Ok, dan gaan we naar de vragen over het onderwerp. Allereerst, welk probleem waren we aan het oplossen in de eerste twee delen van de opdrachten?
LEERLING	Dat je de beste planning zoekt. Het probleem dat je wilt weten wat de beste planning is maar dat dat, als je dat dus gaat uitrekenen voor veel verschillende taken dat het veel te lang kost, te veel tijd kost.
ONDERZOEKER	Welk criterium hingen we daaraan? We wilden de beste planning, dat zei je net al, die wilden we vinden, maar wat vonden we dan een goede planning?

LEERLING	Dat was dat die L-max, wanneer niet één taak heel erg laat was, dan was het een goede planning.
ONDERZOEKER	We gebruikten vervolgens Branch and Bound om dat probleem op te lossen, maar dat was best wel omslachtig. Waarom gebruikten we dat eigenlijk?
LEERLING	Omdat je dan een heel groot deel niet hoeft uit te rekenen, waardoor dat heel veel tijd scheelt eigenlijk. Je rekent uiteindelijk maar een kwart uit ofzo en dat scheelt dus wel heel veel.
ONDERZOEKER	Verder is in de opdracht het begrip heuristiek aan bod gekomen. Kun je je nog herinneren wat dat is en er een voor- en een nadeel van noemen?
LEERLING	Ja, dat was toch dat je niet per se de beste oplossing gaat uitrekenen, maar dat ie dat op een bepaalde andere manier deed, dus bijvoorbeeld, eh, een heuristiek is dat ie dan sorteerde op lanceermoment en dan van laag naar hoog of, eh, hoe heet dat, vervaltijd. En daar het voordeel van was dat je dus niet alles uit hoeft te rekenen maar dat je denkt, nou, we doen dit maar, maar een nadeel was dat je dus niet per se de beste oplossing hebt.
ONDERZOEKER	We hebben Branch and Bound toegepast op planningsproblemen, maar Branch and Bound kun je toepassen op veel verschillende problemen. Welke twee aspecten heb je nodig voordat je Branch and Bound kunt toepassen?
LEERLING	Ehm, hoe bedoel je dat?
ONDERZOEKER	Je moet een probleem in een bepaalde vorm zetten en vervolgens moet je twee dingen kunnen bepalen. Je hebt allemaal oplossingen, maar ook deeloplossingen [*wijst naar deeloplossingen in voorbeeldprobleem] en je moet twee dingen kunnen bepalen om Branch and Bound toe te passen.
LEERLING	Die vraag snap ik niet helemaal.
ONDERZOEKER	Uiteindelijk ben je toegekomen aan het eind, aan vraag 21?
LEERLING	Ja.
ONDERZOEKER	Daar pasten we Branch and Bound toe op een ander probleem, met die studenten en die functies, en toen moest je op zoek gaan naar twee "dingen". Een deel was al getekend, maar er stonden nog wat vraagtekens, en jij moest gaan zoeken naar wat er onder die vraagtekens staat en nog iets moest je uitvinden.
LEERLING	Ja, ik moest, wat was het ook alweer, je moest weten, ja. Als je Branch and Bound toepast moet je weten wat, ehm, nog één keer de vraag. Twee aspecten die je nodig hebt om Branch and Bound toe te kunnen passen?
ONDERZOEKER	Ja.
LEERLING	Nou, je moet weten wat de optimale kan zijn, bedoel je zoiets, of de lower bound zeg maar. Dus dat, is dat dan een aspect zeg maar.
ONDERZOEKER	Ja. Je moet inderdaad een lower bound kunnen bepalen.
LEERLING	En dan moet je dus nog door kunnen rekenen naar wat de uiteindelijke oplossing is, zeg maar.
ONDERZOEKER	Ok mooi, dat was het tweede deel van het interview. Het derde deel is dit. Een probleem zoals waar je gisteren mee bezig bent geweest [*pakt planningsprobleem in tabelweergave erbij] en mijn vraag is nu of jij kunt proberen Branch and Bound hierop toe te passen en terwijl je hiermee bezig bent je gedachtestappen en wat je opschrijft hardop uit te spreken.
LEERLING	Nou hoe ik het deed, was een tabel maken, qua tijd zeg maar, hier dus 1 tot en met 15 en dan kan ik hier verschil maken. Oh, eigenlijk tot meer dan 15, ja [*schrijft 1 tot en met 23 onder elkaar]. En dan ging ik dus gewoon kijken, bij die

	bovenste ovaal kijken wat de allerbeste was en dan ging ik dus gewoon invullen. Nou in de eerste kun je nog niks, dus dan zet ik daar gewoon een kruisje neer [*zet x achter de 1] en dan 1 [*zet 1 achter de 2], en nu kun je dus al weten dat ie tot 17 komt omdat je die eerste niks hebt en daarna moet je dus 16 tijdseenheden, dus zo wist ik dat je tot 17 moest doorschrijven en doordat die de hoogste vervaltijd heeft, moest hier sowieso al 1 komen en dan heb je de laagste L-max zeg maar. Zo dacht ik dat dan [*vult de tweede kolom van de "tabel" verder in waarbij pre-emption correct wordt toegepast] die is dan 2 te laat [*wijst met pen naar 13e rij van tabel waar de laatste 2 in de tweede kolom staat] en dan is hier dus de LB, ehm ja, die is 2 want hier is die ook 2 te laat toch [*wijst met pen naar 17e rij van tabel waar de laatste 1 in de tweede kolom staat] Ja, dus die is 2 [*tekent eerste ovaal en noteert er LB=2 bij, begint vervolgens aan derde kolom]
ONDERZOEKER	Ok, zeg even hardop wat je aan het doen bent.
LEERLING	Zometeen de LB uitrekenen voor wanneer de 1 als eerste begint, en dan waneer de 2 als eerste begint en de rest dus met die pre-emption nog steeds kan, en daarna 3, maar ja, eigenlijk weten we nu al wel dat bij 1, want als je, dat sowieso 1 als eerste moet, ten minste 2 niet in ieder geval want als je moet wachten tot 2 klaar is dan, nou ja, dan ben je wel heel lang bezig, maar dit moet je natuurlijk wel uitrekenen [*mompelt] Nou ja, nu hoef ik het ook niet verder te doen, want dit is gewoon 5 [*noteert LB=5 bij 1-*-*] en dan dus die 2, die mag pas hier beginnen [*begint aan vierde kolom]. Dus dan kom je helemaal tot, nou eerst weer invullen, nou je komt tot 21, klopt dat, nee [*maakt vierde kolom af], ja nu klopt ie en dan is de LB, die is dan, ehm, 8 [noteert LB=8 bij 2-*-*]. En dan met die 3 hetzelfde [*vult vijfde kolom in] en hier is die 3 [*noteert LB=3 bij 3-*-*]. En dan dus met deze [*wijst naar ovaal 3-*-*] verder rekenen want de LB bij die is het laagste [*tekent ovalen 3-1-2 en 3-2-1] en je kan meteen die erbij zetten [*wijst op de 2 in 3-1-2 en de 1 in 3-2-1] want als we er een sterretje neer zetten is het de enige mogelijke oplossing dan. En dan deze opschrijven, maar deze heb ik eigenlijk al hier [*wijst naar ovaal 3-2-1 en naar de vijfde kolom] dus deze is 3 [noteert LB=3 bij 3-2-1] en dan is dit dus de beste, want hier [*wijst naar 3-1-2] hoef je ook niet meer te rekenen want, nou ja, hij kan misschien ook op 3 uitkomen, nou ja dat kan dus niet, maar dat zou misschien kunnen, maar ja dan is ie dus gelijk aan die en dan is het allebei een even goede planning. Dus nu heb je hem eigenlijk al.
ONDERZOEKER	Ja, bedankt.

i	1	2	3
p_i	5	6	5
r_i	1	7	2
d_i	15	11	12

1	x	x		
2	1	1		
3	3	1		3
4	3	1		3
5	3	1		3
6	3	1		3
7	3			3
8	2		2	2
9	2		2	2
10	2		2	2
11	2		2	2
12	2		2	2
13	2		2	2
14	1		3	1
15	1	3	3	1
16	1	3	3	1
17	1	3	3	1
18			3	1
19			1	
20			1	
21			1	
22			1	
23			1	



Interview met leerling 3

ONDERZOEKER	Vond je het een leuke opdracht?
LEERLING	Jawel. Het was wel anders dan alle andere opdrachten zeg maar en ja, het geeft wel iets meer verdieping over dingen die je allemaal nog niet weet. Dat is wel leuk.
ONDERZOEKER	Ja. En het onderwerp?
LEERLING	Ehm, ja, we hebben dat nooit gehad zeg maar, dus het is wel leuk om te weten hoe het toe passen zeg maar, de wiskunde.
ONDERZOEKER	Moesten jullie je haasten om de opdracht af te krijgen?
LEERLING	Ja, we hadden uiteindelijk opdracht 21 niet af, nou daar hadden we geen zin meer in, het duurde ook wel voor het zover was.
ONDERZOEKER	Wat vond je het lastigste aan de opdracht?
LEERLING	Het eerste deel vond ik lastig, want dan zit je er nog niet in en dan begrijp je het nog niet zo goed. Dus toen moesten we wel een aantal keren vragen voordat we eindelijk eens echt een beetje erin zaten en konden beginnen. Dat was het stomste denk ik.
ONDERZOEKER	Ja. Kun je specifiek dingen aanwijzen eruit [*overhandigt opdrachtenboekje]?
LEERLING	Ja, wij snapten allereerst tabel 1 al niet, wat al die cijfers betekenden. [*bladert door opdrachtenboekje] op een gegeven moment snapte ik iets wel en mijn teamgenootje niet, dus toen omdat we toch tijd tekort hadden, maakte ik die opdracht maar dan later. Snapte ik iets niet dan zij wel dus zo compenseerden we elkaar een beetje.
ONDERZOEKER	Dat is mooi. Was er ook een deel dat jullie makkelijk vonden of te logisch was?
LEERLING	Ik vond dit wel makkelijk [*wijst naar vraag 6 in opdrachtenboekje], met die heuristiek, maar dat vond zij dan weer lastig.
ONDERZOEKER	Ok, dan ga ik nu wat vragen stellen over het onderwerp zelf. Dan doe ik die weer dicht [*sluit opdrachtenboekje]. Welk probleem waren we aan het oplossen in de eerste twee delen van de opdracht?
LEERLING	Ehm, volgens mij in een fabriek ofzo en dan met taken die je moest uitvoeren en dan de planning daarvan. Op één machine verschillende taken.
ONDERZOEKER	En wat wilden we uiteindelijk.
LEERLING	Ja, de beste planning.
ONDERZOEKER	Ja, en wanneer was een planning de beste?
LEERLING	Met zo min mogelijk vertraging om ehm ja.
ONDERZOEKER	En wat houdt die vertraging dan in?
LEERLING	Om zo min mogelijk tijd over de, ja zeg maar de eindtijd voor wanneer die af moet zijn en dan zo min mogelijk tijd dat daar overheen ging in totaal.
ONDERZOEKER	Dus als ik twee machines heb met een vertraging van 4 of twee machines waarvan één een vertraging heeft van 1 en één van 5, wat is dan de betere?
LEERLING	Die van 1 en 5.
ONDERZOEKER	Omdat?
LEERLING	Omdat, eh, ja eigenlijk ik weet niet. Omdat bij 4 hebben ze allebei weinig maar bij 1 en 5 heb je samen 6 dus dan is dat weer minder die 8.
ONDERZOEKER	Ja, dus er valt voor allebei wel iets te zeggen ja. Weet je nog welk criterium we hebben gebruikt in de opdracht?
LEERLING	Pfff

ONDERZOEKER	Het is een van die twee. Weet je dat nog?
LEERLING	Ja, die laatste gebruikten we volgens mij, ja ik weet het niet zo goed meer.
ONDERZOEKER	Met laatste bedoel je dan die 1 en 5 samen, wat dan in totaal het laagste is?
LEERLING	Nee, of ja, die 1 die was het laagste dus dan zou je daar in ieder geval [onverstaanbaar]. Dus dan zou je daar met die laagste verdergaan dus ik denk dan die 1 en 5 want omdat die 1 en 5 dan het laagste heeft.
ONDERZOEKER	Ja. Je moet inderdaad een lower bound kunnen bepalen.
LEERLING	En dan moet je dus nog door kunnen rekenen naar wat de uiteindelijke oplossing is, zeg maar.
ONDERZOEKER	Ok. We gebruikten Branch and Bound om dat planningsprobleem op te lossen, maar dat was best wel omslachtig. Hoezo gebruikten we dan eigenlijk die methode?
LEERLING	Ehm, dat was toch met dat je de ene kon stoppen en dan verder kon gaan of was dat iets anders
ONDERZOEKER	Wat zou, als we zo'n probleem zouden willen oplossen, de meest voor de hand liggende methode zijn, waarbij we het minst logisch hoeven na te denken en toch de optimale oplossing vinden?
LEERLING	Ehm, weet ik niet meer.
ONDERZOEKER	Je zou alle mogelijke oplossingen kunnen bekijken en dan daarvan het beste kiezen. Maar dat heeft een nadeel, die methode.
LEERLING	Het is veel werk.
ONDERZOEKER	Ja. En een computer, kan die dat niet voor ons doen?
LEERLING	Jawel, maar die was er ook heel lang mee bezig.
ONDERZOEKER	Het begrip heuristiek, je noemde het net al even kort, is ook aan bod gekomen. Kun je een voorbeeld noemen van een heuristiek voor planningsproblemen?
LEERLING	Nou, op wat voor manier moet ik dan een voorbeeld geven zeg maar.
ONDERZOEKER	Zeg maar, we hebben zo'n tabelletje, zoals dit [*opent opdrachtenboekje en wijst naar tabel 1] en we willen hier een oplossing voor vinden met een heuristiek.
LEERLING	Ja, gewoon bij de eerste beginnen, en dan kijk je naar de eindtijd zeg maar en wie dan het minste is [*wijst naar taak 4 in tabel 1] alleen die mag je dan nog niet kiezen want die kan dan nog niet begonnen zijn dus dan moet je eerst naar die 3 en dan 4 en dan naar 2.
ONDERZOEKER	Ja, dan hebben we dus een mogelijke oplossing gevonden, met een heuristiek. Maar wat is nou een nadeel van het toepassen van een heuristiek, en wat is er een voordeel van?
LEERLING	Ehm, ik denk dat het sneller gaat dan als je alles gaat uitrekenen en een nadeel is wel [stilte] ja, alsnog niet de kortste tijd hebt ofzo.
ONDERZOEKER	Ok. Stel we willen Branch and Bound toepassen op een ander probleem. Dus Branch and Bound hebben we vooral in deze opdracht [*wijst naar opdrachtenboekje] op planningsproblemen toegepast, maar je kunt het op heel veel verschillende problemen toepassen. Dan heb je, om dat te doen, twee aspecten bepalen voor dat probleem. Heb je een idee van wat ik bedoel met die twee aspecten?
LEERLING	Ja, volgens mij zijn ze wel langsgekomen, maar pfff [stilte] je kijkt naar de begintijd, wanneer ie kan beginnen ofzo, nee, ik weet het niet.
ONDERZOEKER	Ok dat maakt niet uit. Dan wil ik je vragen om, hier staat een tabel [*pakt planningsprobleem in tabelweergave erbij], om hier nog eens naar te kijken en

	te proberen dit probleem op te lossen met behulp van Branch and Bound. Wil je dat proberen?
LEERLING	Ja, dit was net het gedeelte dat [] net snapte maar ok.
ONDERZOEKER	Zeg hardop wat je denkt en als je iets opschrijft, zeg dat dan ook even zodat ik het later terug kan luisteren.
LEERLING	Eerst toch gewoon zo'n boom maken? Dan begin je met de eerste met drie sterretjes en dan hier [*tekent eerste vier ovalen]. En dan moet je de LB van de eerste berekenen. Dus dan begin je met de eerste en dan mag die, even kijken [stilte] als je die gewoon door laat gaan dan is die bij de 6 en bij 6 dan gaan we verder naar die, oh nee wacht dan [gemompel] 1-3-2-3 [*noteert 1-3-2-3 onder de gegeven tabel]. 6, 7 plus 6 is 13 dus heb je hier 2 teveel, 2 te laat [*noteert 2 bij de 11] en 13 plus 4 is 17 dus heb je hier 5, dus is de LB van hier 5 [*noteert LB=5 bij *-*-*] en dan moet je de LB van die drie ballonnetjes eronder berekenen.
ONDERZOEKER	Als je een extra blaadje wilt, dan moet je het zeggen [*laat extra blaadjes zien]
LEERLING	Ok. Even kijken, dan moet je sowieso beginnen met de eerste dus hier [*wijst naar 1-*-*] en dan kun je weer dezelfde toepassen als wat je daarvoor hebt gedaan dus heb je weer 1-3-2-3 dus er komt weer 5 uit [*noteert LB=5] bij 1-*-*] en bij die tweede moet je beginnen bij de 2 dus dan, ik denk wel 13 en dan ga je naar de 3, is 2-3-1. 13 plus 5 is 18, hier 6 te veel, 18 plus 5 is 23 dus 8 teveel, dus de LB van 2 is 8 [*noteert LB=5] bij 2-*-*]. Dan de derde, moet je beginnen bij de 3. Daarna naar de 2 dan de 1. Mmm, wacht even, dan heb je 13 dus hier 2 teveel. 13 plus 5 is 18 dus hier 3 teveel, er klopt iets niet.
ONDERZOEKER	Nu zeg je, er klopt iets niet. Hoezo denk je dat?
LEERLING	Omdat ik, ja, ik dacht even dat deze nu 3, nee ehm, nu weet ik niet meer wat bij wat hoort [*wijst naar hulpgetallen die hij/zij zelf bij tabel heeft gezet]
ONDERZOEKER	Ik kan je zeggen, die [*wijst naar hulpgetallen die door leerling bij tabel zijn gezet] waren goed.
LEERLING	Ok [stilte] even kijken [gemompel] [*noteert LB=8 en LB=3 bij respectievelijk 2-*-* en 3-*-*]. En dan kijk je naar de laagste, dus dan moest je bij het ballonnetje van 3 verdergaan. Dan heb je 3-2-1 of 3-1-2 [*tekent ovalen 3-2-1 en 3-1-2]. Ok, dus dan 3, en 3-2-1 is hetzelfde als bij die eerste dus dan is die 3 [*noteert LB=3 bij 3-2-1] en dan 3-1-2 is, ehm, 7 plus 5 is 12, plus 6 is 18. De LB is 7 [*noteert LB=7 bij 3-1-2]. Dus dan is 3-2-1 is dan het antwoord [*omcirkelt 3-2-1-].
ONDERZOEKER	Ja, mooi. Ik ga nog twee dingen vragen om even naar te kijken. Allereerst dit stukje hier, dit [*wijst naar ovalen *-*-* en 3-*-*]. Zie je hier iets vreemds aan?
LEERLING	Ehm. Welk stukje bedoel je nou?
ONDERZOEKER	Dat je van die eerste rij eigenlijk naar die tweede rij gaat.
LEERLING	Ehm nou de eerste is het, oh [stilte] Dat zeg maar die [*wijst naar 3-*-*] is lager dan de eerste.
ONDERZOEKER	Ja en dat is gek, of niet?
LEERLING	Ja, dan denk ik dat ik die [*wijst naar *-*-*] fout heb.
ONDERZOEKER	Ja. Ok nou, die hoef je niet opnieuw te berekenen. Ik wil je wel vragen om die [*wijst naar 1-*-*] nog een keer opnieuw te berekenen.
LEERLING	Deze?
ONDERZOEKER	Ja. En dan duidelijk zeggen wat je doet.
LEERLING	Is die dan ook fout?
ONDERZOEKER	Wel, dat gaan we uitvinden.

