Radboud Universiteit Nijmegen

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

Sudoku's

Bachelorscriptie Wiskunde

Auteur:
Robin Verbeek,
S1006643

Begeleider:
dr. Wieb Bosma

Tweede lezer:
dr. ir. Henk Don

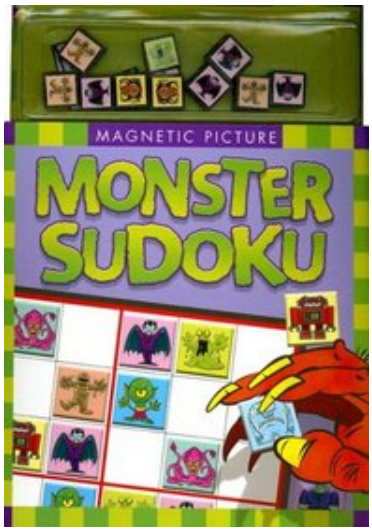
2023

Inhoudsopgave

Introductie

Sudoku's zoals we die nu kennen, zijn waarschijnlijk ontstaan uit magische vierkanten, roosters waarin de sommen van de getallen uit de rijen, kolommen en diagonalen dezelfde waarde hebben. Latijnse vierkanten zijn een variant hiervan, $n \times n$ roosters, gevuld met n symbolen waarvan ieder symbool precies één keer voorkomt in elke rij en kolom. Euler legde in de achttiende eeuw de fundamenten voor de theorie rondom Latijnse vierkanten.
In de late negentiende eeuw maakten puzzelmakers in Frankrijk van deze vierkanten puzzels door sommige getallen weg te laten, soms voegden zij ook al de 3×3 blokken toe. Deze puzzels waren enige tijd in veel Franse kranten te vinden, maar verdwenen rond het begin van de eerste wereldoorlog.
De puzzel werd weer populair toen Dell Magazines ze begon te publiceren onder de naam 'Number Place' in 1979 in Amerika. Deze puzzels volgden dezelfde regels als onze moderne variant. De auteur van deze puzzels was waarschijnlijk Howard Garns, alhoewel hij anoniem bleef en zijn naam niet vermeld werd.
In Japan werd de puzzel voor het eerst uitgegeven in 1984 onder de naam 'Sūji wa dokushin ni kagiru' door Nikoli. Dit betekent ruwweg vertaald 'getallen moeten alleen zijn' en werd verkort tot sudoku, de naam die we tegenwoordig gebruiken. De puzzel werd erg populair in Japan, waarschijnlijk omdat het Japanse schrift weinig geschikt is voor de meeste woordpuzzels.
Wayne Gould, een rechter uit Hong Kong, kwam de puzzel tegen in Tokio in 1997 terwijl hij op reis was. Hij raakte erdoor gefascineerd en ontwikkelde een computerprogramma dat sudoku's kan genereren. In 2004 wist hij de London Times te overtuigen om de puzzel te publiceren, het werdt een enorm succes en groeide uit tot de wereldwijde sensatie die wij vandaag kennen. [3]
Meer informatie is beschikbaar op Wikipedia, Latin square en Wikipedia, Sudoku history.

Persoonlijk zijn sudoku's voor mij een stukje jeugdsentiment. Mijn moeder en oma maakten ze graag en leerden mij hoe het moest. Voor de zomervakantie gaf mijn oma me een puzzelboek met magnetische monstersudoku's. Ik kan me nog steeds herinneren dat ik heerlijk ontspannen lag rond te dobberen in een zwembad in Spanje met dat puzzelboek. Sindsdien ben ik altijd dol geweest op puzzelen, dat is ook wat me aantrok tot de wiskunde.



Het puzzelboek

In mijn scriptie zal ik op een toegankelijke manier uitleggen wat een sudoku is, en een aantal strategieën voor het oplossen van sudoku behandelen. Ook zal ik kort de existentie en uniciteit van de oplossing van een puzzel bespreken. Hierna bespreek ik hoe de moeilijkheidsgraad van een sudoku bepaald wordt en hoe ik heb geprobeerd dit proces op een vereenvoudigde manier te benaderen.

1 Definities en Namen

Een Sudoku puzzel bestaat uit een $n \times n$ rooster, $n \in \mathbb{N}$, waarbij de standaard $n = 9$ is.
Hierin is een *cel* de kleinste eenheid, één van de n^2 vakjes uit het rooster.
Een *blok* is een verzameling van n cellen die dik omlijnd zijn, het 9×9 rooster is meestal ingedeeld in 9 blokken van 3×3 cellen.
Een horizontale verzameling van n cellen heet een *rij*, en een verticale verzameling van n cellen heet een *kolom*.

Bij het invullen van een standaard sudoku moet er worden voldaan aan de restrictie dat ieder symbool precies één keer moet voorkomen in ieder blok, rij en kolom. Voor de symbolen worden standaard de cijfers 1 t/m n gebruikt, maar ook letters, kleuren of andere symbolen zijn mogelijk.
Alle verzamelingen van n cellen die aan de restrictie moeten voldoen heten ook wel *groepen* of *huizen*, in een standaard sudoku zijn dit de blokken, rijen en kolommen, we zeggen ook wel dat de cellen binnen een groep elkaar kunnen 'zien'.

Hints zijn de symbolen die aan het begin van de puzzel al ingevuld zijn.
Kandidaten zijn alle symbolen die in een cel ingevuld zouden kunnen worden, deze worden meestal klein genoteerd in de cel. Preciezer zijn dit alle symbolen die op basis van gedane redenaties nog niet uitgesloten zijn. In de begintoestand van een puzzel zijn dit dus alle symbolen die niet voorkomen als hint in een van de groepen waarin de cel zich bevindt.
Het *antwoord* in een cel is het symbool wat in deze cel ingevuld moet worden om tot een oplossing te komen.
Een *oplossing* van een puzzel bestaat uit een volledig ingevuld rooster, inclusief de hints, wat aan de restrictie voldoet.

Een goede sudoku puzzel heeft een uniek bepaalde oplossing. Tenzij anders aangegeven staat, kan aangenomen worden dat het om een goede sudoku puzzel gaat.

Naast de standaard sudoku bestaan er veel varianten, zowel in formaat (bijvoorbeeld $6 \times 6, 5 \times 5, 16 \times 16$), als in de geldende restricties, bijvoorbeeld X-sudoku, waarbij de symbolen ook één keer moeten voorkomen op de diagonalen, of killer sudoku, waarbij de getallen in aangeven cellen een aangegeven som moeten hebben. Ook kunnen sudoku's andere vormen hebben, bijvoorbeeld door meerdere standaard sudoku's gedeeltelijk te laten overlappen. Tenzij anders aangegeven, zal ik me in deze scriptie beperken tot standaard 9×9 sudoku's.

2 Strategieën

Om een sudoku op te lossen kunnen we gebruik maken van strategieën, dit zijn bepaalde patronen in de kandidaten die we kunnen herkennen, op basis waarvan we antwoorden kunnen invullen of kandidaten kunnen elimineren. Om deze patronen te kunnen herkennen is het dus meestal nodig om eerst in alle cellen alle mogelijke kandidaten te noteren.

Ik zal mij hier beperken tot een aantal strategieën, deze lijst is dus niet volledig. Uitgebreidere overzichten zijn onder andere te vinden op SudokuWiki en Hodoku. Soms worden er verschillende namen voor deze strategieën gebruikt. De omschrijvingen van deze strategieën zijn gebaseerd op meerdere bronnen [7] [2] [3].

2.1 Naked/Hidden Candidates

2.1.1 Naked Singles

Wanneer alle kandidaten genoteerd zijn kan het gebeuren dat er in een cel maar één mogelijke kandidaat is, dit heet een naked single. Omdat er geen andere mogelijkheden zijn in deze cel moet deze kandidaat altijd het antwoord in deze cel zijn.

Hieronder is een voorbeeld te zien van een naked single. In de omcirkelde cel komt alleen 6 voor als kandidaat. Het antwoord in deze cel moet dus 6 zijn.

5	3 6 9 / 7	4	8	1 2 / 7 9	2	6	3 6 9 / 7	1
2 6 / 7	3 6 9 / 7	2 3 / 7 9	4 5 / 7 9	4 5 / 7 9	1	2 / 4 5 6	2 3 / 5 6 9 / 7	8
2 / 7 8	7 8 9	1	6	4 5 / 7 9	3	2 / 4 5	2 / 5 / 7 9	5 / 9
1	5 / 7 8 9	2 / 5 / 7 8 9	3 / 5 / 7	2 3 / 5 6 / 7	5 6 / 7 8	2 / 5 6 / 8	2 3 / 5 6 / 9	4
2 / 4 8	4 5 / 5 8 9	6	3 / 4 5	2 3 / 4 5	4 5 / 8	7	1	3 / 5 / 9
3	4 5 / 7 8	2 / 5 / 7 8	1	2 / 4 5 6 / 7	9	2 / 5 6 / 8	2 / 5 6	5 6
9	3 / 4 5 6	3 / 5	2	8	4 5 6	1	5 6	7
4 6 / 7	1	5 / 7	4 5 / 7 9	4 5 6 / 7 9	4 5 6 / 7	3	8	2
6 / 7 8	2	3 / 5 / 7 8	3 / 5 / 7	1	5 6 / 7	9	4	5 6

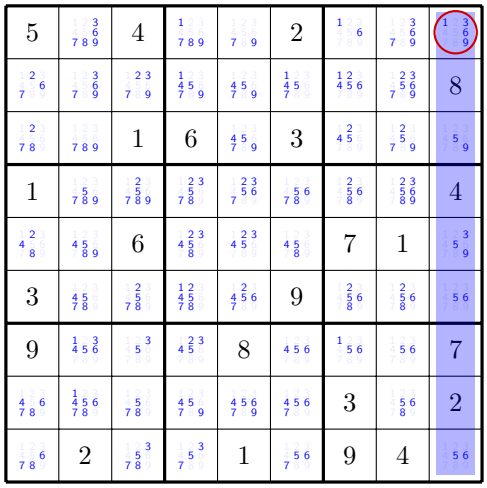
Deze strategie wordt meestal beschouwd als de meest triviale omdat de mogelijke kandidaten in een cel triviaal te bepalen zijn en naked singles hierbij direct zichtbaar worden. Bij het handmatig oplossen van een sudoku is het hiervoor

echter wel belangrijk dat er geen fouten worden gemaakt bij het noteren van de kandidaten, en dat deze na iedere stap waar nodig bijgewerkt worden.

2.1.2 Hidden Singles

Als een getal maar in 1 cel in een groep een kandidaat is, heet dit een hidden single. Deze cel bevat hierbij mogelijk ook nog andere kandidaten. Omdat elk getal precies, en dus ook minimaal, één keer moet voorkomen in elke groep is deze kandidaat altijd de oplossing in deze cel.

Hieronder is een voorbeeld te zien van een hidden single. In de blauwe kolom komt het getal 1 alleen voor als kandidaat in de omcirkelde cel. Het antwoord in deze cel moet dus 1 zijn.



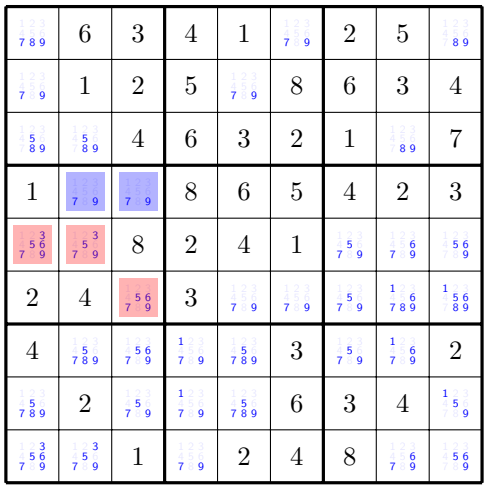
Deze strategie is erg handig omdat je hidden singles kunt vinden zonder eerst alle kandidaten uit te moeten schrijven.

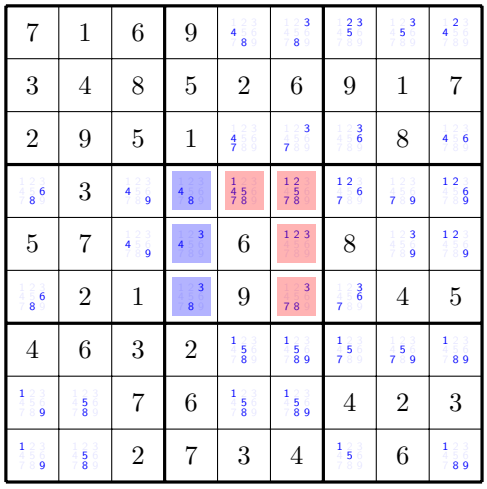
2.1.3 Naked Pairs/Triples

Een naked pair (of triple) kan in iedere groep voorkomen. Het betreft 2 (of 3) cellen in een groep die hetzelfde paar (of drietal) kandidaten bevat, en geen andere kandidaten. In het geval van een triple hoeven de cellen niet ieder alle drie de kandidaten te bevatten. Omdat deze twee (of drie) cellen als antwoord precies alle twee (of drie) van deze kandidaten moeten bevatten, kunnen we deze kandidaten elimineren uit de andere cellen in de groep.

Hieronder zijn twee voorbeelden te zien. De bovenste is een voorbeeld van een naked pair bestaande uit een 7 en 9 in de blauwe cellen. Deze cellen zitten samen in een blok, dus we kunnen deze kandidaten uit de rest van het blok, de rode cellen, elimineren. De onderste puzzel is een voorbeeld van een naked triple

waar de kandidaten 3, 4 en 8 in de blauwe cellen voorkomen. Zoals te zien bevat iedere cel individueel maar twee van deze kandidaten, maar bevatten ze samen het hele trio. Deze cellen zitten samen in het middelste blok, dus we kunnen deze kandidaten uit de andere cellen in het blok, de rode cellen, elimineren.





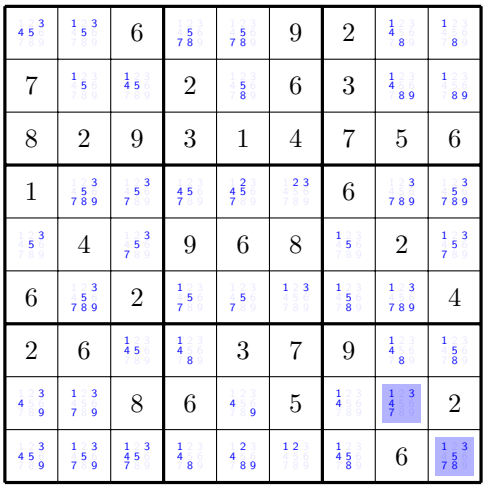
Bekijk voor meer uitleg en voorbeelden ook SudokuWiki, deze of deze video.