LEERLING	[*lacht] Nou, je begint bij de 1. En dan heb je 7, eh 6. En dan ga je naar de 2 en dan naar de 3, dat is 6 plus 6, dan is het 12. Dus heb je hier 1 te veel [*wijst naar taak 2]. Plus 5, 17, en dan is hier 5 te veel [*wijst naar taak 3]. Dus ja, dan is dit alsnog juist [*wijst naar LB=5 bij 1-*-*].
ONDERZOEKER	Hoezo doe je na die eerste die tweede, maar niet die derde? Dat kun je waarschijnlijk verklaren, waarom je dat doet?
LEERLING	Oh nee wacht, ik zie het al. Hier had ik 6, dus dan kun je eigenlijk niet met die tweede beginnen. Dus dan heb moet je [*lacht] heb je 1-3-2-3 [*noteert 1-3-2-3 onder tabel]. Je begint bij de eerste, bij 1, omdat dat moet. Dan ga je naar 3 want 2 kan nog niet beginnen want dat mag pas bij 7, omdat ie nog maar bij 6 is. Dus doe je eerst eentje van daar, van 3, dan moet je daar nog 4 [*wijst naar taak 3 en noteert 4 naast de tabel]. Dan ga je naar 2 en maak je die af, dus heb je, dan ben je wel bij 7. En 7 plus 6 is 13, dus heb je hier 2 te veel [*noteert 2 in tabel, onderaan de kolom behorend bij taak 2]. En dan na die 13 kun je hier nog 4 doen [*wijst naar taak 3] en dan heb je 17 hier. Dus dan heb je 5 te veel [*noteert 5 in tabel, onderaan de kolom behorend bij taak 3].
ONDERZOEKER	Ok, bedankt.

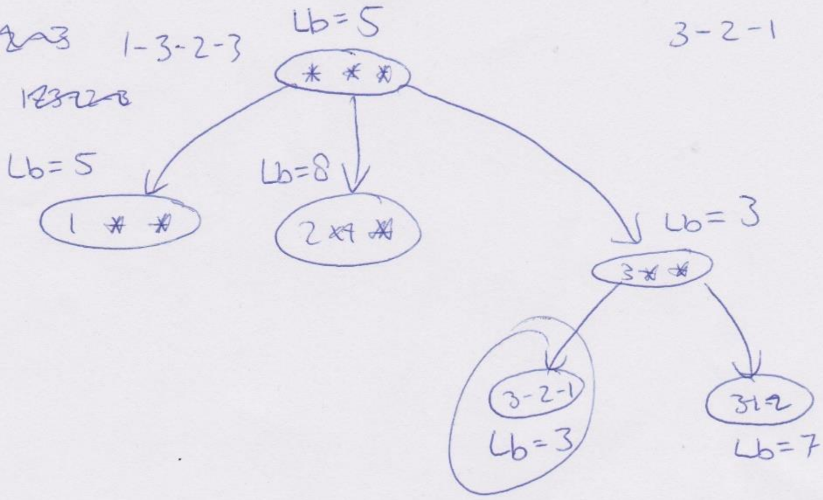
2-3-1

i	1	2	3
p_i	5	6	5 , 4
r_i	1	7	2
d_i	15	11	12

3 2 1 5

1-3-2-3

1-3-2-3 3-2-1



Interview met leerling 4

ONDERZOEKER	Vond je het een leuke opdracht?
LEERLING	Delen wel. Soms iets minder, maar dat was vooral omdat het een hele lange middag was. Maar de opdracht zelf vond ik op zich wel leuk.
ONDERZOEKER	Ten opzichte van gewone wiskundelessen?
LEERLING	Ehm, ja. Ik denk dat het sowieso wel nuttig is. Ja.
ONDERZOEKER	Ok. Moest je je haasten om de opdracht af te krijgen?
LEERLING	Ja, dat best wel ja.
ONDERZOEKER	Maar je hebt hem wel afgrekregen?
LEERLING	Nou we hebben één vraag niet gemaakt want die snapten we niet. Maar voor de rest wel alles afgekregen.
ONDERZOEKER	Ok. Was dat de laatste?
LEERLING	Ja, vraag 21.
ONDERZOEKER	Wat vond je lastig aan de opdracht?
LEERLING	Ik vond vooral lastig dat er hele lappen tekst stonden en het was best wel lawaaiig in het lokaal. Dus dan was het wel moeilijk om je daarop te concentreren, dus dan snapte ik het soms ook niet. Maar als je dan een regel teruglas, dan snapte ik het wel weer.
ONDERZOEKER	Ja. Op zich waren de stukken tekst redelijk afgewisseld met vragen. Maar je denkt nog steeds dat het vrij veel tekst achter elkaar was?
LEERLING	Ja, dat denk ik wel.
ONDERZOEKER	Waren er ook delen waarvan je zegt; nou, dat ging me wel erg makkelijk af?
LEERLING	Ja, wel een paar vragen waarvan we dachten van; oh, dit is eigenlijk best wel heel makkelijk.
ONDERZOEKER	Maar was dat welkom, die afwisseling dat het makkelijk was? Of
LEERLING	Ja zeker [*lacht].
ONDERZOEKER	Ok, dat was al het algemene deel. Dan nu over het onderwerp zelf. Welk probleem waren we aan het oplossen in de eerste twee delen van de opdracht?
LEERLING	Oh jeetje. Ehm, nou het probleem dat een computer heel veel moet uitzoeken en dat dat dan op een makkelijke manier kan. Ongeveer.
ONDERZOEKER	Ja, je gaat eigenlijk al één stap verder. Voordat we gaan kijken hoe we het op gaan lossen, hebben we al een probleem. Kun je dat proberen te omschrijven?
LEERLING	Het is dat er veel, wacht eens even kijken. Ja, dat je heel veel moet uitrekenen of?
ONDERZOEKER	Nou, eigenlijk het probleem is al dat je die vier taken hebt, of meer. Dus dat is een probleem, we weten niet wat een goede planning is.
LEERLING	Oh ja, ja.
ONDERZOEKER	En uiteindelijk wilden we dus een goede planning vinden voor die taken. Maar wat vonden we een goede planning? Wanneer vonden we een planning goed?
LEERLING	Eh, wanneer de L-max het kleinst was. Dus wanneer het het snelst was afgerond eigenlijk.
ONDERZOEKER	Ja, kun je dat nog iets verder specificeren?
LEERLING	Ehm, nou, als er dan een taak was die voor een bepaalde tijd afgerond moest worden, dat de tijd dat die erover heen ging dan zo klein mogelijk was.
ONDERZOEKER	Ja, heel mooi. Branch and Bound, waarmee we het uiteindelijk hebben opgelost, is best wel omslachtig. Dat heb je gezien, met die grote tekening enzo.

LEERLING	Ja.
ONDERZOEKER	Waarom gebruikten we dat dan eigenlijk?
LEERLING	Om het duidelijk te maken denk ik. Dat je er een beeld bij had.
ONDERZOEKER	Duidelijk voor wie?
LEERLING	Voor jezelf wel.
ONDERZOEKER	Ok. Maar in de praktijk gebruiken ze het ook heel vaak. Er zijn heel veel plekken, fabrieken enzo, die dat echt gebruiken om zulke problemen op te lossen [*wijst naar planningsproblemen in openliggend opdrachtenboekje]. Waarom passen zij dat dan toe?
LEERLING	Ehm, ja [*pakt opdrachtenboekje en begint erin te zoeken]. Dit bedoel je? [*wijst naar Branch and Bound boom]
ONDERZOEKER	Ja, dat bedoel ik. Branch and Bound is inderdaad die stappen nemen. Waarom zou dat vaak in de praktijk worden toegepast, terwijl het eigenlijk zo omslachtig lijkt?
LEERLING	Dat weet ik niet zo goed eigenlijk.
ONDERZOEKER	Ok, dat maakt niet uit. Wat is de meest voor de hand liggende methode om zo’n probleem, zo’n planningsprobleem, op te lossen? Als je dat hebt [*wijst naar planningsprobleem in opdrachtenboekje] en je wilt zo weinig mogelijk logisch nadenken. Hoe zou je het dan op kunnen lossen?
LEERLING	Eh, bedoel je die beste planning?
ONDERZOEKER	Ja.
LEERLING	Eh, nou ja, door te kijken naar die, eh, wanneer die, dat was vervalmoment volgens mij, welke dan het snelste afloopt, die moet je dan als eerste doen.
ONDERZOEKER	Kun je daar ook een naam aan geven? Daar is een term voor.
LEERLING	Ja, preempting toch?
ONDERZOEKER	Ja nou, heel goed inderdaad. Dat preempten we dan. Waar ik naar op zoek was eigenlijk, was het woord heuristiek in dit geval.
LEERLING	Oh ja.
ONDERZOEKER	Kun je van heuristieken in zijn algemeenheid een voor- en een nadeel van noemen?
LEERLING	Eens kijken wat dat ook alweer was [*pakt opdrachtenboekje erbij].
ONDERZOEKER	Probeer het zonder kijken [*neemt opdrachtenboekje weg]. Ik wil even kijken wat is blijven hangen.
LEERLING	Oh ja. Wat was de vraag ook alweer?
ONDERZOEKER	Heuristieken zijn kort aan bod gekomen. Kun je een voor- en een nadeel noemen van heuristieken?
LEERLING	Ehm [stilte] dus dat was toch dat het de beste oplossing was? Heuristiek, ik weet niet meer precies wat dat was eigenlijk.
ONDERZOEKER	Nee, dat is niet erg. Een computer zou ook dit probleem [*wijst naar planningsprobleem in opdrachtenboekje] op kunnen lossen door gewoon alle mogelijke planningen te proberen. Dus 1-2-3-4, 1-2-4-3, en dan alles proberen en dan de beste kiezen.
LEERLING	Ja.
ONDERZOEKER	Maar er kleeft een nadeel aan die methode.
LEERLING	Ja, dat kost heel veel tijd en heel veel werk.
ONDERZOEKER	Ja, inderdaad. Maar als we dan wel de optimale oplossing willen vinden, maar dus niet zoveel tijd eraan willen besteden, dan kunnen we Branch and Bound toepassen.
LEERLING	Ja.

ONDERZOEKER	We hebben hier Branch and Bound toegepast op een planningsprobleem, maar je kunt Branch and Bound toepassen op heel veel verschillende problemen. Alleen dan moet je dat probleem in een bepaalde vorm gieten.
LEERLING	Ja.
ONDERZOEKER	Kun jij bedenken welke twee aspecten je dan nodig hebt om Branch and Bound toe te passen op een probleem?
LEERLING	Oh ja dat stond in dat boekje. Ehm, nee dat weet ik niet meer.
ONDERZOEKER	Ok dan gaan we door naar het derde deel van het interview. Daarvoor heb ik dit uitgeprint [*pakt planningsprobleem in tabelweergave erbij]. Hier staat zo'n tabel met een paar taken. En ik wil jou vragen om hier Branch and Bound op toe te passen en dus een optimale oplossing te vinden.
LEERLING	Ok.
ONDERZOEKER	En terwijl je dat doet, probeer dan hardop te zeggen wat er in je omgaat.
LEERLING	Ok. Maar je mag ook preempting toepassen?
ONDERZOEKER	Ja, doe wat jij vindt dat nodig is om een optimale oplossing te bepalen?
LEERLING	Ik begin met nummer 1, die begint bij 1 [*noteert 1, -> taak 1] en die gaat tot 2 [*noteert 2 achter de pijl]. Dus dan heeft ie er 4 over [*noteert (4 over) onder taak 1]. Dan begint taak 3, die duurt tot 7, dus die heeft er 0 over. Dan begint taak 2, die duurt dan tot 13. Die heeft er dan 0 over maar de L is dan 2. En dan is er nog 4 over van taak 1, dus tot 17 en dan is L ook 2. Dus dan is L-max 2. Klopt het? [*legt pen neer]
ONDERZOEKER	Ja, nu heb je iets bepaald. Wat heb je eigenlijk, als je Branch and Bound gaat toepassen, nu bepaald? Je hebt die L-max voor een bepaalde planning met pre-emption bepaald, wat kun je daar nu mee?
LEERLING	Dat je weet dat met deze volgorde van de taken dat het dan het snelst gaat, of dat het dan het best is.
ONDERZOEKER	Ja, maar hier hebben we pre-emption toegepast. En uiteindelijk willen we een optimale oplossing vinden zonder pre-emption. Maar we gebruiken dit wel om tot die oplossing te komen met behulp van Branch and Bound.
LEERLING	Oh ok.
ONDERZOEKER	Doe gewoon net alsof je een computer bent, die niet logisch na kan denken, die dit probleem aan wil pakken.
LEERLING	Ok, dus bedoel je dan dat je weer met die sterretjes begint [*teken ovaal *-*-*]?
ONDERZOEKER	Ja, en nu verdergaat wanneer je weet welke kant je op moet.
LEERLING	Ok, dus dan, even tekenen [*tekent ovalen 1-*-*, 2-*-* en 3-*-*]. Ehm [stilte] dus deze is dan, LB was dat hè, dus die is LB 2 [*noteert LB=2 bij *-*-*].
ONDERZOEKER	Waar staat dat voor, LB?
LEERLING	Eh, lower bound volgens mij. Dus dan reken je het voor deze allemaal uit [*gebaart naar ovalen 1-*-*, 2-*-* en 3-*-*]. Dus 1-*-*, dan begin je dus met taak 1 en die moet je helemaal laten aflopen, dus tot 6. En dan begint taak 3, dan mag je wel preempten toepassen toch? [*kijkt vragend naar onderzoeker]
ONDERZOEKER	Ik wil even zien wat jij denkt.
LEERLING	Oh, ok. Nou, die laat je doorlopen tot 7 dus dan blijft er 4 over. Dan komt taak 2, tot 13. Dan heb je weer L=2. En dan komt nog die 4 van taak 3 en die gaat dan tot 17 dus L=5. Dus bij 1-*-* is LB=5.
ONDERZOEKER	Als je zo een extra blaadje nodig hebt, dan moet je het maar zeggen hoor. Niet dat je moet prutsen in de kantlijn.

LEERLING	Nee, ik denk dat deze nog wel passen. Nou 2-*-*. Dan begin je bij de 7 en je gaat door tot 13. Dus dat is taak 2. En dan komt taak 3, tot 18. Dan is L 6. En dan komt taak 1, tot 23. Dus dan is L 8. Dus bij 2-*-* is LB 8. Dan bij 3, dan begin je bij 2, tot 7. En dan begint taak 2, tot 13. Dan is L 2. En dan gaat taak 1 tot 18 en dan is L 3. Klopt dat? [stilte] Ja. Dus dan ga je verder met 3-*-*. Mag ik een nieuw blaadje?
ONDERZOEKER	Ja [*geeft nieuw blaadje]
LEERLING	Dus dan kan je 3-1-2 nog hebben en 3-2-1. Dus bij 3-1-2 begin je bij 2, das taak 3, tot 7. Dan gaat taak 1, tot 12. En dan taak 2, tot 18. Dus dan is L 7. En bij 3-2-1 begin je ook bij 2, das taak 3, tot 7. En dan komt taak 2, tot 13. Dan is L 2. En dan komt taak 1, tot 18 en dan is L 3. Dus dan is 3-2-1 de beste oplossing [*omcirkelt 3-2-1].
ONDERZOEKER	Ja, mooi.