2.1.4 Hidden Pairs/Triples

Een hidden pair lijkt op een hidden single. Bij een hidden pair is er sprake van een tweetal kandidaten dat binnen een groep in precies een tweetal cellen als mogelijke kandidaat over is. Er zijn mogelijk ook andere kandidaten in deze twee cellen, maar deze kunnen we elimineren omdat we deze twee cellen dus

wel met het paar moeten vullen, we weten alleen nog niet precies welke van de twee kandidaten in de cellen het antwoord moeten zijn. Een hidden triple is op dezelfde manier wanneer een drietal kandidaten binnen een groep slechts in dezelfde drie cellen voorkomt. Net als bij een naked triple hoeft niet iedere kandidaat in iedere cel voor te komen. We kunnen dan weer alle andere kandidaten uit deze drie cellen elimineren.

Hieronder is een voorbeeld van een hidden pair te zien. De kandidaten 3 en 7 komen binnen het blok rechtsonder alleen in de blauwe cellen voor. Dit paar cellen moet dus wel 3 en 7 als antwoord hebben en we kunnen de andere kandidaten in deze cellen elimineren.



Kijk voor meer uitleg en voorbeelden ook op SudokuWiki.

2.1.5 Naked/Hidden Quads

In sommige gevallen is het mogelijk om dezelfde patronen als in naked of hidden triples te zien, maar dan met vier kandidaten in vier cellen. Deze komen echter vrij zeldzaam voor.
Een naked quad bestaat dus uit 4 cellen binnen een groep die ieder alleen kandidaten uit hetzelfde viertal bevatten. Deze cellen hoeven niet ieder het volledige viertal te bevatten, zolang ze samen maar het hele viertal bevatten. We kunnen deze kandidaten elimineren uit de rest van de groep. Een hidden quad bestaat uit een viertal kandidaten wat binnen een groep slechts in dezelfde 4 cellen voorkomt. We kunnen dan de andere kandidaten uit deze cellen elimineren.

Deze technieken zijn zelfs verder uit te breiden naar hogere aantallen, maar in een 9x9 sudoku bestaat iedere groep maar uit 9 cellen. Wanneer er binnen een groep sprake zou zijn van een hidden (of naked) quin (vijftal) is er in het complement automatisch sprake van een naked (respectievelijk hidden) quad (of

lager). In zulke gevallen is het vaak makkelijker om eerst naar de kleinere groep te kijken. In grotere sudoku's zijn deze mogelijk wel nuttig.

Bekijk voor meer uitleg en voorbeelden ook deze of deze pagina van SudokuWiki.

2.2 Intersection Removal

2.2.1 Pointing Pairs/Triples

Binnen een blok kan het gebeuren dat een kandidaat precies in 2 (of 3) cellen voorkomt. Als deze dan ook nog eens op een lijn liggen zijn ze dus precies bevat in de doorsnede van twee groepen, het blok en een rij of kolom. Als dit gebeurt met twee cellen heet dit een pointing pair, bij drie cellen heet het een pointing triple. We weten dat binnen het blok deze kandidaat wel de oplossing in een van de cellen moet zijn, dus deze kan niet de oplossing zijn in de rest van de andere groep. We kunnen de kandidaat dus elimineren uit de cellen van de rij of kolom die buiten het blok vallen.

Hieronder is een voorbeeld te zien van een pointing pair. Binnen het blok komt 2 als kandidaat alleen voor in de blauw gemarkeerde cellen. In een van deze cellen moet 2 dus wel de oplossing zijn, waardoor we het als kandidaat uit de rest van de kolom kunnen elimineren.

5	3	4	8	9	2	6	3	1
2 6	3 6	9	4 5	4 5	1	2 4 5	2 3 5	8
2		1	6	4 5	3	2 4 5	2 5	9
1	5	2 5	7 5	3	6	2 5	9	4
2	9	6	4 5	2 4 5	8	7	1	3
3	4 5	2 5	1	2 4 5	9	2 5	2 5 6	5 6
9	4 5 6	3	2	8	4 5	1	5 6	7
4	1	5	9	6	4 5	3	8	2
6	2	5	3	1	5	9	4	5 6

Bekijk voor meer uitleg en voorbeelden ook SudokuWiki of deze video.

2.2.2 Claiming Pairs/Triples

Een claiming pair (of triple) is het tegengestelde van een pointing pair. We kijken naar een kandidaat die binnen een rij of kolom maar in twee (of drie) cellen voorkomt. Als deze twee (of drie) cellen ook binnen hetzelfde blok liggen,

noemen we dit een claiming pair (respectievelijk triple) en kunnen we deze kandidaat elimineren uit de rest van het blok.

Bekijk voor meer uitleg en voorbeelden ook SudokuWiki of deze video.

2.3 Fish

2.3.1 X-Wing

Een X-Wing kan plaatsvinden in de rijen of in de kolommen. Een X-wing in de rijen is wanneer een kandidaat in vier cellen voorkomt die een rechthoek vormen, en verder niet voorkomt in de rijen die deze cellen bevatten. Wanneer dit het geval is moet in ieder van deze rijen één van deze twee cellen de oplossing zijn, natuurlijk kunnen deze niet recht boven elkaar staan, dus is de kandidaat altijd de oplossing van de twee cellen die een diagonaal van de rechthoek vormen (vandaar de X). Dit betekent dat in beide kolommen die de cellen bevatten deze kandidaat de oplossing is van één van de twee cellen, en dus niet de oplossing kan zijn van andere cellen in deze kolommen. De kandidaat wordt zo dus geëlimineerd uit de andere cellen in deze kolommen.
Een X-wing in de kolommen is vergelijkbaar. In plaats van dat de kandidaat niet voorkomt in de rijen mag deze niet voorkomen in de kolommen en in plaats van de kandidaat te elimineren uit de overige cellen in de kolommen wordt deze geëlimineerd uit de overige cellen in de rijen. De logica is dus volledig hetzelfde, maar 90 graden gedraaid.

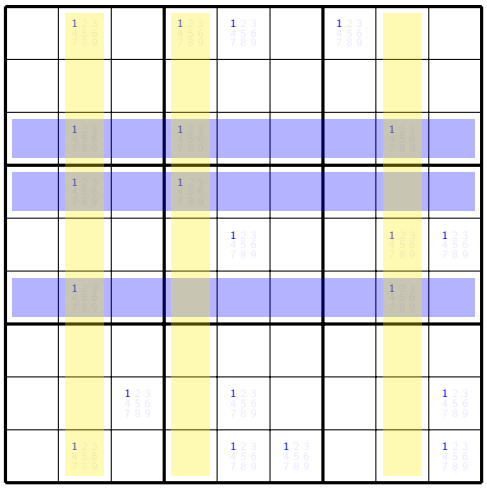
Hieronder is een voorbeeld te zien van een X-wing in de kolommen. De gekleurde kolommen en rijen vormen een rechthoek, de cellen in de hoeken van de rechthoek bevatten de kandidaat 9 die verder niet voorkomt in de blauwe kolommen. Hierdoor kun je de 9 dus elimineren uit de andere cellen in de gele rijen.

	1		8		4		2	
	6			5	2	1		
	4	2		1				
1	3	4	7	2	8		6	
9		6	1	4	5	2		
2		5	6	9	3			1
6	2	1	4	8	9			7
4	9	7	5	3	1	6	8	2
3	5	8	2				1	

Kijk voor meer uitleg en voorbeelden ook op SudokuWiki of Hodoku.

2.3.2 Swordfish

Een swordfish kan net als een X-Wing plaatsvinden in de rijen of kolommen. Een swordfish in de rijen is wanneer in 3 rijen een kandidaat in iedere rij in (hoogstens) drie cellen voorkomt, en deze cellen in dezelfde 3 kolommen bevat zijn. Omdat de kandidaat in iedere rij in één van de drie cellen de oplossing moet zijn, en deze niet in dezelfde kolom mag staan, moet de kandidaat binnen de 9 cellen in elke kolom één keer de oplossing zijn. We kunnen de kandidaat dus uit de andere cellen van de kolommen elimineren. Hieronder een abstracte illustratie van een swordfish in de rijen. In de blauwe rijen komt 1 alleen als kandidaat voor in de gele kolommen. Hierdoor kunnen we 1 elimineren uit de rest van deze kolommen.



Bekijk voor meer uitleg en voorbeelden ook SudokuWiki, Hodoku of deze video.

2.3.3 Jellyfish

Het soort patroon wat voorkomt bij X-wing en swordfish kan ook meer algemeen voorkomen in een hoger aantal rijen/kolommen. Dit patroon in 4 rijen of kolommen heet ook wel een Jellyfish. Ook grotere aantallen zijn mogelijk, maar komen weinig voor.

Kijk voor meer uitleg en voorbeelden ook op SudokuWiki of Hodoku.

2.4 Wings

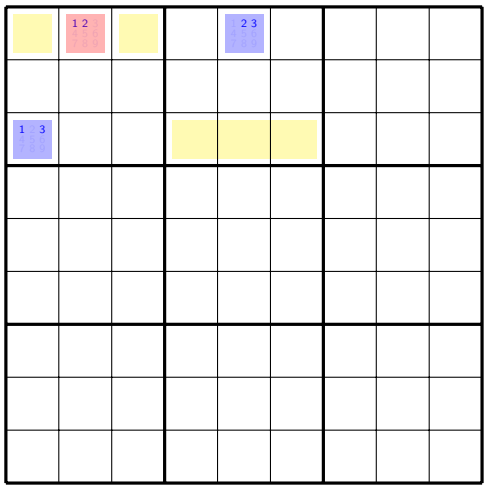
Wings zijn een groep van strategieën die door vergelijkbare logica kandidaten elimineren. Ik zal mij hier beperken tot één variant, maar meer informatie

over de andere varianten is te vinden op Hodoku, deze en deze pagina van SudokuWiki. Wings zijn ook op te vatten als een variant van een chain, deze behandel ik in de volgende paragraaf kort.

2.4.1 Y-Wing

Een Y-Wing, ook wel XY-Wing genoemd, komt voor wanneer 3 kandidaten paarsgewijs (naakt) voorkomen in 3 cellen waarvan één cel beide andere cellen 'ziet'. Deze centrale cel wordt de pivot of hinge genoemd, de andere twee heten pincers of wings. De kandidaat die niet in de pivot voorkomt, maar wel in beide pincers voorkomt, moet wel de oplossing in één van de pincers zijn, hierdoor kunnen we deze elimineren uit alle cellen die door beide pincers 'gezien' worden.

Hieronder is een abstract voorbeeld te zien van een Y-Wing met kandidaten 1,2,3. 1 en 2 zijn de kandidaten in de rode pivot, 1 en 3, en 2 en 3 zijn de kandidaten in de blauwe pincers. Als de oplossing in de linker pincer 1 zou zijn (en dus niet 3), zou de oplossing in de pivot 2 moeten zijn, en de oplossing in de rechter pivot dus 3 zijn. Als de oplossing in de rechter pincer 2 zou zijn (en dus niet 3), zou de oplossing in de pivot 1 moeten zijn, en dus de oplossing in de linker pivot 3. We zien dus inderdaad dat 3 hoe dan ook in één van de pincers de oplossing moet zijn, en we deze dus kunnen elimineren uit de gele cellen.



Kijk voor meer uitleg en voorbeelden ook op SudokuWiki.

2.5 Chains

Chains zijn een groep van strategieën waarin gebruik wordt gemaakt van bepaalde verbanden tussen cellen. Deze vormen dan samen een langere 'chain' waar soms weer bepaalde gevolgen uit afgeleid kunnen worden. Ik zal mij hier

beperken tot één variant, maar meer informatie over de andere varianten is te vinden op Hodoku, deze en deze pagina van SudokuWiki.

2.5.1 Single's Chains/Simple Coloring

Neem een vaste kandidaat. Iedere keer dat de kandidaat in een groep in precies twee cellen voor komt link je deze (dit heet ook wel een bi-location link of conjugate pair). Een samenhangende verzameling van zulke cellen en links kun je opvatten als een graaf die je met twee kleuren kunt kleuren. Als je dit doet (bv met groen en rood) weet je dat voor een van de twee kleuren de gekozen kandidaat de oplossing is in alle cellen met die kleur.
Als twee cellen in dezelfde groep dezelfde kleur hebben, spreekt deze kleur zichzelf dus als het ware tegen, en moet de andere kleur dus wel 'het antwoord' zijn. Als een cel (buiten de graaf) met de gekozen kandidaat gezien wordt door cellen van beide kleuren, kan de kandidaat niet het antwoord in deze cel zijn, en dus geëlimineerd worden (lijkt op Y-Wing).

Hieronder is een voorbeeld te zien van een simple coloring met kandidaat 2. Hierin zijn een aantal gelinkte cellen afwisselend geel en blauw gemarkeerd. De rood gemarkeerde cel wordt zowel door een gele als een blauwe cel gezien. We weten dat 2 wel het antwoord moet zijn in één van deze cellen, dus kunnen we 2 als kandidaat elimineren uit de rode cel.

3	5	2 7 8	2 8	1	9	6	2 7 8	4
1	2 8	9	7	4	6	5	2 8	3
6	2 7 8	4	3	2 8	5	1	2 7 8 9	2 8 9
7 8	1	5	4	6	2	7 8 9	3	8 9
4	3	2 8	1	9	7	2 8	6	5
2 7	9	6	5	3	8	2 7	4	1
2 8	6	3	9	2 8	1	4	5	7
9	2 7 8	2 7 8	2 6 8	5	4	3	1	2 6 8
5	4	1	2 6 8	7	3	8 9	2 8 9	2 6 8 9

Kijk voor meer uitleg en voorbeelden ook op SudokuWiki of Hodoku.

2.6 Uniqueness Strategies

Er bestaan ook een aantal strategieën die uitgaan van een unieke oplossing van de puzzel, en op basis van deze aanname redeneren. Wanneer deze toegepast

worden op een puzzel met meerdere mogelijke oplossingen kan dus 'fout' gaan. Een aantal van deze strategieën zijn:

- Unique Rectangles
- Hidden Rectangles
- Avoidable Rectangles
- BUG

Deze strategieën, en de patronen waarop ze gebaseerd zijn, zal ik kort behandelen in het komende hoofdstuk.

3 Uniciteit Oplossing

Zoals gezegd willen we dat een sudoku puzzel altijd precies één unieke oplossing heeft. De meeste uitgevers letten er zeker op dat hun puzzels hieraan voldoen, maar soms worden er ook sudoku's gepubliceerd met meerdere oplossingen. Een voorbeeld hiervan is hieronder te zien, deze komt uit het Haarlems Dagblad van 8 december 2005 en heeft 44 verschillende oplossingen. Een aantal andere zulke sudoku's zijn hier verzameld.