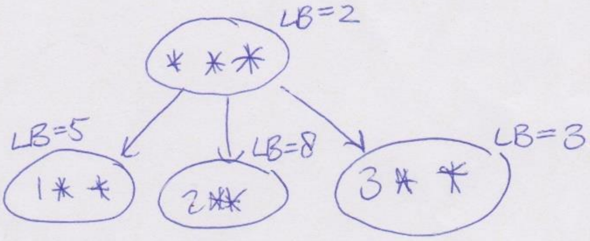
i	1	2	3
p_i	5	6	5
r_i	1	7	2
d_i	15	11	12

$1 \xrightarrow[\text{taak 1}]{} 2 \xrightarrow[\text{taak 3}]{} 7 \xrightarrow[\text{taak 2}]{} 13 \xrightarrow[\text{taak 1}]{} 17$

(4 over) (0 over) (0 over)

L = 2 L = 2

Lmax = 2



1 * *

↳ $1 \xrightarrow[\text{taak 1}]{} 6 \xrightarrow[\text{taak 3}]{} 7 \xrightarrow[\text{taak 2}]{} 13 \xrightarrow[\text{taak 3}]{} 17$

(4 over) L = 2 L = 5

2 * *

↳ $7 \xrightarrow[\text{taak 2}]{} 13 \xrightarrow[\text{taak 3}]{} 18 \xrightarrow[\text{taak 1}]{} 23$

L = 6 L = 8

3 * *

↳ $2 \xrightarrow[\text{taak 3}]{} 7 \xrightarrow[\text{taak 2}]{} 13 \xrightarrow[\text{taak 1}]{} 18$

L = 2 L = 3

312

↳ $2 \xrightarrow[\text{taak 3}]{} 7 \xrightarrow[\text{taak 1}]{} 12 \xrightarrow[\text{taak 2}]{} 18$

L = 7

321

↳ $2 \xrightarrow[\text{taak 3}]{} 7 \xrightarrow[\text{taak 2}]{} 13 \xrightarrow[\text{taak 1}]{} 18$

L = 2 L = 3

Interview met leerling 5

ONDERZOEKER	Vond je het een leuke opdracht?
LEERLING	Nou, ik vond het wel leuk om te doen ja. Ja, wat kan ik erover zeggen. Ja, het was zeg maar anders. Niet gewoon saai dingen uitrekenen, maar het was echt, ja je kon er echt mee bezig. Het hield je ook bezig. Zeker dat je bijvoorbeeld dan de beste oplossing moest gaan zoeken, dan moest je wel even zelf denken van nou, wat nu. Ja, het was leuk om te doen. Het was ook een nieuw onderwerp.
ONDERZOEKER	Moest je je haasten om de opdracht af te krijgen?
LEERLING	Nee. Nee, we waren een half uur vantevoren klaar met alle opdrachten, dus ja.
ONDERZOEKER	Denk je dat dat er mee te maken had dat jullie met z'n drieën waren? Denk je dat het daardoor sneller is gegaan?
LEERLING	Ehm, ja ik denk dat het wel wat invloed heeft gehad maar ik denk dat [] de grootste invloed heeft want dat is zeg maar een wiskunde-meester. Die had alles heel gauw door.
ONDERZOEKER	Maar jullie begrepen het alle drie ook?
LEERLING	Ja, tenminste, ik begreep het wel. Ik weet niet of [] het begreep.
ONDERZOEKER	Wat vond je lastig aan de opdracht?
LEERLING	Ehm, tja. Echt iets lastigs zat er niet tussen. Ik kreeg op een gegeven moment kramp in mijn vingers van het vele schrijven maar verder…
ONDERZOEKER	Was er een deel waarvan je zegt; nou, dat ging me wel erg makkelijk af?
LEERLING	Ja, op een gegeven moment kreeg je een paar opdrachten over dat je telkens een ballonnenschema moest tekenen en dat vond ik wel redelijk herhalend dus dan kon je je hersenen gewoon uitzetten en doe je het gewoon een paar keer achter elkaar en dan is het klaar.
ONDERZOEKER	Ok, dat waren al de algemene vragen. Dan nu over het onderwerp zelf. Welk probleem, in de eerste twee delen [*wijst naar opdrachtenboekje], wilden we eigenlijk oplossen?
LEERLING	Je wilde een zo goed mogelijke oplossing vinden voor een machine [onverstaanbaar] de tijd zo efficiënt mogelijk gebruikte. En dan wilde je graag daar een programmeerding voor bedenken zodat je dat makkelijk uit kon rekenen.
ONDERZOEKER	En, die planning dus. Je zei al; we wilden een efficiënte planning maken. Wanneer is een planning, volgens de opdracht die we hebben gemaakt, wanneer is een planning goed?
LEERLING	Ehm, als de, eh, de. Nou je hebt zeg maar die deadline van wanneer het af moet zijn. Als dan de tijd tussen die deadline en wanneer het af is, maar het moet dan wel later zijn, als die zo klein mogelijk is, dan heb je een optimale planning.
ONDERZOEKER	Dus stel, ik heb twee taken, die allebei op moment 4 klaar. Zeg maar die een vertraging hebben van 4, sorry. Of ik heb twee taken waarvan één een vertraging heeft van 1 en één een vertraging van 5. Welke planning vinden we dan beter?
LEERLING	Ja, dat je eerst die enen mag beginnen. Of ja, wat bedoel je eigenlijk met een vertraging?
ONDERZOEKER	Ik bedoel de vertraging. Een taak heeft een vertraging van 4 als die 4 uur na zijn vervalmoment klaar is.
LEERLING	Oh ja.

ONDERZOEKER	Dus hebben we dan liever dat beide taken 4 momenten, 4 uren, na hun vervalmoment klaar zijn, of dat één taak 1 uur en de andere taak 5 uur na zijn vervalmoment klaar is?
LEERLING	Beiden 4.
ONDERZOEKER	Ja. Daar waren we tenminste in deze opdracht mee bezig. Het ligt natuurlijk aan wat je ermee wilt in de praktijk.
LEERLING	Ja.
ONDERZOEKER	Ok. Op zich dat Branch and Bound, dat is best wel een omslachtige methode. Dus hoezo pasten we dat eigenlijk toe, als dat zo omslachtig is?
LEERLING	Omdat je toch wel op een redelijk goed antwoord uitkwam en je hoefde niet alles op te schrijven. Dus heel omslachtig vond ik het eigenlijk ook weer niet. Maar het was in ieder geval, dan hoef je niet alles stuk voor stuk uit te rekenen, wat heel veel tijd kost.
ONDERZOEKER	Ja, precies. Want die andere methode ligt wat meer voor de hand, dan ga je alle oplossingen na. Weet je nog hoe dat genoemd wordt? Het werd alleen maar even kort genoemd.
LEERLING	Eh, nee.
ONDERZOEKER	Ok. Het begrip heuristiek is ook kort aan bod gekomen. Kun je je dat nog herinneren en kun je daar misschien een voor- en een nadeel van noemen, van heuristieken?
LEERLING	Nou ja, een heuristiek was dat je dan een soortement van logica had voor wanneer je die taak inplande. Als voorbeeld hadden ze dan dus dat je op volgorde van eindtijd deed, dus die het eerste af moest zijn daar begon je mee. En die die het laatst af moest zijn, die deed je als laatste. Ehm, nou als nadeel heeft dat dat het, ja, als je dan, dat was toch die opdracht. Als je dan eentje die vroeg klaar is maar heel laat kan beginnen en de rest is allemaal later klaar, dan moet je eerst wachten op die eerste die vroeg klaar moet zijn. En dan heb je dus eigenlijk de eerste tijd al verspild, dus dat is niet helemaal efficiënt.
ONDERZOEKER	En een voordeel van zo'n heuristiek toepassen?
LEERLING	Nou ja, je hebt iets zeg maar om aan vast te houden. Dus het geeft wel een redelijke richting aan. Als het niet te gek afwijkt dan kom je ook wel op een redelijk goed antwoord uit. En je hoeft dus niet alles uit te rekenen want als je gewoon denkt van nou, ik doe het op deze volgorde, dan kun je het zo snel bepalen.
ONDERZOEKER	Ja. Hier hebben we vooral Branch and Bound toegepast op een planningsprobleem [*wijst naar opdrachtenboekje]. Maar je kunt dat toepassen op veel verschillende soorten problemen. Om dat te doen, heb je eigenlijk twee aspecten nodig. Je moet problemen in zo'n vorm gieten dat je Branch and Bound kunt toepassen. Heb je een idee van waar ik naar toe wil met die twee aspecten?
LEERLING	Ja, eentje is dat je dan als het ware zo'n schema maakt van hoeveel tijd het kost, wanneer het mag beginnen, als dat zo is, en je moet een manier hebben om de lower bound te berekenen.
ONDERZOEKER	Een lower bound inderdaad, die heb je nodig. En wat zei je als eerste?
LEERLING	Dat je een overzicht hebt van hoeveel tijd iets kost en dan eventueel ook of iets op een bepaalde tijd pas mag beginnen en wanneer het af moet zijn.
ONDERZOEKER	Ok, dat was al het tweede deel. Tot slot wil ik wil je vragen om een planningsprobleem op te lossen met behulp van Branch and Bound. Dus net

	alsof je een computer bent. Met een beetje logisch nadenken kun je misschien de oplossing heel snel bepalen maar gebruik Branch and Bound alsof je een voorgeprogrammeerde computer bent en spreek daarbij hardop de stappen uit die je zet.
LEERLING	Ok.
ONDERZOEKER	Eventueel heb ik er nog één als kladblaadje [*overhandigt extra papier].
LEERLING	Moet ik ook helemaal die boom uitschrijven, die met die ballonnen? Of niet?
ONDERZOEKER	Ja, dat is wel nodig, wan…
LEERLING	Ja dat is wel Branch and Bound ja.
ONDERZOEKER	Jij kunt wel beredeneren wat de oplossing is, maar als je dit straks met 20 taken doet, dan moet je het door een computer laten oplossen en die kan niet op dezelfde manier redeneren als jij. Dus die boomstructuur is dan wel nodig om weer te geven wat je doet.
LEERLING	Ja, en je mag dus niet dat pre-emption, dus niet dat je dan mag stoppen en dan met de andere mag beginnen en die ene later afmaken?
ONDERZOEKER	Wellicht, probeer eens.
LEERLING	Nou, begin eens bovenaan met sterretje, sterretje, sterretje [*tekent ovaal *-*-*]. En we kunnen taak 1 als eerste doen, taak 2 en taak 3, die kunnen allemaal als eerste [*tekent ovalen 1-*-*-*, 2-*-*-* en 3-*-*-*]. Nou nu gaan we eens kijken, want die sterretjes die moet je dus nog invullen en dan mag je dus wel dat pre-emption doen. Oh, overal twee sterretjes [*schrapt laatste sterretjes in zojuist getekende ovalen]. Nou ja, dus dan gaan we nu kijken wat voor 1, als je 1 als eerste doet, wat dan het snelste is. Oh nee wacht, we moeten eerst even eigenlijk helemaal bovenaan doen, dus als je alles met pre-emption doet. Dan beginnen we met 1 [gemompel] dus dan is die af en dan maak je die af, zit je op 13 dus 2 overheen en dan moet je die nog afmaken, zit je er 3 boven, toch. Nee want dan zit je op 17 dus zit er er 2 boven, toch. Wacht [gemompel] [*noteert 1 – 1x1 – 5x3 – 6x2 – 4x1] lower bound 2 [*noteert LB=2 bij ovaal *-*-*]. Nou dan doen we 1 als eerste, dus dan zit je op 1 keer niks [*noteert 1 –] en dan heb je 5 keer 1 en dan zit je inmiddels op 6 dus dan is het handigste om nog 1 keer 3 te doen, vervolgens 6 keer 2 en dan 4 keer 3. [gemompel] dan zit je op 17, dan zit je er 5 overheen [*noteert LB=5 bij ovaal 1-*-*]. Dan 2 als eerste. Dan moet je eerst 7 wachten, dan zit je op, eh, 7 keer niks, 6 keer 2, dan zit je op 13, zit je er 2 overheen. Ehm, dan doe je 5 keer 3 en dan nog 5 keer 1. Bij die zit je er 2 overheen. Plus 5 is 17, plus 5 is 22, dan zit je er 7 overheen [*noteert LB=7 bij ovaal 2-*-*]. Nou 3 als eerste. Dan doe je 2 keer niks, dan 5 keer 3. Ehm, dan zit je inmiddels op 7. Dan ga je 2 doen en dan als laatste nog 1. Dan heb je 3 voor de tijd af, zit je op 7, plus 6 is 13, dus dan zit je er weer 2 boven. Plus 5 is 18, daar zit je 3 over [*noteert LB=3 bij ovaal 3-*-*]. 3-*-* heeft nu de laagste lower bound dus dan ga je verder. Dan heb je dus combinatie 3-1-2 en 3-2-1 [*tekent ovalen 3-1-2 en 3-2-1]. Ga je daar de lower bounds van bepalen [*schrijft LB onder ovaal 3-1-2]. Of nee, dat is gewoon de oplossingswaarde [*schrapt LB en schrijft in plaats daarvan Lmax op]. 3-1-2 dus dan heb je eerst 3 helemaal. Dus eerst 2 keer niks, dan 5 keer 3. 3-1-2 dus kan daarna 1, dan zit je al over de 7 dus kan er nog 6 keer 2. Ehm, dan heb je 3 voor die tijd af, 1 heb je voor die tijd af, zit je dus op 12, plus 6 is 18, zit je 7 over 2. Dus dan is oplossingswaarde 7 [*noteert 7 achter Lmax onder ovaal 3-1-2]. Ehm, 3-2-1. Dat is eerst 3, doe je 2 keer niks, doe je 5 keer 3, dan zit je op 7 dus dan kun je 2 doen. 6 keer 2 en dan nummer 1, das 5 keer 1. Dan zit je 2 over 2 heen en zit je 3 over 1 heen [*noteer Lmax 3 onder ovaal 3-2-1]. Nou dan heb je dus het antwoord gevonden. Dus 3-2-1 is dan

	de efficiëntste oplossing omdat dan dus, nou ja, je begint met die lower bound van 2, als ik die goed heb uitgerekend, en dan kijk je dus van wat heeft de laagste lower bound en reken je daar mee verder. En nou ja, de L-max van 3-2-1 die is 3 en dat is het allerlaagste wat er is dus dan is dat de beste oplossing.
ONDERZOEKER	Ja, mooi. Bedankt.

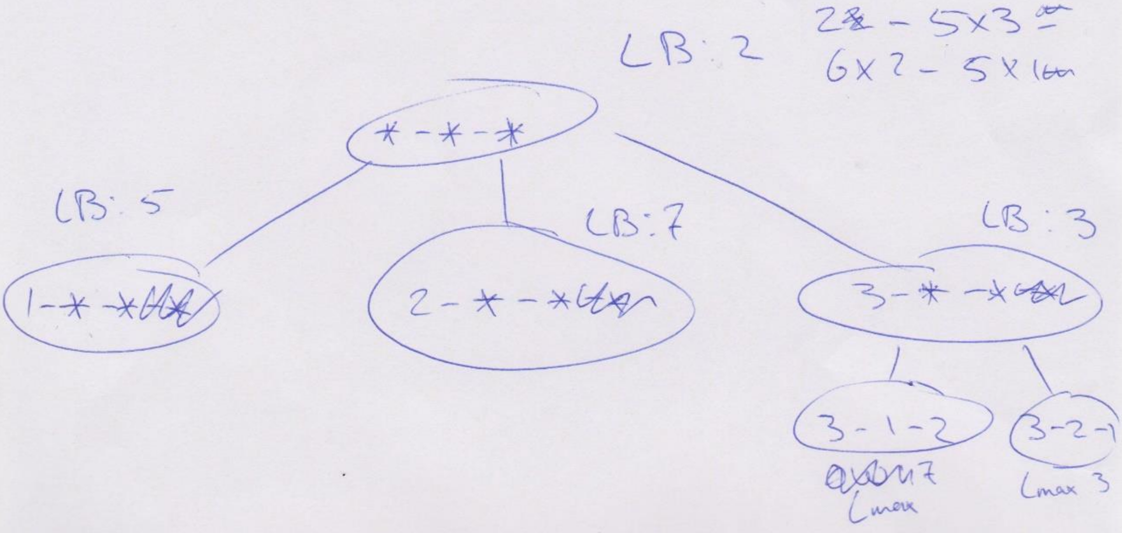
i	1	2	3
p_i	5 4	6	5
r_i	1	7	2
d_i	15	11	12

1 – 1x1 – 5x3
– 6x2 – 4x1

1 – 5x1 – 1x3
– 6x2 – 4x3

7 – 6x2 – 5x3 –
5x1

2 – 5x3 –
6x2 – 5x1



2 – 5x3 – 5x1 – 6x2
2 – 5x3 – 6x2 – 5x1

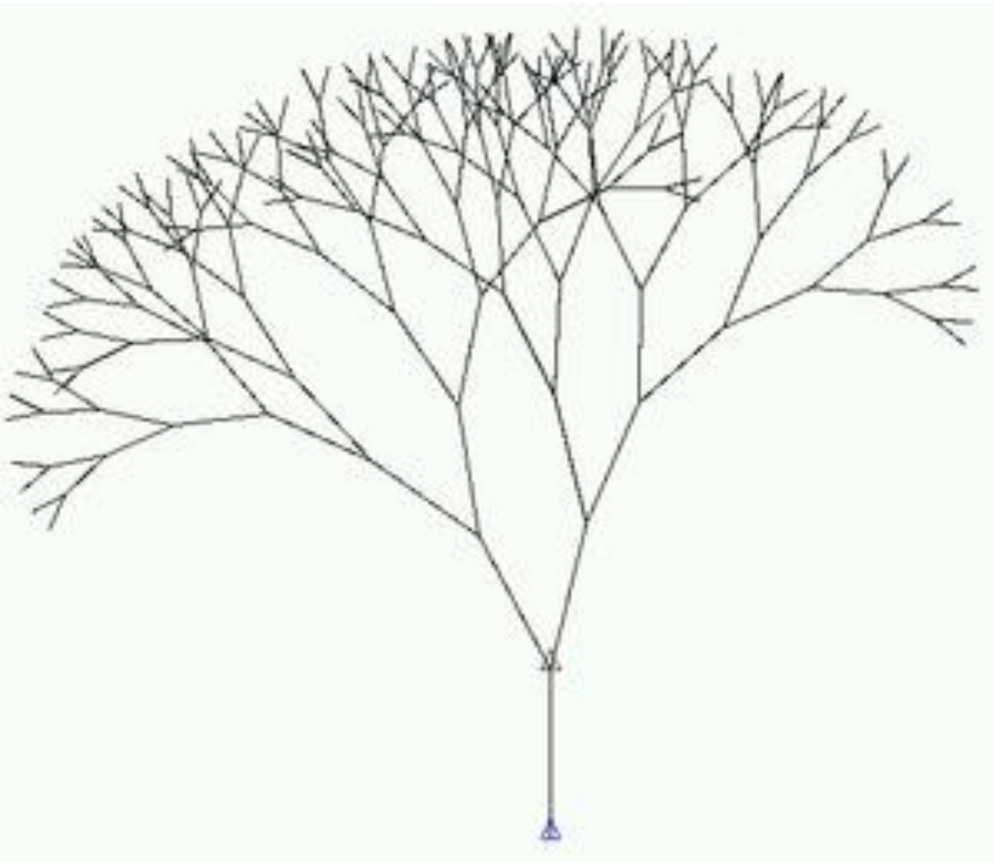
Bijlage E: Praktische opdracht - versie 2

De komende vijftien pagina’s geven de verbeterde versie van de praktische opdracht weer. De aanbevelingen uit Hoofdstuk 6 zijn hierin verwerkt. Het correctievoorschrift behorende bij deze versie van de praktische opdracht is te vinden in Bijlage F.

Praktische Opdracht

5 VWO Wiskunde B

Branch & Bound



28 maart 2017

Inleiding

Welkom op deze werkmiddag! Je gaat vandaag van [tijdsbestek van 4 uur] bezig met *Branch and Bound*, een onderwerp op het raakvlak van wiskunde en informatica. Wat Branch and Bound precies inhoudt, wordt verderop in deze opdracht duidelijk.

De opdracht maken jullie in groepjes van twee. Gedurende de middag is overleg met andere groepjes niet toegestaan. Als iets onduidelijk is of jullie ergens echt niet uitkomen, mag je uiteraard wel de docent om extra uitleg vragen. De docent heeft een lijstje met vragen waarbij hij jullie niet zal helpen. Deze vragen kenmerken zich in de opdracht doordat er een sterretje voor staat. Mochten jullie uit zo'n niet komen, sla deze dan over en kijk er eventueel later nog eens naar.

Laptops zijn gedurende de middag niet toegestaan en ditzelfde geldt uiteraard voor mobieltjes. Jullie eindproduct is dan ook een handgeschreven verslag waarin de opgaven uit dit boekje op volgorde zijn uitgewerkt. Het verslag leveren jullie vandaag om uiterlijk [eindtijd] in bij de docent.

De beoordeling vindt in principe per tweetal plaats, tenzij een individuele beoordeling meer op zijn plaats lijkt. Er wordt beoordeeld op (wiskundig) niveau van de aanpak en uitwerkingen van de opgaven. Het is niet nodig om een voorblad, inhoudsopgave of inleiding te maken. Wel is het nodig iedere opgave te voorzien van een uitwerking of toelichting. Antwoorden zonder de vereiste berekeningen of verklaringen leveren geen of weinig punten op!

We raden jullie aan om de opdrachten <u>niet</u> op te delen, dat gaat niet lukken! Samen erdoorheen en telkens overleggen en samen nadenken zal tot de beste antwoorden leiden.

Veel succes met deze opdracht!

Deel 1: planningsproblemen

In een fabriek staat een machine die verschillende taken moet verrichten: taak 1, taak 2, etc. Elk van deze taken heeft een bepaalde duur. Deze duur geven we aan met $p_1, p_2, \ldots$, waarbij het getal het nummer van de taak is. Naast een bepaalde duur kent iedere taak ook een lanceer- en een vervalmoment. Het *lanceermoment* van een taak geeft aan vanaf wanneer de machine met de desbetreffende taak kan beginnen en het *vervalmoment* van een taak geeft aan wanneer we willen dat de machine met de desbetreffende taak klaar is. Het lanceermoment geven we aan met $r_1, r_2, \ldots$ en het vervalmoment met $d_1, d_2, \ldots$.

NB. *p, r* en *d* zijn ontleend aan de Engelse woorden *processing time*, *release date* en *due date.*

Met de notatie die we nu voorhanden hebben, kunnen we overzichtelijk de taken weergeven die een machine moet uitvoeren. Neem bijvoorbeeld de twee taken in Tabel 1, waarin de duur, het lanceermoment en het vervalmoment van twee taken weergegeven worden. De 3 en de 4 in Tabel 1 zijn de waarden van p_1 en p_2. Uit de tabel kunnen we dus aflezen dat de duur van taak 1 gelijk is aan 3 tijdseenheden en de duur van taak 2 aan 4 tijdseenheden. Op dezelfde manier kunnen we aflezen wat de lanceer- en vervalmomenten van de twee taken zijn. De lanceermomenten van de beide taken zijn $r_1 = 5$ en $r_2 = 1$ en de vervalmomenten zijn $d_1 = 9$ en $d_2 = 6$. Probeer goed te begrijpen hoe deze tabel werkt; er zullen gedurende deze opdracht meerdere soortgelijke tabellen gepresenteerd worden.

Tabel 1: twee taken

taak	1	2
p	3	4
r	5	1
d	9	6

Wat wij in deze opdracht gaan doen, is het systematisch proberen in te plannen van deze taken. Zo systematisch, dat we het met een klein beetje programmeerkennis gemakkelijk door een computer zouden kunnen laten doen, ook als we veel meer dan twee taken moeten inplannen. Met inplannen bedoelen we simpelweg een volgorde kiezen waarop de machine de taken aanpakt. Bij het inplannen van taken moet met twee dingen rekening gehouden worden:

- De machine kan aan slechts één taak tegelijk werken
- De machine kan pas aan een taak beginnen zodra er het lanceermoment van deze taak bereikt is

Merk op dat het zo kan zijn dat een machine soms een tijdje niks aan het doen is.

Kijk nog eens naar Tabel 1, waarin twee taken weergegeven worden die door een machine uitgevoerd moeten worden. Uit Tabel 1 kunnen wij de lanceermomenten van de beide taken aflezen: $r_1 = 5$ en $r_2 = 1$. Dit betekent dat de eerste taak pas mag beginnen zodra 5 tijdseenheden zijn verstreken en de tweede taak zodra er één tijdseenheid verstreken is. De duur van de tweede taak is vier tijdseenheden (immers, $p_2 = 4$), waardoor we kunnen concluderen dat taak 2 afgerond kan worden voordat taak 1 gestart mag worden. Het zou dus logisch zijn deze als eerste in te plannen. Beide taken hebben ook een vervalmoment, waarmee het moment aangegeven wordt waarop we willen dat de desbetreffende taak afgerond is. Dit vervalmoment wordt niet altijd gehaald, soms kan dat niet anders. Dan geldt over het algemeen: hoe minder te laat, hoe beter. Indien de machine eerst taak 1 uit Tabel 1 uitvoert en vervolgens pas taak 2, is taak 1 na 5 + 3 = 8 tijdseenheden afgerond en taak 2 na 8 + 4 = 12 tijdseenheden. Dit is 6 tijdseenheden later dan het vervalmoment van taak 2.

Vraag 1) Na hoeveel tijdseenheden is de machine klaar indien eerst taak 2 wordt uitgevoerd en vervolgens taak 1? Houd er rekening mee dat een taak niet gestart mag worden voor zijn lanceermoment.

In Tabel 2 worden vier taken weergegeven die door een machine uitgevoerd moeten worden.

Tabel 2: vier taken

taak	1	2	3	4
p	4	2	6	5
r	0	1	3	5
d	8	12	11	10

Vraag 2) Uit Tabel 2 volgt dat de machine nooit alle taken voor hun vervalmomenten uit kan voeren. Leg dit uit.

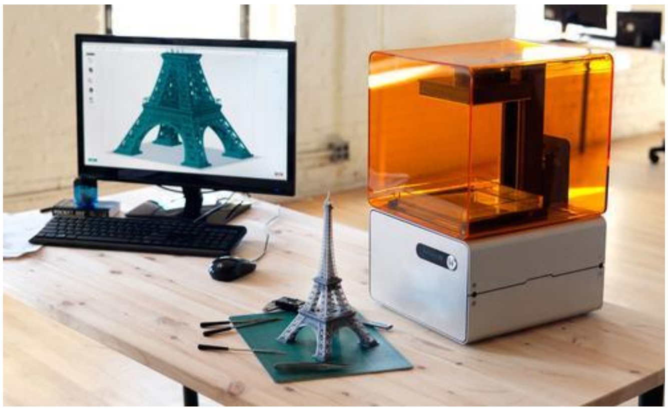
Een taak die niet uiterlijk op zijn vervalmoment uitgevoerd kan worden, is te laat. De *vertraging* van een taak die te laat is, is gelijk aan het aantal tijdseenheden tussen de afronding en het vervalmoment van een taak. Taken die uiterlijk op hun vervalmoment uitgevoerd kunnen worden, zijn op tijd en lopen geen vertraging op.

Vraag 3) Als we de taken uit Tabel 2 inplannen in numerieke volgorde, dus in de volgorde 1-2-3-4, wat is dan de vertraging van ieder van de taken?

We weten nu dat de machine nooit alle taken uit Tabel 2 voor hun vervalmomenten uit kan voeren. Alle taken moeten echter wel vervuld worden. Om goede en slechte planningen van elkaar te kunnen onderscheiden, is het noodzakelijk een criterium op te stellen aan de hand waarvan bepaald kan worden hoe goed een bepaalde planning is. Wat een goed criterium is, hangt af van de situatie. Neem bijvoorbeeld de volgende twee situaties waarin 3D-modellen worden geprint in opdracht van klanten.

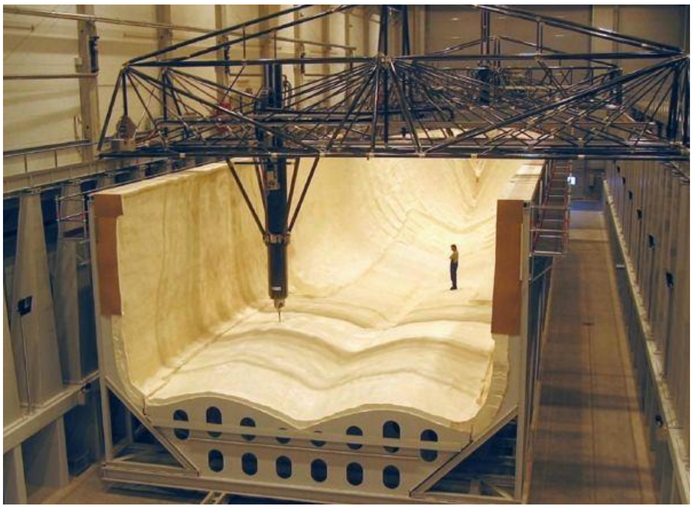
Situatie 1: Sarah is een eigen bedrijfje gestart. Ze heeft een 3D-printer aangeschaft en print modellen op verzoek van klanten. Zij belooft haar klanten binnen een bepaalde tijd te leveren. Deze tijd wordt direct na binnenkomst van een bestelling vastgesteld op basis van de grootte en complexiteit van het model. Lukt het Sarah niet om binnen de tijd een model aan te leveren, dan krijgt de klant 25 euro korting per model dat te laat is. Sarah heeft niet alle soorten en kleuren filament (de “inkt” die een 3D-printer gebruikt) op voorraad, waardoor het soms een paar dagen na binnenkomst van een bestelling duurt voor zij kan beginnen met het printen van een model.

Figuur 1: De machine van Sarah: een 3D-printer voor thuisgebruik



Situatie 2: Dirks, Draaisma en Van Dam is een bedrijf dat zeer grote 3D-modellen print. Klanten plaatsen bij hun een order, geven aan wanneer er met printen begonnen kan worden (tot die tijd kunnen klanten nog wijzigingen in het design aanbrengen) en wanneer zij verwachten dat het klaar is. Uit ervaring weten Dirks, Draaisma en Van Dam dat klanten hebben er over het algemeen begrip voor hebben als hun model iets later klaar is dan dat zij verwachtten. Als het veel langer duurt, bestaat er echter de kans dat een klant boos wordt en in de toekomst naar een concurrerend bedrijf op zoek gaat om iets te laten printen. Dit is slecht voor de business en voor de reputatie van Dirks, Draaisma en Van Dam.

Figuur 2: De machine van Dirks, Draaisma en Van Dam: een grote, industriële 3D-printer



Zowel Sarah als Dirks, Draaisma en Van Dam hebben te maken met planningsproblemen die omschreven kunnen worden met behulp van een tabel zoals Tabel 1 of Tabel 2. De machine is een 3D-printer. De taken die de machine uit moet voeren zijn de orders van de klanten. Ga eens na hoe in de situatieschetsen van Sarah en van Dirks, Draaisma en Van Dam de duur, het lanceermoment en het vervalmoment van een taak naar voren komen.

Vraag 4*) Op pagina 4 is genoemd dat het noodzakelijk is om een criterium op te stellen aan de hand waarvan bepaald kan worden hoe goed een bepaalde planning is en dat het per situatie verschilt wat een goede planning is. Hieronder worden drie mogelijke criteria genoemd. Welk van deze criteria is het meest geschikt voor Sarah? En welke zou je Dirks, Draaisma en Van Dam aanraden om te gebruiken?

Criterium 1: Het aantal taken dat op tijd is zo groot mogelijk maken (zoals genoemd op pagina 4, is een taak op tijd als het uiterlijk op zijn vervalmoment afgerond is).

Criterium 2: De vertraging van alle taken bij elkaar zo klein mogelijk maken (zoals genoemd op pagina 4, is de vertraging van een taak gelijk aan het aantal tijdseenheden tussen de afronding en het vervalmoment van een taak die niet op tijd is).

Criterium 3: De maximale vertraging zo klein mogelijk maken.

Wij gaan in deze opdracht gebruik maken van criterium 3. Wij vinden een planning dus goed als de maximale vertraging zo klein mogelijk is. De vertraging van een taak duiden we met $L_1, L_2, \ldots,$ waarbij het getal wederom het nummer van de taak is (de L komt van het Engelse “lateness”). Indien een taak op tijd is, geldt voor die taak $L = 0$. Als een bepaalde planning, oftewel een bepaalde volgorde voor de taken uit een tabel, gegeven is, kunnen we L uitrekenen voor alle taken. De maximale vertraging duiden we dan aan met $L_{\max}$, de *oplossingswaarde* behorende bij deze planning. Als $L_{\max}$ zo klein mogelijk is, spreken we van een *optimale planning* of een *optimale oplossing*.

Vraag 5) Overtuig jezelf ervan dat 1-3-4-2 een optimale planning is voor de taken in Tabel 2. Wat is de oplossingswaarde behorende bij deze planning?

Het is goed mogelijk om zonder computer een optimale oplossing te vinden voor het inplannen van vier taken op één machine. In de praktijk zullen echter regelmatig veel meer taken optimaal ingepland moeten worden. Jullie kunnen je waarschijnlijk wel voorstellen dat dit zonder hulp van een computer een vrijwel onmogelijke opgave is. Om de hulp van een computer in te kunnen schakelen, moeten we ‘denken’ als een computer (*algoritmisch denken*). Een mogelijkheid waarop een optimale oplossing gevonden kan worden door een computer is door deze $L_{\max}$ uit te laten rekenen voor alle mogelijke planningen en dan de planning waarvoor $L_{\max}$ het kleinst is als antwoord te laten geven. Deze methode staat bekend als *complete enumeratie* (enumeratie komt van het Engelse *enumeration,* wat zich laat vertalen met opsomming).

Vraag 6*) Voor hoeveel mogelijke planningen moet de computer L_{max} uitrekenen in ons voorbeeld als we de hierboven beschreven strategie toepassen? Laat jullie berekening zien.

Vraag 7*) Stel dat we in opdracht van een fabrikant een optimale planning willen vinden voor de 20 taken die zijn machine moet uitvoeren. Neem aan dat de computer waarover we beschikken vijftig nanoseconden (een nanoseconde is één-miljardste van een seconde) nodig heeft om L_{max} uit te rekenen voor een mogelijke planning. Hoe lang heeft de computer dan nodig om met de hierboven beschreven strategie een optimale planning voor de 20 taken te vinden? Geef jullie antwoord in een zo groot mogelijke doch logische tijdseenheid.