3					1		7	
4					2		3	
	1	8	6	3		2		4
9					3		8	
	4	1	8	9		3		2
6					4		9	
1					5		2	
2					8		1	
	3	9	2	1		5		7

Daarnaast zou het natuurlijk ook kunnen gebeuren dat een puzzel geen oplossingen heeft. Deze worden (bijna) nooit gepubliceerd, maar zouden wel kunnen ontstaan wanneer een puzzel gedeeltelijk is ingevuld met een fout antwoord. Een voorbeeld van zo'n sudoku is hieronder te zien. Er is geen enkel nummer dat in de blauwe cel ingevuld kan worden, dus de puzzel heeft geen oplossing.

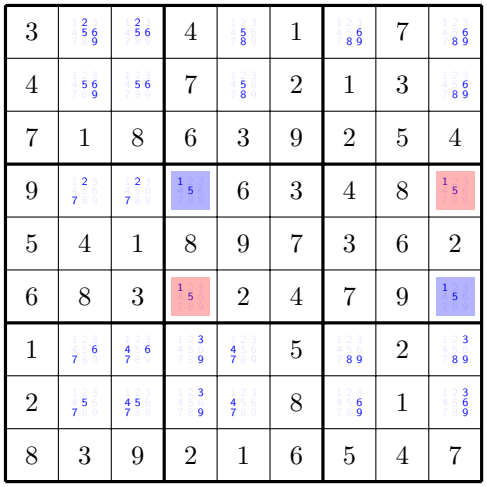
	1							
	2							
	3							
	4							
1		3	4	5	6	7	8	9
	6							
	7							
	8							
	9							

Dit leidt vanzelfsprekend tot de vraag, hoe kunnen we bepalen of een sudoku een uniek bepaalde oplossing heeft? Op deze vraag heb ik helaas nog geen bevredigend antwoord kunnen vinden. De meest gebruikte manier om het aantal oplossingen van een sudoku te bepalen lijkt te zijn om simpelweg door middel van een brute force algoritme alle mogelijke oplossingen te bepalen. Wel heb ik een aantal kenmerken gevonden die ons iets over het aantal oplossingen van een sudoku kunnen vertellen.

De eenvoudigste hiervan betreffen sudoku's zonder oplossingen. Wanneer een symbool 2 keer voorkomt in een rij, kolom of blok is dit in tegenspraak met de restrictie die moet gelden, en kan er dus geen oplossing bestaan. Daarnaast kan het gebeuren dat er in een cel geen kandidaten meer over zijn, zoals in het bovenstaande voorbeeld. Ook dan kan er vanzelfsprekend geen oplossing bestaan. Helaas zijn deze kenmerken niet altijd meteen zichtbaar, maar moet een foute sudoku soms eerst verder opgelost worden voordat deze hiermee herkend kan worden.

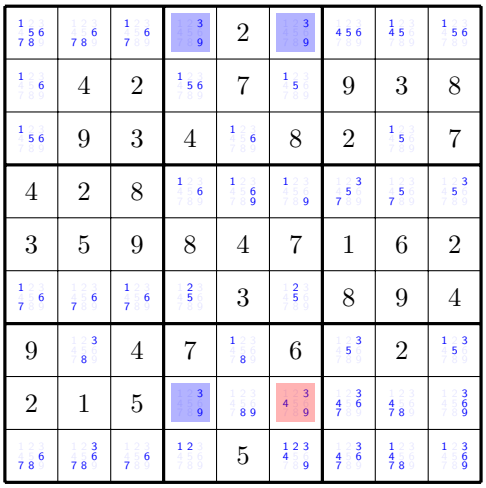
Er zijn twee kenmerken waaraan een sudoku puzzel minimaal moet voldoen om te garanderen dat deze niet meerdere oplossingen heeft. In een sudoku puzzel moeten minimaal 17 hints gegeven worden, en minimaal 8 van de 9 cijfers moeten voorkomen als hint. Dit laatste geldt ook algemener, minimaal $n - 1$ van de n symbolen moeten voorkomen als hint.
Deze eerste eis is bewezen met een computerbewijs waarin voor ieder mogelijk volledig ingevuld rooster werd bekeken of hieruit 16 hints konden worden geselecteerd zodat de resulterende puzzel het gevraagde rooster als unieke oplossing had. Zulke puzzels met 16 hints bleken niet te bestaan, er zijn echter zeker wel een aardig aantal sudoku puzzels met 17 hints [4].
Deze tweede eis kan vrij simpel beredeneerd worden. Als een puzzel hier niet aan voldoet zijn er (minimaal) twee symbolen die niet als hint voorkomen. Wanneer we een mogelijke oplossing voor deze puzzel gevonden hebben, kunnen we hierin alle instanties van deze twee symbolen omwisselen. Vanzelfsprekend verandert dit de gegeven hints in het rooster niet, en voldoet het nieuwe rooster nog steeds aan de nodige restrictie. Dit is dus ook een oplossing van de puzzel, dus we zien dat deze er meer dan één heeft.

Naast deze twee eigenschappen zijn er ook twee patronen in de kandidaten die we tijdens het oplossen kunnen tegenkomen die wijzen op het hebben van meerdere oplossingen. Deze patronen zijn bekend als een 'Deadly pattern' en een 'Bivalue Universal Grave', ook wel bekend als BUG.
Een deadly pattern is wanneer er in vier cellen die een rechthoek vormen en die paarsgewijs in dezelfde blokken zitten, alleen nog dezelfde twee kandidaten voorkomen. Een voorbeeld hiervan is hieronder te zien, met de kandidaten 1 en 5 in de gekleurde cellen. We zien dat dit tot twee mogelijkheden leidt, en dus ook tot (minimaal) twee oplossingen; 1 is het antwoord in de blauwe cellen en 5 is het antwoord in de rode cellen, of 5 is het antwoord in de blauwe cellen en 1 is het antwoord in de rode cellen.



Een BUG is een generalisatie van dit patroon waarin alle resterende cellen van de sudoku precies twee kandidaten bevatten en iedere kandidaat precies twee keer voorkomt in elke groep. Ook dan geldt dat de sudoku meerdere oplossingen heeft [2].

De strategieën die ik in Paragraaf 2.6 noemde, zijn gebaseerd op deze patronen. Deze zijn erg gecompliceerd, dus ik zal hier enkel het idee erachter schetsen. Voor een uitgebreidere uitleg inclusief voorbeelden, zie Hodoku.



Zoals gezegd nemen deze strategieën aan dat de puzzel een unieke oplossing heeft, en deze patronen dus niet mogen voorkomen. Er zijn echter situaties waarin zo'n patroon 'dreigt' te ontstaan, maar waarin bijvoorbeeld één of twee

cellen nog net een extra kandidaat bevatten. Door in zo'n geval specifieke kandidaten te elimineren, kunnen we dit patroon verstoren en dus voorkomen. In het voorbeeld hierboven dreigt een deadly pattern te ontstaan in de gekleurde cellen. Wanneer 4 hier niet het antwoord in de rode cel zou zijn, blijft het deadly pattern over. Dit mag niet, dus 4 moet wel het antwoord in deze cel zijn en de 3 en 9 kunnen hier geëlimineerd worden.

Het hebben van een uniek bepaalde oplossing betekent nog niet dat we deze met de bekende strategieën kunnen vinden. Er bestaan zeker ook puzzels met een unieke oplossing die 'onoplosbaar' zijn. Een bekend voorbeeld hiervan is onderstaande puzzel, gemaakt door Arto Inkala [7].