Uit het antwoord dat jullie hebben gegeven bij vraag 5 blijkt als het goed is dat het geen goede strategie is om de computer L_{max} uit te laten rekenen voor alle mogelijke planningen om een optimale planning te vinden. Als je genoegen neemt met een goede planning die niet per se optimaal is maar wel snel gevonden kan worden door een computer, zijn er vele alternatieve aanpakken mogelijk. Zulke atechnieken worden *heuristieken* genoemd. Door het toepassen van een heuristiek verkrijg je snel een planning die hopelijk goed is. Een voorbeeld van een heuristiek voor planningsproblemen is de volgende:

1. Beschouw de taken die nog niet uitgevoerd zijn en waarvan het lanceermoment al bereikt is, oftewel: de taken waar je nu aan zou kunnen beginnen. (Als van geen enkele taak die nog uitgevoerd moet worden het lanceermoment al bereikt is, wacht dan tot dit wel het geval is).
2. Kies uit deze beschikbare taken degene met het vroegste vervalmoment (oftewel: de taak die het eerst af moet zijn) en voer die uit. Als dit een keus uit meerdere taken oplevert (omdat meerdere taken op hetzelfde moment af moeten zijn), kies daar dan willekeurig eentje van.
3. Herhaal de bovenstaande twee stappen totdat alle taken klaar zijn.

Vraag 8*) Pas de hierboven beschreven heuristiek toe op de vier taken uit Tabel 2. Geeft de heuristiek een optimale oplossing?

Vraag 9*) Precisie is in de wiskunde en de informatica erg belangrijk. De hierboven beschreven heuristiek is niet geheel eenduidig. Voor bepaalde planningsproblemen zal niet duidelijk zijn wat er moet gebeuren. Een computer die het stappenplan volgt zal dan waarschijnlijk vastlopen. Verzin voor de vier taken uit Tabel 2 nieuwe lanceer- en vervalmomenten zodanig dat de hierboven beschreven heuristiek niet eenduidig voorschrijft wat er moet gebeuren.

Vraag 10*) Leg uit hoe je kunt laten zien dat een heuristiek zwak is. Verzin voor de vier taken uit Tabel 2 nieuwe lanceer- en vervalmomenten zodanig dat de zwakte van deze heuristiek ten opzichte van complete enumeratie duidelijk wordt.

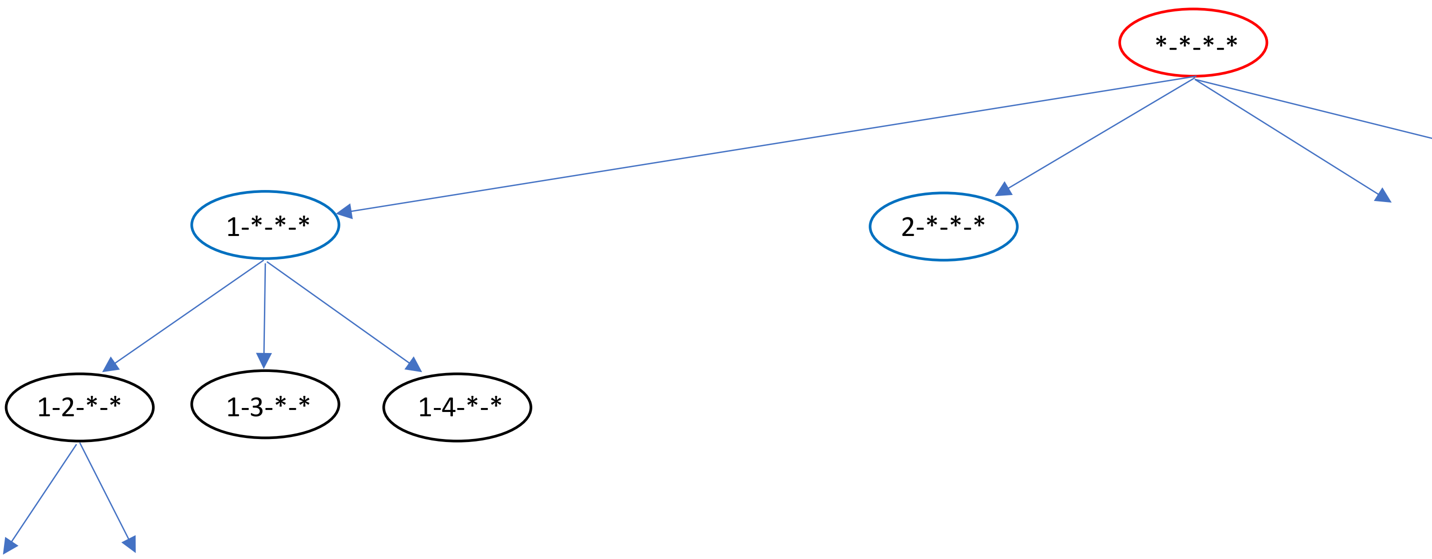
Vraag 11*) Bedenk een andere mogelijke heuristiek die toegepast kan worden op het planningsprobleem. Noteer deze stapsgewijs en pas hem toe op de vier taken uit Tabel 2.

Zoals gezegd, vindt een heuristiek niet altijd een optimale oplossing. Heuristieken worden ontwikkeld om een computer binnen afzienbare tijd met grote kans een goede oplossing te laten vinden. Er zijn echter ook manieren om met zekerheid een optimale oplossing te vinden voor een planningsprobleem zonder dat je alle mogelijkheden na hoeft te gaan. Eén zo'n manier is *Branch and Bound*, waar we in deze werkmiddag in meer detail naar gaan kijken.

Deel 2: Branch and Bound toepassen op planningsproblemen

In dit deel van de opdracht gaan jullie Branch and Bound toepassen om optimale oplossingen te vinden voor de planningsproblemen uit deel 1. Om dit te kunnen doen, moeten jullie uiteraard eerst leren hoe Branch and Bound in zijn werk gaat. Branch and Bound is een systematische manier om een optimale oplossing van een probleem te bepalen, zonder dat onnodig werk wordt gedaan. Deze systematische manier van oplossingen bepalen kan wat omslachtig lijken, maar is nuttig als je het bepalen van een optimale oplossing door een computer uit wilt laten voeren.

Vraag 12) Bepaal op een systematische manier alle mogelijke manieren waarop de taken uit Tabel 2 ingepland kunnen worden. Neem hiertoe onderstaande figuur over in jullie verslag en breidt de ovalen 1-*-*-* en 2-*-*-* uit. De kleuren van de ovalen mogen jullie voorlopig negeren.



Vraag 13*) Wat voor verband zien jullie tussen jullie antwoord op vraag 4 en de bovenstaande figuur?

Vraag 14*) Stel dat er in plaats van vier taken vijf taken ingepland moeten worden. Hoeveel ovalen krijg je dan in totaal als je een figuur tekent zoals die in vraag 9? (Hint: Het is niet nodig om deze figuur te tekenen en dit doen levert geen punten op, je kunt het antwoord beredeneren.)

Het Engelse werkwoord *to branch* is te vertalen met *zich vertakken*, zoals een boom dat doet. Nu jullie in vraag 9 alle mogelijke oplossingen op een systematische manier bepaald hebben en de resulterende figuur vergelijken met de figuur op de voorpagina van deze opdracht, zien jullie waar het Branch-gedeelte van Branch and Bound vandaan komt (zeker als je een van de twee figuren op zijn kop houdt). Het Bound-gedeelte van Branch and Bound is erop gericht om het aantal vertakking dat gemaakt moet worden in het Branch-gedeelte te verkleinen zodat maar een (klein) deel van alle mogelijke oplossingen bekeken hoeft te worden en er dus tijd bespaard kan worden zonder dat we een optimale oplossing over het hoofd kunnen zien.

Alvorens te zien hoe het Branch-gedeelte werkt in het voorbeeld van taken inplannen, maken we kennis met *pre-emption (afbreken/onderbreken)*. Als we pre-emption toestaan bij het inplannen van taken, kan een taak op elk moment gewisseld worden met een andere taak (waarvan het lanceermoment is bereikt) ook al is die eerste taak nog niet afgerond. Het werk dat al verricht was gaat niet verloren: als een taak bijvoorbeeld 10 minuten kost en na 4 minuten onderbroken wordt, dan hoeft er later als hij wordt hervat nog maar 6 minuten aan te worden besteed.

Vraag 15) Plan de taken uit Tabel 3 zodanig in dat L_{max} zo klein mogelijk is. Sta hierbij pre-emption toe. Als het goed is, vinden jullie $L_1 = L_3 = 1$ en $L_2 = 0$, dus $L_{max} = 1$. Mocht dit niet zo zijn, vraag de docent dan even om jullie te helpen.

Tabel 3: drie taken

taak	1	2	3
p	5	6	5
r	0	4	2
d	15	11	12

Vraag 16) Zie nu het planningsprobleem dat weergegeven wordt door Tabel 4. Stel dat we weten dat taak 1 als eerste ingepland en afgerond moet worden. Plan gebaseerd op dit gegeven de taken uit Tabel 4 zodanig in dat L_{max} zo klein mogelijk is. Taak 1 wordt als eerste ingepland; pre-emption is toegestaan zodra taak 1 is afgerond. Wat is nu de kleinst mogelijk waarde voor L_{max}? Laat jullie antwoord even checken door de docent.

Tabel 4: drie taken

taak	1	2	3
p	5	6	5
r	1	7	2
d_i	15	11	12

In de praktijk is pre-emption niet vaak toegestaan: taken kunnen terwijl zij uitgevoerd worden over het algemeen niet stopgezet worden en later weer gestart (zoals bijvoorbeeld bij de 3D-printers; het ene model moet afgemaakt worden voor er aan een nieuwe begonnen kan worden). Het uitrekenen van de oplossingswaarde terwijl pre-emption is toegestaan lijkt daarom niet direct nuttig voor het oplossen van het originele planningsprobleem. Het kan echter wiskundig bewezen worden dat als we pre-emption toestaan en de taken inplannen volgens de heuristiek die beschreven is in deel 1 van deze praktische opdracht, we *ondergrenzen* kunnen vinden voor een planningsprobleem (ondergrens is een vertaling van het Engelse *lower bound*, vandaar dat vaak de afkorting LB wordt gebruikt om een ondergrens te noteren). Een ondergrens voor een probleem is een waarde waar een optimale oplossingswaarde nooit onder kan liggen. Van deze eigenschap maakt Branch and Bound gebruik en het zal gauw duidelijk worden waarom het uitrekenen van de oplossingswaarde terwijl pre-emption is toegestaan toch nuttig is voor het oplossen van het originele planningsprobleem.

Vraag 17) Combineer de heuristiek uit deel 1 van deze praktische opdracht met pre-emption om een planning te verkrijgen voor het planningsprobleem dat weergegeven wordt door Tabel 2. De oplossingswaarde behorende bij deze planning is een ondergrens voor dit planningsprobleem.

Bij het uitvoeren van Branch and Bound begint een computer (of in dit geval, beginnen jullie) met het uitrekenen van een ondergrens voor het probleem. Dit hebben jullie in vraag 16 gedaan (het is handig deze ondergrens alvast te noteren naast het ovaal in jullie verslag dat overeenkomt met het rode ovaal uit vraag 11). Vervolgens bepaal je ondergrenzen voor de situaties waarin één taak al gepland is. Dit is te zien als een vertakking vanuit het rode ovaal, waarbij de blauwe ovalen uit vraag 11 gevormd worden. Er wordt bovenaan begonnen, normaal gesproken ga je voor toepassing van Branch and Bound dus niet al het hele (of halve) plaatje tekenen, zoals jullie in vraag 9 gedaan hebben.

Bij het bepalen van de ondergrenzen voor de situaties waarin één taak al gepland is (of meerdere taken al gepland zijn) moet goed op het volgende gelet worden: op de taken die al vast staan in de ovalen wordt pre-emption <u>niet</u> toegepast, maar op de taken die op de sterretjes kunnen komen <u>wel</u>. Op deze manier wordt een ondergrens gevonden voor alle mogelijke planningen die beginnen met de taken die al vast staan. Neem bijvoorbeeld het ovaal “2-*-*-*”: dit ovaal geeft aan dat taak 2 gepland is als eerste taak en dat er nog drie taken niet gepland zijn. Het lanceermoment van taak 2 is 1 en de duur van deze taak is 2. Dat betekent dus dat vanaf tijdstip 3 nog de taken 1, 3 en 4 gepland moeten worden in een nog niet vastgelegde volgorde.

Vraag 18) Hieronder wordt in een aantal stappen de werking van Branch and Bound verder toegelicht. Er ontbreken echter hier en daar iets. Ga na wat er op de puntjes zou moeten staan en vul aan:

1. Kies uit de nog niet vertakte ovalen het ovaal met …….
2. Vertak de gekozen ovaal en bereken ondergrenzen voor alle nieuwe ovalen die uit deze vertakking voortkomen. Indien ……, bereken je …… in plaats van een ondergrenzen voor alle nieuwe ovalen die uit deze vertakking voortkomen.
3. Herhaal stappen 1 en 2 totdat een optimale oplossing gevonden is. We weten dat een optimale oplossing gevonden is zodra ……

Vraag 19) Vind door toepassing van Branch and Bound op systematische wijze een optimale oplossing voor planningsprobleem dat weergegeven wordt door Tabel 2. Bereken en noteer telkens de ondergrenzen bij de ovalen waarvoor ze uitgerekend worden. Licht toe waarom je weet dat een optimale oplossing gevonden is en je niet verder hoeft te zoeken.

Belangrijke hint: Voor het toepassen van Branch and Bound moeten vaak ondergrenzen berekend worden. Het is dus van belang om hier een handige, systematische wijze voor te vinden

Vraag 20*) Voor welk percentage van de ovalen uit vraag 9 hoef je door het toepassen van Branch and Bound in het planningsprobleem dat weergegeven wordt door Tabel 2 geen ondergrens te berekenen? En voor welk percentage van de mogelijke planningen hoef je door het toepassen van Branch and Bound in dit probleem geen oplossingswaarde te berekenen?

Vraag 21*) Door het uitrekenen van ondergrenzen heb je uitgevonden dat voor een bepaald planningsprobleem met 6 taken de planning 5-6-3-1-4-2 beter is dan elke mogelijke planning die begint met taak 1, 2, 3 of 4. Voor hoeveel mogelijke planningen hoef je nu sowieso geen oplossingswaarde te berekenen in je zoektocht naar een optimale oplossing?

In deel 1 van deze praktische opdracht hebben jullie kennis gemaakt met twee oplosstrategieën voor planningsproblemen: complete enumeratie en heuristieken. Het nadeel van complete enumeratie is dat het heel lang duurt; kijk nog maar eens terug naar vraag 6. Heuristieken zijn een alternatief dat dit nadeel niet kent. Aan het gebruik van heuristieken kleeft echter een ander nadeel; namelijk dat je niet per se een optimale of zelfs maar een goede oplossing voor het probleem vindt. Planningsproblemen zijn lastige problemen in die zin dat er, in tegenstelling tot sommige andere problemen, geen “slimme oplossingsmethoden” bekend zijn die gegarandeerd sneller tot een optimale oplossing komen dan complete enumeratie. Echter, door toepassing van Branch and Bound hoeven voor veel problemen oplossingswaarden berekend te worden voor slechts een klein deel van alle mogelijke oplossingen. Voor veel problemen is Branch and Bound dus een stuk efficiënter dan complete enumeratie (over het algemeen geldt dat voor grotere probleeminstanties de efficiëntie ten opzichte van complete enumeratie toeneemt). Deze efficiëntie dankt Branch and Bound aan de ondergrenzen, die gebruikt worden om te bepalen dat alle mogelijke vertakkingen van bepaalde ovalen niet meer bekeken hoeven te worden.

NB. Naast de ondergrenzen bestaan er andere manieren om ovalen en al hun mogelijke vertakkingen vroegtijdig af te schrijven. Gezien de tijd gaan we hier tijdens deze werkmiddag niet verder op in.

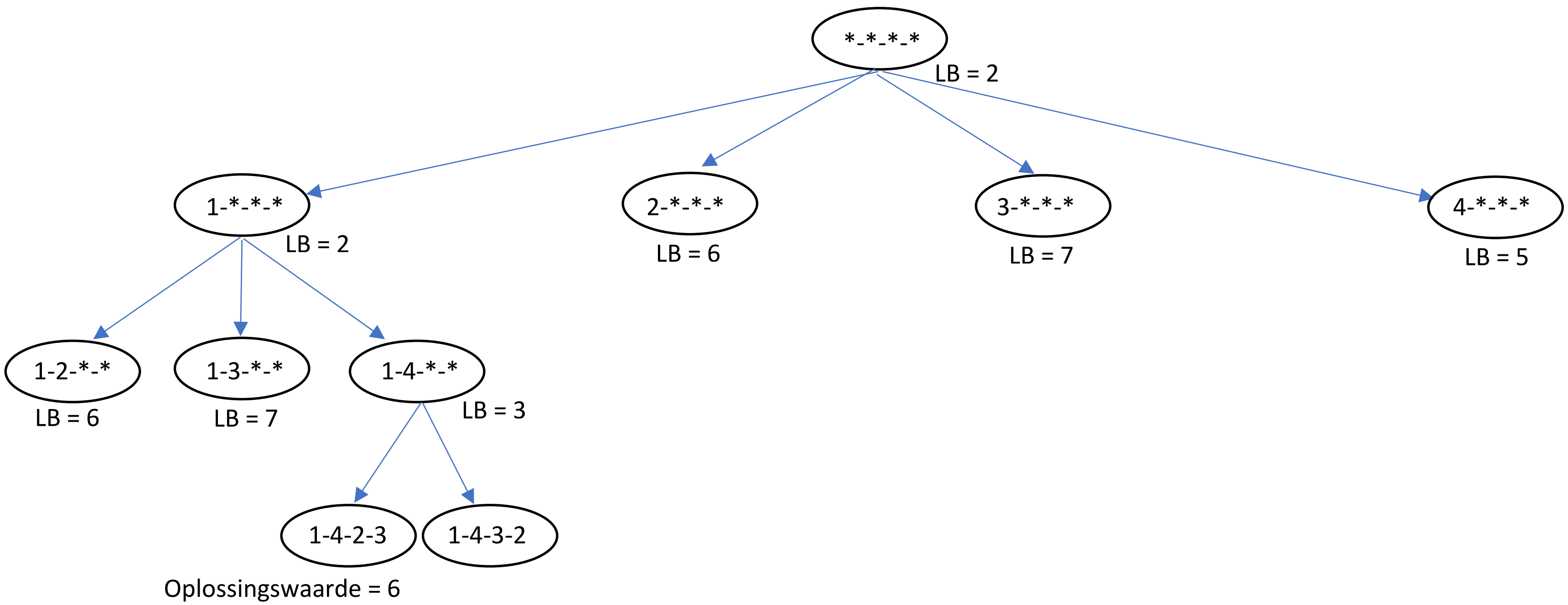
Vraag 22) Stel dat je bezig bent met het toepassen van Branch and Bound op een planningsprobleem en aangekomen bent bij de situatie die wij kunnen weergeven als in de figuur op pagina 13. De volgende stap is om de oplossingswaarde te berekenen voor het ovaal “1-4-3-2”. Wat er na deze stap gebeurt, hangt af van die oplossingswaarde. Beschrijf welke waarden deze oplossingswaarde kan hebben en wat er op basis van de verschillende mogelijke oplossingswaarden zal gebeuren in het vervolg van Branch and Bound (alleen de eerste stap van het vervolg).