8								
		3	6					
	7			9		2		
	5				7			
				4	5	7		
			1				3	
		1					6	8
		8	5				1	
	9					4		

4 Moeilijkheidsgraad

Iedereen die een aantal sudoku puzzels heeft opgelost zal hebben gemerkt dat er een verschil zit in hoe moeilijk deze zijn. Vaak geven uitgevers dit aan met een term als 'makkelijk', 'medium', 'moeilijk', 'expert' of met een aantal sterren. Dit leidt vanzelfsprekend tot de vraag; wat maakt een sudoku moeilijk en hoe wordt de moeilijkheidsgraad bepaald?

Er zijn veel factoren die invloed kunnen hebben op de moeilijkheidsgraad van een sudoku. Het aantal hints, en het patroon waarin deze voorkomen kunnen hierin meespelen. Dit lijkt echter geen goede maatstaf te zijn omdat bijvoorbeeld een puzzel met 20 hints veel makkelijker kan zijn dan een puzzel met 28 hints, en twee puzzels met hints in dezelfde positie kunnen ook heel verschillend zijn [5].

Natuurlijk is het ook belangrijk om te overwegen dat 'moeilijkheid' een erg subjectieve, persoonlijke en menselijke ervaring is. Wat sommigen verwarrend vinden, vinden anderen logisch, hierin spelen eerdere ervaringen mogelijk ook mee. Ieder mens heeft zijn eigen aanpak, sterktes en voorkeuren. Daarnaast is er natuurlijk ook een enorm verschil tussen wat moeilijk is voor en mens en wat moeilijk is voor een computer, een computer kan vrij simpelweg 100 dingen uitproberen, maar dit is voor een mens erg veel werk. Aan de andere kant kan een computer een erg complex algoritme toepassen om iets te vinden, terwijl een mens dit uit ervaring of geluk al meteen zag.

4.1 Voorbeelden

Uitgevers lijken wat terughoudend om informatie te onthullen over hoe zij precies de moeilijkheidsgraden van hun sudoku's bepalen. Dit is vrij logisch omdat zij hier natuurlijk veel tijd en geld in geïnvesteerd hebben. Online zijn een paar programma's te vinden die iets meer onthullen over hun eigen aanpak, deze zullen we eerst verkennen. Het is hierbij belangrijk om op te merken dat niet alle programma's dezelfde namen voor de strategieën gebruiken.

4.1.1 SudokuWiki, Andrew Stuart

Stuart [6] benoemt twee metrieken om de moeilijkheidsgraad te bepalen, 'frequency of opportunity' heeft te maken met hoeveel antwoorden er op ieder moment te vinden zijn, de tweede heeft te maken me de strategieën die nodig zijn om tot een oplossing te komen. Voor dit tweede is het nodig om een ordening in de moeilijkheid van strategieën aan te brengen. Hij geeft aan hiervoor van een grote verzameling puzzels statistisch te hebben bekeken hoe vaak een strategie gebruikt kan worden en hoe veel strategieën een 'bottleneck', weinig mogelijkheden voor antwoorden, kunnen openen. Hij geeft aan hiermee de strategieën te kunnen ordenen op hoe moeilijk het is om ze te herkennen, hoe vaak ze nodig zijn en hoe veel kandidaten ze kunnen elimineren. Hierbij blijft natuurlijk

sprake van enige subjectiviteit.

Om van een specifieke puzzel de moeilijkheidsgraad te bepalen lost zijn programma deze eerst op door strategieën in een gespecificeerde volgorde toe te passen. Hierbij houdt het van iedere strategie bij hoeveel kandidaten ermee zijn verwijderd en hoeveel antwoorden het heeft opgeleverd (in totaal). De volgorde van deze strategieën is bepaald op basis van complexiteit, en hierbij hangt er per strategie een gewicht aan geëlimineerde kandidaten en antwoorden. Dit levert voor de gehele puzzel een totaal aantal punten, wat met een vaste factor vermenigvuldigd wordt om de 'score' te bepalen. Het programma houdt hiernaast ook bij hoeveel 'rondes' er nodig waren en berekend hiermee het gemiddelde aantal antwoorden per ronde.

Stuart verdeelt sudoku's in 6 categorieën: kids, gentle, moderate, tough, diabolical (ook wel very hard genoemd) en extreme. Behalve alleen de score stelt hij ook bepaalde eisen aan sudoku's van lagere niveaus, met name dat er alleen specifieke strategieën nodig mogen zijn.

Een lijst van de specifieke strategieën, geordend op de volgorde waarin deze worden toegepast, is terug te vinden in de Appendix 8.2.1. Stuarts programma is hier te vinden.

4.1.2 Hodoku, Bernhard Hobiger

Hobiger [2] is waarschijnlijk het meest transparant in zijn aanpak. Deze is enigszins vergelijkbaar aan die van Stuart in dat alle strategieën (subjectief) zijn geordend op moeilijkheid, en op basis hiervan een niveau (een 'level'; easy, medium, hard, unfair of extreme) en een score toegewezen hebben. Deze zijn in het programma te configureren. Het programma zoekt vervolgens een oplossing van de sudoku door de strategieën op volgorde van 'moeilijkheid' toe te passen, deze volgorde is ook te configureren. Het maximale niveau van de gebruikte strategieën geeft vervolgens een ondergrens van het niveau van de sudoku. Daarnaast bepaalt het de som van de scores van de strategieën. Ieder niveau is een interval toegewezen wat ook weer te configureren is. Het interval waarin de totale score valt bepaalt van welk niveau de sudoku is, mits dit gelijk of hoger is aan het niveau van de moeilijkste strategie. Is dit niet het geval, dan wordt het niveau van de moeilijkste strategie aangehouden.

Een lijst van de specifieke strategieën, inclusief level en score, geordend op de volgorde waarin Hodoku deze toepast, is terug te vinden in de Appendix 8.2.2.

4.1.3 Uitgevers

Om te achterhalen hoe traditionele uitgevers de moeilijkheidsgraad van hun sudoku's benaderen, heb ik geprobeerd een aantal van hen hierover te benaderen met weinig succes. Denksport, een onderdeel van de Keesing media groep,

reageerde als enige op mijn verzoek met onderstaand bericht. Voor context is het goed om te weten dat zij puzzels op een schaal van 1 tot 15 sterren uitgeven.

> "Hi Robin,
>
> Een interessante maar ingewikkelde vraag!
> Voor het maken van sudoku's maken we gebruik van zoals wij dat noemen 'recepten', een ander woord voor het gebruik van strategieën.
> Hoe meer sterren, hoe ingewikkelder de strategieën.
> Het is natuurlijk wel zo dat we onze recepten niet aan derden willen geven, maar er zijn een heleboel manieren om meer duidelijkheid te krijgen over de gebruikte strategieën. Deze zijn voor alle sudoku's van alle puzzelmakers namelijk hetzelfde, alleen de mix van de strategieën kan variëren.
>
> Een site die veel gebruikt wordt en die ik zeker kan aanraden is deze:
> https://sudowiki.org
> Bij de sudokusolver op deze website staat rechts een heel overzicht van verschillende technieken die gebruikt worden plus voorbeelden. Dit zijn in essentie ook de technieken die wij gebruiken. De mix van de verschillende technieken bepaalt de moeilijkheidsgraad.
> Zo gebruiken wij bijvoorbeeld de X en Y techniek alleen bij boekjes hoger dan 5 sterren.
>
> Een andere bruikbare site is:
> https://hodoku.sourceforge.net
>
> Verder bepaalt het aantal weggevers tot op zekere hoogte de moeilijkheidsgraad, bij de makkelijke sudoku's worden eigenlijk meer weggevers geplaatst dan in essentie nodig zijn om de puzzel op te kunnen lossen. Het minimumaantal weggevers is 17. Bij 16 of minder weggevers is een sudoku nooit op te lossen!
>
> Mocht je nog verdere vragen hebben kun je me natuurlijk altijd mailen.
>
> Met vriendelijke groet,"

Hiermee lijkt het er dus op dat de uitgevers zeer vergelijkbare methoden gebruiken aan de eerder besproken programma's. Ook zij maken gebruik van strategieën, en bepaalde strategieën komen alleen voor in sudoku's vanaf een bepaald niveau. Dit wordt ook door een aantal experts beaamd [3] [5].

4.2 Ordening strategieën

Zoals gezegd gebruiken uitgevers meestal een programma dat nagaat welke strategieën nodig zijn om een sudoku op te lossen. Deze strategieën hebben ze hiervoor geordend op moeilijkheid. Op basis van de moeilijkheid van de nodige strategieën en een aantal andere factoren, zoals de effectgrootte (het aantal opgeleverde antwoorden of geëlimineerde kandidaten) en de mogelijke verschillende combinaties van strategieën, kan aan een puzzel een score worden toegewezen waarmee deze op moeilijkheidsgraad kan worden ingedeeld.

De simpelste manier om een ordening van strategieën te bepalen is door vooral vanuit menselijk oogpunt naar 'moeilijkheid' te kijken. Door naar eigen gevoel, intuïtie en ervaring te kijken, kunnen de strategieën op moeilijkheid gerangschikt worden. Deze rangschikking is vooral gebaseerd op eigen meningen en is daarmee natuurlijk persoonlijk en subjectief. Wat de ene persoon logisch en intuïtief vind, kan een ander natuurlijk veel moeite mee hebben. Als een persoon bijzonder geoefend is in een specifieke strategie, kunnen zij gevallen waarin deze is toe te passen waarschijnlijk snel herkennen en zullen zij deze strategie dus makkelijker vinden dan een ander persoon. Een 100% algemene ordening kan op deze manier dus niet verkregen worden, maar toch zijn er wel breed geaccepteerde ordeningen zoals die van SudokuWiki. Omdat het doel van het bepalen van de moeilijkheidsgraad van een sudoku is om een indicatie aan het algemene publiek te geven van wat voor een ervaring zij kunnen verwachten van de puzzel, vind ik dit de meest logische manier van ordenen.

Het is natuurlijk ook mogelijk om op andere manieren naar de complexiteit van de strategieën te kijken, bijvoorbeeld hoeveel informatie je nodig hebt om een strategie toe te passen. Dit is dan echter ook weer afhankelijk van hoe je deze informatie in eerste instantie structureert.
Dit zou je kunnen relateren aan een maat voor de complexiteit van de strategie wanneer deze geïmplementeerd is als een algoritme. Een aantal strategieën zijn al op een meer abstracte en formele manier omschreven, bijvoorbeeld in dit paper van Kevin Gromley.
Ook kan er statistisch worden gekeken naar het gebruik en nut van strategieën. Hiervoor zouden deze tegen een (representatieve) verzameling puzzels getest kunnen worden om te meten op hoeveel puzzels deze van toepassing zijn en wat de effectgrootte ervan is. Daarbij zou ook onderzocht kunnen worden welk effect het toevoegen van een strategie aan een bepaalde deelverzameling van strategieën door te kijken hoeveel meer sudoku's er met de nieuwe verzameling strategieën opgelost kunnen worden in vergelijking tot de oude. De strategie die op basis van deze maatstaven het meeste 'nut' heeft, levert dus de meeste waarde op om te leren. De verhouding tussen deze waarde en de moeite die het kost om de strategie te leren en toe te passen (wat afhankelijk is van de complexiteit) zou dan gebruikt kunnen worden om de strategieën te ordenen.

Zoals gezegd geef ik de voorkeur aan de eerste manier om strategieën te ordenen

omdat ik denk dat dit een beter idee geeft over hoe een mens het oplossen van een puzzel zou ervaren. Ter illustratie zal ik hier een aantal strategieën direct met elkaar vergelijken, en mijn eigen voorkeuren hierin beargumenteren.

4.2.1 Naked vs Hidden Singles

Zie Paragraaf 2.1.1 en 2.1.2 Voor een uitleg over naked en hidden singles respectievelijk.

In een puzzel waarin kandidaten genoteerd zijn, zijn naked singles natuurlijk in een oogopslag herkenbaar. Ik vermoed dat de meesten dit hierom als makkelijker beschouwen. Wanneer ik zelf een puzzel oplos heb ik echter de voorkeur om niet meteen de kandidaten te noteren, op papier maken al deze markeringen de puzzel wat rommelig en daardoor onoverzichtelijk. Daarnaast vind ik dit eerlijk gezegd ook erg saai werk. Hierom loop ik liever eerst alle cijfers één voor één af en kijk in binnen ieder blok op welke posities deze ingevuld kan worden, ik zoek dus hidden singles binnen de blokken.

Het is natuurlijk ook mogelijk om naked singles te vinden zonder kandidaten te noteren door de cellen één voor één langs te gaan en te bekijken welke cijfers hier ingevuld zouden kunnen worden. Dit is dus echter precies het werk wat ik saai vindt, en als je op deze manier alle kandidaten in iedere cel gaat bepalen, zou ik deze liever toch meteen noteren omdat je dit werk anders later dubbel moet doen.

SudokuWiki lijkt naked singles als triviaal te behandelen, en past deze dan ook als eerste toe, direct gevolgd door hidden singles. Hodoku maakt naast naked singles ook gebruik van 'Full House', een specifieke variant van naked singles. Hodoku noemt full house, naked singles en hidden singles allen 'easy', full house en naked singles hebben een score van 4, hidden singles hebben een score van 14. Hodoku past eerst full house, dan naked singles en daarna hidden singles toe als eerste drie strategieën.

4.2.2 Y-Wing vs Simple Coloring

Zie Paragraaf 2.4.1 en 2.5.1 Voor een uitleg over Y-wing en simple coloring respectievelijk.

Het patroon van de Y-wing vindt altijd plaats in 3 cellen, daartegenover betreft het patroon van een simple coloring een onbepaald aantal cellen, maar meestal meer dan 3. Een Y-wing betreft altijd drie kandidaten, maar een simple coloring gaat maar over 1 kandidaat. Zelf vind ik het makkelijker om naar patronen te zoeken voor 1 kandidaat tegelijk, dus hierin heeft de simple coloring mijn voorkeur, vind ik deze 'makkelijker', maar het kan goed zijn dat een ander het juist makkelijker vindt om maar naar een paar cellen te hoeven kijken. Daarbij komt

wel kijken dat, zeker in een papieren sudoku, het moeilijker is om een simple coloring te noteren.
Voor een Y-wing is het vereist dat de kandidaten altijd 'naakt' zijn, terwijl dit bij een simple coloring niet hoeft. In een simple coloring moeten de kandidaten paarsgewijs de enige zijn in een groep, maar dit hoeft voor een Y-wing weer niet.