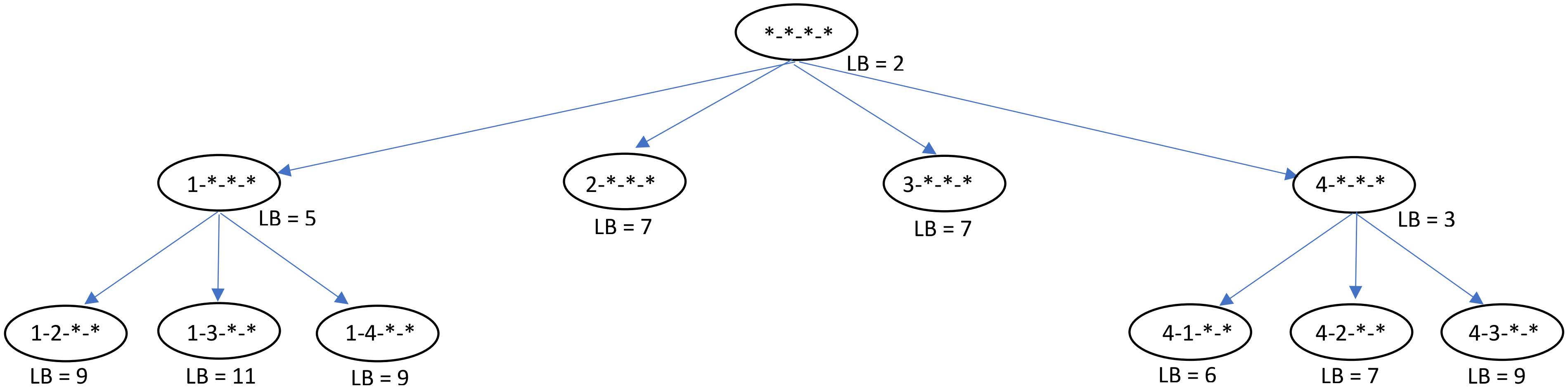
Vraag 23) Stel dat je bezig bent met het toepassen van Branch and Bound op een planningsprobleem en aangekomen bent bij de situatie die wij kunnen weergeven als in de figuur op pagina 14. Wat is de laatste stap die gedaan is om tot de huidige situatie te komen? En wat is nu de volgende stap? Zoals altijd geldt, licht jullie antwoord duidelijk toe.

Vraag 24*) Pas Branch and Bound toe op de taken uit Tabel 5 om een optimale planning te vinden. Geef hierbij duidelijk alle stappen en de volgorde van deze stappen aan.

Tabel 5: drie taken

taak	1	2	3
p	2	4	7
r	3	8	0
d_i	6	12	14


--*-*
LB = 2
1-*-*-*
LB = 2
2-*-*-*
LB = 6
3-*-*-*
LB = 7
4-*-*-*
LB = 5
1-2-*-*
LB = 6
1-3-*-*
LB = 7
1-4-*-*
LB = 3
1-4-2-3
1-4-3-2
Oplossingswaarde = 6


--*-*
LB = 2
1-*-*-*
LB = 5
2-*-*-*
LB = 7
3-*-*-*
LB = 7
4-*-*-*
LB = 3
1-2-*-*
LB = 9
1-3-*-*
LB = 11
1-4-*-*
LB = 9
4-1-*-*
LB = 6
4-2-*-*
LB = 7
4-3-*-*
LB = 9

Deel 3: Branch and Bound toepassen op andere problemen

Jullie hebben nu geleerd hoe je met behulp van Branch and Bound een optimale oplossing van een planningsprobleem kunt vinden. Het mooie van Branch and Bound is dat deze methode om een optimale oplossing van een probleem te vinden toe te passen is op een breed scala aan problemen. Ook mooi is dat je slechts een klein beetje kennis van programmeren hoeft te hebben om Branch and Bound te implementeren, onafhankelijk van de programmeertaal die je geleerd hebt. Het is zelfs zo dat je helemaal geen kennis van programmeren hoeft te hebben om Branch and Bound te gebruiken, zolang je voor een optimalisatieprobleem de volgende twee aspecten weet te verzinnen:

1. Een manier om te "branchen" of vertakken en zo stap voor stap steeds completere deeloplossingen te creëren.

2. Een manier om ondergrenzen te bepalen voor de deeloplossingen.

Zodra je deze twee aspecten weet te verzinnen, zijn er voldoende programma's of programmeurs beschikbaar die de computer met behulp van Branch and Bound een optimale oplossing kunnen laten vinden voor het optimalisatieprobleem. Heel veel problemen kunnen zo geformuleerd worden dat ze met Branch and Bound op te lossen zijn. Voorbeelden hiervan zijn het vinden van optimale locaties voor radio- en telefoonmasten of wifihotspots in de binnenstad en het bepalen van optimale routes voor bijvoorbeeld vrachtschepen, vuilniswagens of fietskoeriers. Hoe efficiënt Branch and Bound werkt ten opzichte van een computer domweg alle mogelijke oplossingen te laten bekijken, hangt allereerst af van het probleem zelf maar is vooral afhankelijk van hoe goed de ondergrenzen zijn die berekend worden.

Vraag 25*) Hoe efficiënt Branch and Bound werkt ten opzichte van een computer domweg alle mogelijke oplossingen te laten bekijken, hangt voornamelijk af van hoe "goed" de ondergrenzen zijn die berekend worden. Wanneer zou je een ondergrens "goed" noemen, in de zin dat hij de efficiënte vergroot?

Als je tegen een optimalisatieprobleem aanloopt dat je niet met de hand op kunt lossen en het een computer veel tijd kost om alle mogelijke oplossingen na te gaan, is het waarschijnlijk de moeite waard om te kijken of je Branch and Bound toe kunt passen. Hiertoe hoef je enkel de eerdergenoemde twee aspecten te bedenken: een manier om te vertakken en een manier om ondergrenzen te bepalen. Dat is geen grote klus, maar toch vaak lastiger dan het lijkt. In vraag 21 is een begin gemaakt met Branch and Bound om een bepaald probleem op te lossen. Een manier om te vertakken en een manier om ondergrenzen te bepalen zijn dus al bedacht, anders had dit begin niet gemaakt kunnen worden. Aan jullie de taak om op basis van wat gegeven is te bedenken hoe deze keuzes gemaakt zijn en vervolgens Branch and Bound door te zetten om een optimale oplossing te vinden.

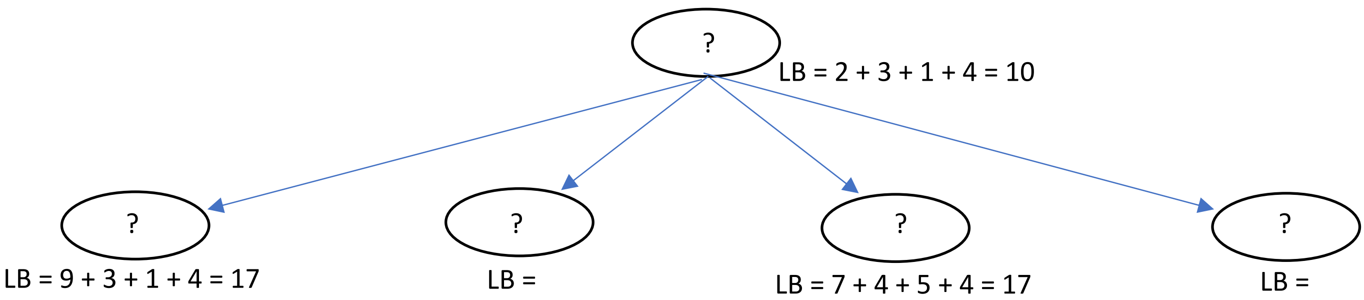
Vraag 26*) Inspecteer het probleem en de gedeeltelijke uitwerking met Branch and Bound op de volgende pagina. Leg uit welke manier van vertakken en welke manier om ondergrenzen te bepalen gebruikt zijn en werk Branch and Bound verder uit in jullie verslag. Geef hierbij duidelijk alle stappen en de volgorde van deze stappen aan.

Vier studenten hebben de luxe dat zij onderling vier verschillende functies mogen verdelen bij het bedrijf waar zij voor willen werken. Alle vier de studenten vinden elk van deze functies ongeveer even aantrekkelijk, maar er is een probleem. Om de functies uit te kunnen voeren, zullen zij nog even door moeten studeren. Iets waar zij allen niet zo op zitten te wachten.

Wegens verschillende vooropleidingen zit er nogal een verschil in studietijd die de vier functies van de verschillende studenten vereisen. In Tabel 3 is de vereiste studietijd in jaren per student per functie weergegeven. De studenten besluiten om de vier functies op zo'n manier te verdelen dat hun totale studietijd zo kort mogelijk is. Om er zeker van te zijn dat zij de verdeling van functies vinden die overeenkomt met de kortst mogelijke totale studietijd zonder dat zij alle mogelijke verdelingen hoeven te bekijken, passen de studenten Branch and Bound toe. Het begin van deze procedure is weergegeven onderaan deze pagina; hierbij is een deel weggevallen achter de vraagtekens. Ga om vraag 21 te kunnen beantwoorden allereerst na wat er onder de vraagtekens moet staan. Bedenk vervolgens hoe de ondergrenzen berekend worden en werk tot slot Branch and Bound verder uit.

Tabel 6: vereiste studiejaren per student per functie

	Functie A	Functie B	Functie C	Functie D
	9	2	7	8
	6	4	3	7
	5	8	1	8
	7	6	9	4



Bijlage F: Correctievoorschrift - versie 2

Een aantal opmerkingen vooraf:

- Waar om een correcte berekening/toelichting wordt gevraagd, hoeft het niet zo gedaan te zijn als in dit antwoordmodel, zolang duidelijk wordt hoe de leerlingen op hun antwoord zijn gekomen.
- Vragen of delen van vragen, zijn eveneens slechts 1 punt waard. Hieronder een lijst met vragen waarbij van de docent verwacht wordt dat hij leerlingen door deze vragen loodst met extra uitleg als zij er niet uit komen als zij er niet zelf uitkomen. Dit is belangrijk voor het verdere begrip van de opdracht.
- Bij vragen die niet in onderstaande lijst voorkomen (in zowel de opdracht als het correctieformulier aangegeven met sterretjes voor de vraag), wordt van de docent verwacht dat hij de leerlingen niet helpt. Hierdoor is er meer tijd om leerlingen die het nodig hebben extra uitleg te verschaffen bij bovenstaande vragen en wordt er bovendien voor gezorgd dat de behaalde cijfers een betere weerspiegeling zijn van het niveau van de leerlingen.
    - Vraag 1
    - Vraag 2
    - Vraag 3
    - Vraag 5
    - Vraag 12
    - Vraag 15 -> antwoord in vraag gegeven, laten zien dat het klopt
    - Vraag 16 -> aangegeven in vraag: check antwoord bij docent (gebruik dit moment van controle om leerlingen die dat nodig hebben de goede richting op te sturen, vraag hen bijvoorbeeld om te laten zien hoe zij een oplossingswaarde berekenen)
    - Vraag 17
    - Vraag 18
    - Vraag 19
    - Vraag 22 (niet te snel helpen, leerlingen aansporen nog eens goed terug te lezen)
    - Vraag 23 (slechts hulp bieden bij een van de twee deelvragen)

Vraag 1) Na hoeveel tijdseenheden is de machine klaar indien eerst taak 2 wordt uitgevoerd en vervolgens taak 1? Houd er rekening mee dat een taak niet gestart mag worden voor zijn lanceermoment.

1 punt voor een correct antwoord

Na 8 tijdseenheden.

Vraag 2) Uit Tabel 2 volgt dat de machine nooit alle taken voor hun vervalmomenten uit kan voeren. Leg dit uit.

1 punt voor een correcte uitleg

De gezamenlijke duur van de taken is 4 + 2 + 6 + 5 = 17 terwijl de vervalmomenten van alle taken kleiner zijn.

Vraag 3) Als we de taken uit Tabel 2 inplannen in numerieke volgorde, dus in de volgorde 1-2-3-4, wat is dan de vertraging van ieder van de taken?

1 punt voor het correcte antwoord met een correcte berekening/toelichting

Taak	Afrondmoment	Vervalmoment	Vertraging
1	4	8	nvt
2	6	12	nvt
3	12	11	1
4	17	10	7

Vraag 4) Op pagina 4 is genoemd dat het noodzakelijk is om een criterium op te stellen aan de hand waarvan bepaald kan worden hoe goed een bepaalde planning is en dat het per situatie verschilt wat een goede planning is. Hieronder worden drie mogelijke criteria genoemd. Welk van deze criteria is het meest geschikt voor Sarah? En welke zou je Dirks, Draaisma en Van Dam aanraden om te gebruiken?*

2 punten als volgt toegekend:

Criterium 1 werkt het beste voor Sarah (1 punt)

Criterium 3 werkt het beste voor de heren 3D (1 punt)

Vraag 5) Overtuig jezelf ervan dat 1-3-4-2 een optimale planning is voor de taken in Tabel 2. Wat is de oplossingswaarde behorende bij deze planning?

1 punt voor het correcte antwoord voorzien van enige toelichting

Oplossingswaarde = max. vertraging = 5 en een optimale planning is 1-3-4-2.

NB. Geen punten toekennen voor 1-4-3-2, dit is niet optimaal vanwege het vervalmoment van taak 4

NB. Wel punten toekennen als verderop in het verslag geconcludeerd wordt dat het antwoord op vraag 5 fout is, het correcte antwoord te vinden is in de Branch and Bound uitwerking verderop en daarbij terugverwezen wordt naar deze vraag.

Vraag 6) Voor hoeveel mogelijke planningen moet de computer L_{max} uitrekenen in ons voorbeeld als we de hierboven beschreven strategie toepassen? Laat jullie berekening zien.*

1 punt voor het correcte antwoord met een correcte berekening

Aantal verschillende manieren waarop de taken ingepland kunnen worden is 4! = 24.

Vraag 7) Stel dat we in opdracht van een fabrikant een optimale planning willen vinden voor de 20 taken die zijn machine moet uitvoeren. Neem aan dat de computer waarover we beschikken vijftig nanoseconden (een nanoseconde is één-miljardste van een seconde) nodig heeft om L_{max} uit te* *rekenen voor een mogelijke planning. Hoe lang heeft de computer dan nodig om met de hierboven beschreven strategie een optimale planning voor de 20 taken te vinden? Geef jullie antwoord in een zo groot mogelijke doch logische tijdseenheid.*

2 punten voor het correcte antwoord met een correcte berekening

Aantal verschillende manieren waarop de taken ingepland kunnen worden is 20!. 20! x 50 = 121645100408832000000 nanoseconden, oftewel 3857 jaar (3855 Gregoriaanse jaren).

NB. 1 punt aftrek voor een correct antwoord in een andere tijdseenheid dan jaren (decennia, eeuwen en millennia mogen wel, mits correct afgerond).

NB. ½ punt aftrek voor een nagenoeg correcte berekening maar een fout antwoord, bijv. door een verkeerd aantal nullen achter de komma bij het rekenen met nanoseconden.

NB. Geen punten aftrek voor een iets preciezere of iets minder precies antwoord bijv. door schrikkeljaren mee te nemen in de berekening

Vraag 8) Pas de hierboven beschreven heuristiek toe op de vier taken uit Tabel 2. Geeft de heuristiek een optimale oplossing?*

1 punt voor het correcte antwoord met een correcte berekening/toelichting

Ja, de heuristiek geeft een optimale oplossing (zoals eerder bepaald in vraag 5).

Vraag 9) Precisie is in de wiskunde en de informatica erg belangrijk. De hierboven beschreven heuristiek is niet geheel eenduidig. Voor bepaalde planningsproblemen zal niet duidelijk zijn wat er moet gebeuren. Een computer die het stappenplan volgt zal dan waarschijnlijk vastlopen. Verzin voor de vier taken uit Tabel 2 nieuwe lanceer- en vervalmomenten zodanig dat de hierboven beschreven heuristiek niet eenduidig voorschrijft wat er moet gebeuren.*

2 punten voor een correct antwoord

Stap 2 is niet eenduidig. Het kan namelijk zo zijn dat er meerdere taken met het vroegste vervalmoment kunnen zijn.

Vraag 10) Leg uit hoe je kunt laten zien dat een heuristiek zwak is. Verzin voor de vier taken uit Tabel 2 nieuwe lanceer- en vervalmomenten zodanig dat de zwakte van deze heuristiek ten opzichte van complete enumeratie duidelijk wordt.*

3 punten als volgt toegekend:

Dat een heuristiek zwak is kun je laten zien door een voorbeeldprobleem te geven waarop de heuristiek geen optimale oplossing geeft (1 punt).

Een voorbeeld van correcte nieuwe lanceer- en vervalmomenten (2 punten):

i	1	2	3	4

p_i	4	2	6	5
r_i	1	0	0	0
d_i	5	20	20	20

Vraag 11) Bedenk een andere mogelijke heuristiek die toegepast kan worden op het planningsprobleem. Noteer deze stapsgewijs en pas hem toe op de vier taken uit Tabel 2.*

2 punten als volgt toegekend:

Heuristiek duidelijk uitgelegd (1 punt).

Heuristiek correct toegepast op de vier taken uit Tabel 1 (1 punt).

Vraag 12) Bepaal op een systematische manier alle mogelijke manieren waarop de taken uit Tabel 2 ingepland kunnen worden. Neem hiertoe onderstaande figuur over in jullie verslag en breidt de ovalen 1--*-* en 2-*-*-* uit. De kleuren van de ovalen mogen jullie voorlopig negeren.*

1 punt voor enigszins nette en overzichtelijke, maar vooral correcte tekening

Zie de linkerhelft van het plaatje na vraag 19.

NB. Geen punten aftrekken indien ovalen in correcte generatie zitten maar in een andere volgorde staan (1234 en 1243 verwisseld bijvoorbeeld).

Vraag 13) Wat voor verband zien jullie tussen jullie antwoord op vraag 4 en de bovenstaande figuur?*

1 punt voor een correct antwoord

Het aantal ovalen in de onderste generatie is gelijk aan het aantal mogelijke oplossingen.

Vraag 14) Stel dat er in plaats van vier taken vijf taken ingepland moeten worden. Hoeveel ovalen krijg je dan in totaal als je een figuur tekent zoals die in vraag 9? (Hint: Het is niet nodig om deze figuur te tekenen en dit doen levert geen punten op, je kunt het antwoord beredeneren.)*

2 punten als volgt toegekend:

In de bovenste generatie bevindt zich 1 ovaal.

In de volgende generatie bevinden zich 5 ovalen.

In de volgende generatie bevinden zich 5 x 4 ovalen. (1 punt)

In de volgende generatie bevinden zich 5 x 4 x 3 ovalen.

In de onderste generatie bevinden zich 5 x 4 x 3 x 2 ovalen.

Het totaal aantal ovalen is dan 1 + 5 + 5 x 4 + 5 x 4 x 3 + 5 x 4 x 3 x 2 = 206

Vraag 15) Plan de taken uit Tabel 3 zodanig in dat L_{max} zo klein mogelijk is. Sta hierbij pre-emption toe. Als het goed is, vinden jullie $L_1 = L_3 = 1$ en $L_2 = 0$, dus $L_{max} = 1$. Mocht dit niet zo zijn, vraag de docent dan even om jullie te helpen.

1 punt voor een correcte uitwerking waaruit blijkt dat $L_1 = L_3 = 1$ en $L_2 = 0$.

NB. Leerlingen op weg helpen kan bij deze vragen door hen een tabel te laten tekenen en daarin aan te geven van welke taak er op elk minimaal tijdsinterval (0-1, 1-2, etc.) een “stukje” ingepland wordt.

Vraag 16) Zie nu het planningsprobleem dat weergegeven wordt door Tabel 4. Stel dat we weten dat taak 1 als eerste ingepland en afgerond moet worden. Plan gebaseerd op dit gegeven de taken uit Tabel 4 zodanig in dat L_{max} zo klein mogelijk is. Taak 1 wordt als eerste ingepland; pre-emption is toegestaan zodra taak 1 is afgerond. Wat is nu de kleinst mogelijk waarde voor L_{max}? Laat jullie antwoord even checken door de docent.

1 punt voor een correcte uitwerking waaruit blijkt dat $L_{max} = 5$ ($L_1 = 0$, $L_2 = 2$ en $L_3 = 5$).

NB. Leerlingen op weg helpen kan bij deze vragen door hen een tabel te laten tekenen en daarin aan te geven van welke taak er op elk minimaal tijdsinterval (0-1, 1-2, etc.) een “stukje” ingepland wordt.

Vraag 17) Combineer de heuristiek uit deel 1 van deze praktische opdracht met pre-emption om een planning te verkrijgen voor het planningsprobleem dat weergegeven wordt door Tabel 2. De oplossingswaarde behorende bij deze planning is een ondergrens voor dit planningsprobleem.

1 punt voor het correcte antwoord met een correcte berekening/toelichting

LB = 5.

Vraag 18) Hieronder wordt in een aantal stappen de werking van Branch and Bound verder toegelicht. Er ontbreken echter hier en daar iets. Ga na wat er op de puntjes zou moeten staan en vul aan:

1. *Kies uit de nog niet vertakte ovalen het ovaal met …….*
2. *Vertak de gekozen ovaal en bereken ondergrenzen voor alle nieuwe ovalen die uit deze vertakking voortkomen. Indien ……, bereken je …… in plaats van een ondergrenzen voor alle nieuwe ovalen die uit deze vertakking voortkomen.*
3. *Herhaal stappen 1 en 2 totdat een optimale oplossing gevonden is. We weten dat een optimale oplossing gevonden is zodra ……*

2,5 punten als volgt toegekend:

Op de eerste rij puntjes moet staan: “de laagste ondergrens” (0,5 punt).

Op de tweede rij puntjes moet staan: “de ovaal een mogelijke oplossing representeerd/tot de laagste generatie behoort” (0,5 punt).

Op de derde rij puntjes moet staan: “de oplossingswaarde” (0,5 punt).

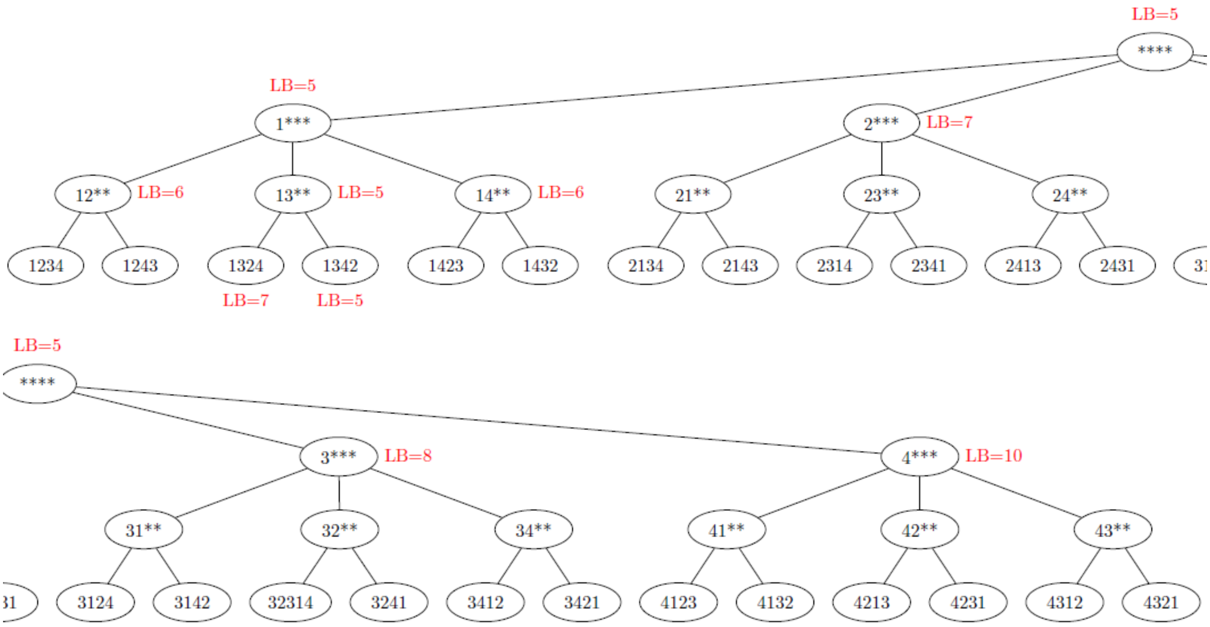
Op de vierde rij puntjes moet staan: “er geen ovalen meer zijn waarvan de ondergrens lager is dan de best gevonden oplossingswaarde tot nu toe” (1 punt).

Vraag 19) Vind door toepassing van Branch and Bound op systematische wijze een optimale oplossing voor planningsprobleem dat weergegeven wordt door Tabel 2. Bereken en noteer telkens de ondergrenzen bij de ovalen waarvoor ze uitgerekend worden. Licht toe waarom je weet dat een optimale oplossing gevonden is en je niet verder hoeft te zoeken.

2,5 punten toegekend als volgt:

Er zijn geen ontbrekende of overbodige vertakkingen en de LB’s zijn uitgerekend voor de correcte ovalen, zie de tekening (1 punt).

De LB’s zijn allemaal correct uitgerekend, zie de tekening (gesplitst in een linker- en rechtergedeelte) hieronder (1 punt).



De computer ‘weet’ dat een optimale oplossing gevonden is doordat er geen (onvertakte) ovalen meer zijn waarvoor de LB lager is dan de oplossingswaarde in het ovaal 1-3-4-2 (0,5 punt).

NB. Per ontbrekende of overbodige vertakking of per fout gekozen LB om uit te rekenen -0,5 met een max. van -1

NB. Per fout uitgerekende LB -0,5 met een max. van -1

Vraag 20) Voor welk percentage van de ovalen uit vraag 9 hoef je door het toepassen van Branch and Bound in het planningsprobleem dat weergegeven wordt door Tabel 2 geen ondergrens te* *berekenen? En voor welk percentage van de mogelijke planningen hoef je door het toepassen van Branch and Bound in dit probleem geen oplossingswaarde te berekenen?*

1 punt als volgt toegekend:

0,5 punt voor het correcte antwoord met een correcte berekening/toelichting van deelvraag 1: 31 van de 41 ovalen hoeft de computer niet te bekijken, dit is ongeveer 75,6%.

0,5 punt voor het correcte antwoord met een correcte berekening/toelichting van deelvraag 2: 22 van de 24 oplossingswaarden hoeft de computer niet te berekenen, dit is ongeveer 91,7%.

Vraag 21) Door het uitrekenen van ondergrenzen heb je uitgevonden dat voor een bepaald planningsprobleem met 6 taken de planning 5-6-3-1-4-2 beter is dan elke mogelijke planning die begint met taak 1, 2, 3 of 4. Voor hoeveel mogelijke planningen hoef je nu sowieso geen oplossingswaarde te berekenen in je zoektocht naar een optimale oplossing?*

2 punten voor een correct antwoord met een correcte berekening/toelichting

6! x 4/6.

Vraag 22) Stel dat je bezig bent met het toepassen van Branch and Bound op een planningsprobleem en aangekomen bent bij de situatie die wij kunnen weergeven als in de figuur op pagina 13. De volgende stap is om de oplossingswaarde te berekenen voor het ovaal "1-4-3-2". Wat er na deze stap gebeurt, hangt af van die oplossingswaarde. Beschrijf welke waarden deze oplossingswaarde kan hebben en wat er op basis van de verschillende mogelijke oplossingswaarden zal gebeuren in het vervolg van Branch and Bound (alleen de eerste stap van het vervolg).

2 punten toegekend als volgt:

Mogelijke situatie 1: De oplossingswaarde is 3, 4 of 5. In dat geval is de planning 1-4-3-2 een optimale planning en is Branch and Bound afgerond (1 punt).

Mogelijke situatie 2: De oplossingswaarde is hoger dan 5. In dat geval is de volgende stap het vertakken van ovaal 4-*-*-* (1 punt).

NB. 1 punt aftrekken als er een derde situatie bij is gekomen.

Vraag 23) Stel dat je bezig bent met het toepassen van Branch and Bound op een planningsprobleem en aangekomen bent bij de situatie die wij kunnen weergeven als in de figuur op pagina 14. Wat is de laatste stap die gedaan is om tot de huidige situatie te komen? En wat is nu de volgende stap? Zoals altijd geldt, licht jullie antwoord duidelijk toe.

2 punten toegekend als volgt:

De voorgaande stap was het vertakken van ovaal "1-*-*-*" (1 punt).

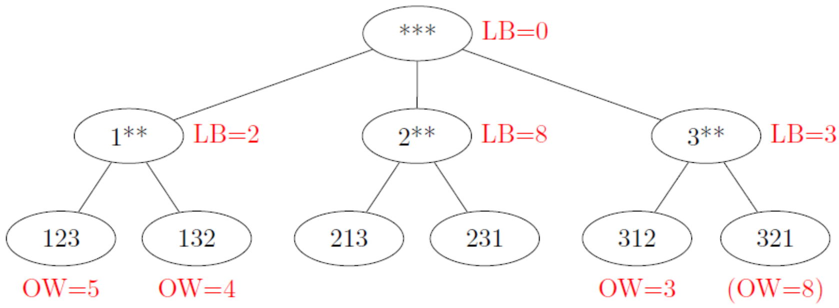
De volgende stap is het vertakken van het ovaal "4-1-*-*" (1 punt).

Vraag 24) Pas Branch and Bound toe op de taken uit Tabel 5 om een optimale planning te vinden. Geef hierbij duidelijk alle stappen en de volgorde van deze stappen aan.*

2 punten toegekend als volgt:

Er zijn geen ontbrekende of overbodige vertakkingen en de LB's zijn uitgerekend voor de correcte ovalen, zie de tekening op de volgende volgende pagina (1 punt).

De LB's zijn allemaal correct uitgerekend, zie de tekening hieronder (1 punt).



NB. Per ontbrekende of overbodige vertakking of per fout gekozen LB om uit te rekenen -0,5 met een max. van -1

NB. Per fout uitgerekende LB -0,5 met een max. van -1

Vraag 25) Hoe efficiënt Branch and Bound werkt ten opzichte van een computer domweg alle mogelijke oplossingen te laten bekijken, hangt voornamelijk af van hoe "goed" de ondergrenzen zijn die berekend worden. Wanneer zou je een ondergrens "goed" noemen, in de zin dat hij de efficiënte vergroot?*

1 punt voor een duidelijke en correcte uitleg

Ondergrenzen moeten zo hoog mogelijk zijn (maar nog wel de eigenschap van ondergrens bewaren in die zin dat een optimale oplossing nooit lager kan zijn dan een lower bound).

Vraag 26) Inspecteer het probleem en de gedeeltelijke uitwerking met Branch and Bound op de volgende pagina. Leg uit welke manier van vertakken en welke manier om ondergrenzen te bepalen gebruikt zijn en werk Branch and Bound verder uit in jullie verslag. Geef hierbij duidelijk alle stappen en de volgorde van deze stappen aan.*

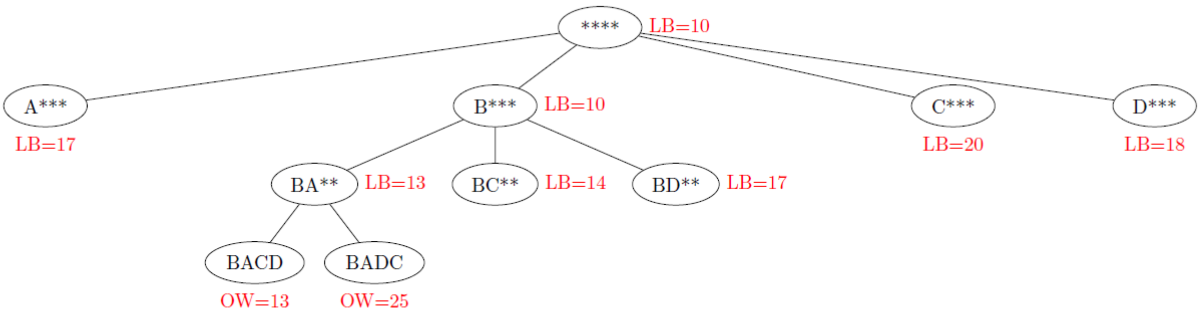
4 punten toegekend als volgt:

De manier van vertakken is correct en duidelijk uitgelegd: in de eerste vertakking wordt persoon 1 toegewezen aan een functie, in de tweede vertakking wordt persoon 2 toegewezen aan een nog niet bezette functie, etc. (1 punt).

De manier van lower bounds bepalen is correct en duidelijk uitgelegd: aan iedere student wordt de functie die van hem of haar het minste aantal studiejaren vereist toegewezen, mits deze functie nog beschikbaar is (1 punt).

Er zijn geen ontbrekende of overbodige vertakkingen en de LB's zijn uitgerekend voor de correcte ovalen, zie de tekening hieronder (1 punt).

De LB's zijn allemaal correct uitgerekend, zie de tekening op de volgende pagina (1 punt).



NB. In de houdt de positie van de letter verband met de positie van de student in Tabel 3 uit de praktische opdracht. Zo staat BA** voor de situatie waarin persoon 1 functie B toegewezen krijgt, persoon 2 functie A toegewezen krijgt en personen 3 en 4 nog geen functie toegewezen hebben gekregen.

NB. Per ontbrekende of overbodige vertakking of per fout gekozen LB om uit te rekenen -0,5 met een max. van -1

NB. Per fout uitgerekende LB -0,5 met een max. van -1

Bijlage G: Brief aan ouders

Beste ouder(s)/verzorger(s),

Zoals voor de meeste vakken, heeft uw zoon/dochter dit jaar ook voor het vak Wiskunde B een werkmiddag (op 28 maart 2017). Het idee achter zo'n middag is om eens op een andere manier met het vak bezig te zijn, qua onderwerp en/of werkvorm.

Dit jaar heeft onze collega meneer Van der Meulen (eerstegraads docent wiskunde in opleiding) in het kader van zijn afstudeeronderzoek een nieuwe opdracht voor de werkmiddag van Wiskunde B ontwikkeld, in samenspraak met mijzelf. Meneer Van der Meulen wil graag voor zijn onderzoek de verslagen van de leerlingen bekijken en na afloop van de werkmiddag en/of in de dagen daaropvolgend enkele leerlingen interviewen betreffende hun ervaringen met de opdracht. De interviews zullen worden opgenomen (audio) om het mogelijk te maken deze later te transcriberen. Vanzelfsprekend hebben de antwoorden van de leerlingen geen effect op hun beoordeling en wordt alles vertrouwelijk behandeld. De opnamen en transcripties zullen geanonimiseerd worden opgeslagen. Ook eventuele interessante resultaten uit de verslagen zullen anoniem worden verwerkt in het onderzoek.

Als u akkoord gaat met het gebruik van het verslag van uw kind voor het verder verbeteren van de werkmiddag, en met de mogelijkheid dat uw kind geïnterviewd wordt, dan hoeft u niets te doen. Mocht u hier wel bezwaren tegen hebben, dan kunt u dat kenbaar maken via een e-mail aan mij (m.timmer@carmelcollegesalland.nl), liefst uiterlijk op vrijdag 24 maart. U kunt uw bezwaren ook kenbaar maken door onderstaand strookje door uw zoon/dochter bij mij te laten inleveren, liefst uiterlijk op vrijdag 24 maart. Uiteraard kan uw zoon/dochter zelf nog op ieder moment beslissen niet aan het onderzoek te willen deelnemen.

Ik hoop u hiermee voldoende te hebben geïnformeerd. Voor vragen kunt u contact opnemen via het bovengenoemde e-mailadres.

Met vriendelijke groet,
Mark Timmer, docent wiskunde

Ik geef **geen** toestemming voor …………………………………………. uit klas ………… om geïnterviewd te worden betreffende zijn/haar ervaringen met de werkmiddag van Wiskunde B en/of voor dat zijn/haar verslag gebruikt wordt ten behoeve van het genoemde onderzoek.

Naam: …………………………………………

Handtekening: …………………………………………

Bijlage H: Verslag van walkthrough

Hieronder het verslag van de walkthrough van ons ontwerp met een eventuele belanghebbende. Zoals toegelicht in Hoofdstuk 4, hebben wij de eventuele belanghebbende bij ons ontwerp de praktische opdracht laten maken en daarbij in het opdrachtenboekje opvallende punten laten noteren of aanstippen. Vervolgens hebben wij samen met de eventuele belanghebbende bij ons ontwerp deze punten doorgelopen en erover gediscussieerd. Belangrijke punten zijn door de eventuele belanghebbende verder uitgewerkt in onderstaand verslag. Alhoewel de punten die hierin naar voren komen zoals genoemd is samen met de onderzoeker doorlopen en bediscussieerd zijn, geven zij de mening van de eventuele belanghebbende weer en niet per se die van de onderzoeker.

Walkthrough van de praktische opdracht Branch and Bound voor 5 vwo wiskunde B

Over het algemeen vinden we de praktische opdracht over Branch and Bound een didactische en inhoudelijk goede opdracht. Er wordt voor de leerlingen een goed voorstelbaar en praktisch probleem geïntroduceerd (1^e^ gezicht van de wiskunde (Ziegler & Loos, 2014)) en dit wordt mooi gekoppeld aan de mogelijkheden en beperkingen van hedendaagse ICT-oplossingen, een gebied waarin mogelijk een aantal van de leerlingen uit de doelgroep (5 vwo Wiskunde B) in de toekomst gaan werken (3^e^ gezicht van de wiskunde (Ziegler & Loos, 2014)). Daarnaast zijn wij persoonlijk groot fans van het soort wiskunde achter dit type problemen en hopen we dat dit ook in zekere mate geldt voor de leerlingen (2^e^ gezicht van de wiskunde (Ziegler & Loos, 2014)).

Ten tweede wordt er in de opdracht zeer sterk gestuurd op relationeel begrip (Skemp, 1977). Bij veruit de meeste opgaven is de vraagstelling dusdanig dat het antwoord niet direct duidelijk is, maar de leerlingen echt goed de situatie moeten doorgronden voordat ze een goed antwoord kunnen geven op de vraag. Dit lijkt ons een goede methode om degelijk begrip te kweken van de achterliggende gedachtes achter Branch and Bound. Zeker dit type algoritmiek leent zich erg goed om heel instrumenteel aangeleerd te worden ("maak een nieuwe tak", "bereken de nieuwe lower bound", "wat is de optimale planning als je het algoritme netjes volgt?"), maar het is goed dat de auteur van de praktische opdracht deze neiging heeft weten te onderdrukken. Uiteindelijk denken we dat de leerlingen misschien wel niet 100% soepel een algoritme kunnen uitvoeren, maar wel de finesses van de algoritmiek begrijpen, het nut er van inzien en kunnen redeneren over vergelijkbare vraagstukken.

Tenslotte zit er veel mogelijkheid tot differentiatie in de opdracht. Over het algemeen denken we dat het niveau van de opgaven erg hoog is en er zeker een aantal echte inzichtsvragen in zitten. De zwakkere leerlingen zullen dan ook wel wat hulp nodig hebben hier en daar. Het is aan de docent om een goede afweging te maken waar hij/zij hulp wenst te geven. Uiteindelijk zijn alle opgaven met enige hulp goed uit te voeren en is er ook voldoende tijd voor de docent om deze hulp te bieden. Voor de betere leerlingen zijn er voldoende zeer uitdagende opgaven te vinden in de opdracht. Met name opgaven 18 en 21 vergen echt goed begrip van de techniek van Branch and Bound en zijn daarmee uitstekende differentiatieopgaven.

Nu zullen we de opdracht in meer detail gaan analyseren. Hierbij beginnen we op pagina 3 en eigenlijk direct op de 1^e^ regel. Hier wordt het planningsprobleem geïntroduceerd met "In een fabriek staat een machine die verschillende taken moet verrichten". Hoewel hiermee wel een praktisch en goed voorstelbaar probleem wordt geïntroduceerd, had dit wat ons betreft wel wat smeuïger gemogen door een fabrieksnaam, een soort machine en enkele specifieke taken te benoemen. Misschien gecombineerd met een verhaaltje over de carrièreswitch van Arie Schoptgoed naar het bedrijfsleven waar hij meteen te maken krijgt met lastige planningsproblemen en te trage ICT. In ieder geval had de toepassing iets meer uitgemolken mogen worden om de opdracht nog wat aantrekkelijker te maken voor de leerlingen.

Daarna worden er variabelen geïntroduceerd met een subscript, zoals bijvoorbeeld pi voor de duur van taak i. Hoewel dit in onze ogen volstrekt logische variabelen zijn, zou het voor de leerlingen wel eens verassend lastig kunnen zijn om deze variabelen te doorgronden. In de lespraktijk blijken leerlingen bijvoorbeeld ook ontzettend veel moeite te hebben met functies met een parameter er in. Soms slaat de paniek volledig toe bij het zien van de functies $f_p(x) = x^2 - 3px + 4$. Het zou daarom goed kunnen dat de "symbol sense" (Drijvers, 2012) van leerlingen nog niet genoeg ontwikkeld is om deze variabelen direct goed te kunnen begrijpen en gebruiken. Het is daarom waarschijnlijk goed om hier iets langer bij stil te staan in de opdracht; het is belangrijk dat leerlingen deze notatie goed begrijpen omdat het anders afleidt van de daadwerkelijke inhoud van de opdracht.

Positief punt van het eerste stuk is dat het probleem mooi geformuleerd is in leerlingentaal. Ondanks de complexiteit van dit type problemen, verwachten we dat de leerlingen met de korte introductie toch wel direct het probleem zullen snappen.

In de 2e alinea op pagina 3 wordt beweerd dat met een klein beetje programmeerkennis het plannen gemakkelijk door een computer gedaan kan worden. We verwachten dat dit bij sommige leerlingen de nieuwsgierigheid wel prikkelt. Daarom zou het ontzettend mooi zijn als er mogelijkheden beschikbaar zijn om deze nieuwsgierigheid te bevredigen. Het is goed om hiervoor als docent vooraf over na te denken, zodat leerlingen gestimuleerd kunnen worden om het programmeergedeelte in de praktijk te gaan brengen. Dit is misschien mogelijk via het vak NLT of via een profielwerkstuk.

Vervolgens komen we aan bij vraag 1. Deze vinden we eigenlijk direct al erg uitdagend en vraagt al direct om relationeel begrip; een leerling moet inzien dat de totale duur van de taken al groter is dan het maximale vervalmoment waardoor niet alle taken op tijd kunnen zijn. Hoewel we positief zijn over de mate van relationeel begrip in de opdracht is dit misschien iets te veel van het goede. We verwachten dat de leerlingen, ook de sterkere, toch wel wat behoefte hebben aan een simpel getallenvoorbeeldje om er even in te komen. Daarom zouden we toch nog een vraag 0 willen invoegen die aan die behoefte invulling geeft. Voorbeeld van zo'n vraag 0 zou kunnen zijn: "Stel dat we de taken inplannen in de volgorde 1-2-3-4. Maak voor die planning een tijdlijn en teken de taken op deze tijdlijn. Geef op de tijdlijn ook de lanceer- en vervalmomenten aan. Wat is je conclusie over deze planning?". Via deze vraag worden ze direct gedwongen om het probleem, met een simpele opdracht, meer te visualiseren. Dit zou goed kunnen helpen bij het restant van de opdracht en geeft daarnaast ook de zwakkere leerlingen alvast wat zelfvertrouwen.

Na vraag 1 worden er inplancriteria geopperd voor het vervolg van de opdracht. Het zal leerlingen op dat moment duidelijk zijn dat niet elke verzameling taken perfect kan worden ingepland en dat er daarom soort criteria nodig zijn. We vinden het echter jammer dat er niet meer draagvlak wordt opgebouwd voor de specifieke criteria. Dit zou al makkelijk kunnen door klanten te beschrijven die behoefte hebben aan de uitgevoerde taken. Zo is het criterium "zo weinig mogelijk te late taken" makkelijk te verdedigen door de doelstelling "we willen zo weinig mogelijk teleurgestelde klanten". Ook het criterium "we willen het maximale aantal tijdseenheden die tussen de afronding en het vervalmoment van een taak zitten, zo klein mogelijk maken" komt enigszins uit de lucht vallen.

Mogelijke verklaring voor dit criterium is dat klanten vaak een beetje vertraging nog wel accepteren, maar moeite hebben met grote vertragingen. Het zou dus mooi zijn als er bij het introduceren van de criteria een duidelijkere link met de toepassing zou worden gemaakt, zodat leerlingen ook begrijpen hoe deze criteria in de praktijk kunnen ontstaan.

Op pagina 4 vinden we vraag 5 het vermelden waard. Met deze simpele vraag wordt in één klap onderbouwd waarom dit soort vraagstukken zo belangrijk zijn om goed en efficiënt op te lossen. Leerlingen zullen waarschijnlijk onder de indruk zijn van de snelheid van computers ("Wow, 50 miljardste van een seconde? Dat is echt snel!"), vervolgens onder de indruk zijn van het aantal mogelijke planningen bij slechts 20 taken ("Wow, zijn er bij 20 taken al 2,43 x 10^{18} mogelijkheden?!") en tenslotte onder de indruk zijn van de tijd die nodig is om die mogelijkheden door te rekenen met wat zij dachten dat een snelle computer was ("Wow, heb je 38,57 eeuwen nodig om dit uit te rekenen met een computer?!"). In dit kader is het ook leuk om er nog een keer een schepje bovenop te doen met een leuke visualisatie van het aantal mogelijkheden van 52! (oftewel de mogelijkheden bij het schudden van een pakje kaarten): https://www.youtube.com/watch?v=T69cguFzZ_w. Kortom, de wiskunde achter vraag 5 geeft indrukwekkende resultaten en toont daarmee direct het belang van dit soort wiskunde in de praktijk aan; ook nu er al supersnelle computers zijn!

Op pagina 5 verwachten we dat er mooi onderscheid gemaakt kan worden tussen mindere en sterkere leerlingen. Omdat je de zwakte van een heuristiek moet zien te herkennen en dit vervolgens moet vertalen in een vervelend getallenvoorbeeld, zal er veel inzicht nodig zijn. Daarnaast geeft de vraagstelling veel ruimte voor interpretatie. We verwachten dat zwakkere leerlingen hier dan ook in de problemen zullen komen. Op zich is dit geen probleem voor de rest van de opdracht en daarom is vraag 7 een mooie differentiatievraag.

Het algemene deel 1 wordt vervolgens mooi afgesloten met een bruggetje naar Branch and Bound. Deel 1 heeft op mooie wijze het probleem geïntroduceerd en tegelijkertijd aangetoond dat we Branch and Bound nodig hebben. Door dit relationele begrip te kweken, wordt niet alleen het begrip van leerlingen goed opgebouwd, maar leerlingen blijven bovendien gemotiveerd voor de rest van de opdracht; "Branch and Bound is niet zomaar een wiskundige techniek, maar zorgt ervoor dat onze industrie efficiënt functioneert dus daar wil ik meer over weten!".

De motivatie die is opgebouwd met deel 1 wordt in onze ogen echter voor een deel afgebroken door vraag 9. We moeten als docenten leerlingen niet onnodig vermoeien met het maken van bijvoorbeeld boomdiagrammen (Coenen, Timmer, & Verhoef, 2016), vraag 9 vraagt hier echter wel om. We zouden daarom vraag 9 zo herformuleren dat ze moeten beginnen met het uitwerken van de boom op een systematische manier, totdat ze het concept begrijpen en dit kunnen uitleggen. Sterkere leerlingen zien dit mogelijk al na 2 ovaaltjes en vermoeien we op die manier niet met geestdodend tekenwerk. Wel vinden we het belangrijk dat leerlingen de hele boom en de opbouw daarvan voor zich zien; ze moeten dit dus wel goed onder woorden kunnen brengen bij vraag 9.

Op pagina 8 wordt er voor vraag 15 wel heel kort de Branch and Bound procedure uitgelegd. Op zich is dit prima, zodat leerlingen de procedure zelf leren ontdekken en daarmee relationeel begrip van dit algoritme kweken. Dit wil je ook juist bij het aanleren van een procedure. Aan de andere kant is de uitleg dusdanig kort dat leerlingen naar onze verwachting wat tijd nodig zullen hebben voor het ontdekken van de procedure. Dit is op zich niet erg, maar het is een afweging of je die tijd daarvoor wilt gebruiken.

Bij vraag 16 wordt de winst van Branch and Bound mooi duidelijk. Het is belangrijk dat leerlingen ook inzien dat deze algoritmiek goed zijn werk doet in de praktijk en daarmee de problemen van snelle, doch nog steeds veel te trage, computers grotendeels oplossen.

Bij vraag 16 is er overigens nog wel een klein gevaar bij de meest briljante leerlingen. Deze zullen mogelijk herkennen dat je in een worst-case scenario (hoewel de kans hierop erg klein is) nog steeds alle oplossingen moet uitrekenen en daarnaast nog lower bounds moet berekenen. En erger nog: in het voorbeeldprobleem is het uitrekenen van een lower bound tijdrovender dan het uitrekenen van een oplossing bij een gegeven planning. Potentieel is Branch and Bound daarmee dus zelfs langzamer dan brute-force methoden. Het zal als docent genieten zijn als leerlingen tot dit goede relationele begrip van Branch and Bound komen tijdens deze opdracht, maar het is wel belangrijk dat je als docent een goed antwoord hebt voor deze leerlingen. Dit antwoord kan relatief simpel gegeven worden door te vermelden dat voor grotere problemen (dit zijn ook de problemen waarvoor je dit soort algoritmiek nodig hebt!) de kans op veel ovaaltjes veel kleiner wordt en dat in de praktijk het uitrekenen van een lower bound vaak veel sneller gaat dan het uitrekenen van de optimale oplossing in een bepaald eindpunt van een tak.

Ook deel 3 van de opdracht wordt weer op een goede manier geïntroduceerd. Daar waar het nut van Branch and Bound bij de introductie van deel 2 werd benadrukt ("we moeten wat slims verzinnen, want onze pc's zijn niet snel genoeg"), wordt hier wederom het nut en de schoonheid van Branch and Bound benadrukt ("kijk eens hoe belangrijk deze wiskunde is, we kunnen het overal toepassen").

De eerste zin onder de genummerde bullets van pagina 12 zouden we eventueel anders formuleren. Nu staat er dat er voldoende programmeurs zijn die Branch and Bound voor je kunnen implementeren, maar het zou natuurlijk mooi zijn als leerlingen zelf dit leren programmeren bij NLT of in hun profielwerkstuk. Wat ons betreft mag er dus een wat prikkelender stuk tekst hier worden opgenomen.

Tenslotte zijn we tevreden over vragen 20 en 21. Vraag 20 zorgt ervoor dat leerlingen hun opgedane begrip goed moeten formuleren terwijl vraag 21 zoals eerder al gezegd een mooie differentiatievraag is voor de echt sterke leerlingen. Al met al vinden we de opdracht leuk en uitdagend voor 5 vwo wiskunde B leerlingen en didactisch dusdanig goed opgezet dat we verwachten dat de leerlingen goed relationeel begrip krijgen van algoritmiek en de toepassingen ervan.