Een Y-wing kan tot 5 kandidaten elimineren, een simple coloring kan meer kandidaten elimineren en ook meerdere cellen invullen in het geval waarin twee cellen van dezelfde kleur elkaar zien, alhoewel dit relatief weinig voorkomt.
De redenatie achter simple colorings is iets complexer dan die achter Y-wings. Ik schat echter in dat dit voor de meeste mensen geen hindernis hoeft te zijn omdat het zeker voor beide strategieën mogelijk is om het patroon en de gevolgen uit je hoofd te leren, en je deze dus kunt toepassen, ook wanneer je de redenatie mogelijk niet volgt, maar dit verschilt natuurlijk ook per persoon.

Over het algemeen vind ik om deze redenen een simple coloring iets makkelijker om te gebruiken dan een Y-wing. SudokuWiki noemt beide strategieën 'tough', maar past simple coloring toe vóór Y-wing. Hodoku noemt beide strategieën 'hard' en geeft simple coloring een score van 150 en Y-wing een score van 160. Hodoku past echter standaard eerst de Y-wing toe.

4.3 Diagram

Het invullen van een sudoku kan voorgesteld worden als een soort pad door een graaf waar de oorspronkelijke puzzel het beginpunt is en de oplossing het eindpunt. De tussenstappen zijn dan de toestanden waar er telkens een vakje meer is ingevuld. Hierin zitten eventueel ook fouten die leiden tot doodlopende paden.
Het is natuurlijk mogelijk om een fout te maken door een getal in een cel in te vullen waarin deze geen kandidaat was, dit levert over het algemeen een directe tegenspraak op met de rest van de puzzel. Wanneer een (zorgvuldig) mens een sudoku maakt zullen ze dit soort fouten echter nooit maken. Het is dus logisch om dit soort fouten niet mee te nemen in deze graaf. Ook kan een mens, wanneer ze kandidaten bijhouden, naked singles meteen herkennen, en zal deze dus meteen invullen. Het is dus logisch dat we dit in onze graaf ook doen. Omdat hidden singles van een vergelijkbare moeilijkheid zijn, kunnen we deze ook meteen in onze graaf invullen. Deze aannames helpen om de omvang van de graaf enigszins te beperken omdat we hiermee een deel van de fouten voorkomen.
Hierbij zouden we ervoor kunnen kiezen om het invullen van naked en hidden singles niet te zien als een stap in de graaf, maar dit te zien als dezelfde toestand en dus hetzelfde punt in de graaf. Dit omdat het invullen van naked en hidden singles altijd een recht pad zonder zijtakken is, en we op deze manier onderscheid tussen het triviale invullen van naked en hidden singles, en het niet-triviale 'gokken' kunnen maken.

Ik verwacht dat bepaalde aspecten van deze graaf verband houden met de moeilijkheidsgraad van een sudoku. Zo verwacht ik bijvoorbeeld dat het aantal paden naar de oplossing in verhouding met het aantal doodlopende paden iets over de moeilijkheid zegt. Ook verwacht ik dat de diepte van deze graaf, waarmee ik de lengte van de succesvolle paden bedoel, hier iets over zegt.

Wanneer we deze graaf proberen te tekenen lopen we er echter al snel tegenaan dat deze erg groot is. Zo zijn in de onderstaande sudoku geen naked of hidden singles aanwezig. Wanneer we in de graaf alle mogelijke hierop volgende toestanden willen weergeven, moeten we deze dus voor iedere overgebleven kandidaat de toestand bepalen wanneer we deze zouden invullen. Dit levert dus voor één stap, vanaf één toestand al 68 nieuwe toestanden op. Het is dus niet realistisch om dit handmatig na te gaan, maar mogelijk kunnen we wel een programma gebruiken wat deze graaf gaat verkennen.

3	5	2 7 8	2 8	1	9	6	2 7 8	4
1	2 8	9	7	4	6	5	2 8	3
6	2 7 8	4	3	2 8	5	1	2 7 8 9	2 8 9
1 2 3 7 8	1	5	4	6	2	1 2 3 7 8 9	3	1 2 3 8 9
4	3	2 8	1	9	7	2 8	6	5
2 7	9	6	5	3	8	2 7	4	1
2 8	6	3	9	2 8	1	4	5	7
9	2 7 8	2 7 8	2 6 8	5	4	3	1	2 6 8
5	4	1	2 6 8	7	3	2 8 9	2 8 9	2 6 8 9

4.4 Programma

Dit programma moet natuurlijk eerst alle mogelijke kandidaten die geen directe tegenspraak veroorzaken bepalen. Daarnaast vult het alle antwoorden die we als triviaal beschouwen in. Zoals gezegd zou ik over het algemeen naked en hidden singles als triviaal willen beschouwen, maar het is wenselijk om de optie te hebben om te kiezen of het programma naked singles, hidden singles, allebei of geen een als triviaal moet beschouwen om het effect hiervan te kunnen bekijken. Dit telt het programma niet als het zetten van een stap.

Wanneer er geen triviale stappen zijn zal het programma de sudoku verder in moeten vullen door een kandidaat te kiezen en deze in te vullen. Hierbij kunnen we bepalen hoe het programma deze kandidaat moet uitkiezen en ook hierbij is het wenselijk om meerdere opties te hebben. Een logische optie is om het programma willekeurig te laten kiezen.

Een andere manier zou zijn om het programma eerst cellen met weinig kandidaten in te laten vullen. We kunnen aan een kandidaat immers een soort naïeve kans hangen van $\frac{1}{k}$ waar k het aantal kandidaten in die cel is, deze kans is dus groter voor kandidaten in een cel met minder kandidaten. Wanneer het programma op deze manier een cel heeft gekozen, moet het vervolgens de in te vullen kandidaat bepalen. Ook hier zouden we dit weer willekeurig kunnen doen, of een kandidaat kiezen die het minst voorkomt. De kans dat een kandidaat binnen een bepaalde groep (rij, kolom of blok) correct is, kunnen we namelijk weer naïef inschatten als $\frac{1}{k}$ waar k deze keer staat voor hoe vaak deze kandidaat in de groep voorkomt. Dit zouden we op een iets eenvoudigere manier kunnen benaderen door te tellen hoe vaak de kandidaat in de gehele sudoku voorkomt. Door op deze manier slim te gokken kiezen we dus een kandidaat met naar schatting een zo groot mogelijke kans dat deze correct is.
Tot slot zouden we er natuurlijk ook voor kunnen kiezen om alle cellen en kandidaten simpelweg in een vaste volgorde, van linksboven naar rechtsonder en van laag naar hoog af te lopen. Afhankelijk van de puzzel heb je dan misschien een 'toevalstreffer' die je toevallig binnen weinig gokken oplost, maar die hele andere resultaten oplevert wanneer je deze een kwartslag draait of twee getallen omwisselt.

Natuurlijk willen we meer dan één enkel pad in de graaf verkennen. Dit zouden we op twee manieren kunnen bereiken. Wanneer het programma het einde van een pad bereikt (doodlopen door een fout, of het vinden van een oplossing) kan het een stap terug doen en een ander pad uit proberen. Hierbij zou je echter alle verkende paden in detail bij moeten houden om te zorgen dat het programma deze geen tweede keer doorloopt, en om te bepalen wanneer het programma een volgende stap terug moet zetten. Het systematisch van linksboven naar rechtsonder werken zou dit kunnen vereenvoudigen, maar zoals gezegd verwacht ik dat dit niet erg representatieve resultaten oplevert.
De andere optie is simpelweg om het programma de graaf herhaaldelijk vanaf het begin te laten doorlopen. Om te zorgen dat het programma verschillende paden verkent is het belangrijk dat er bij iedere keuze sprake is van enige willekeur. Met de eerste omschreven methode is dit natuurlijk het geval. Bij de tweede methode is er ook ruimte voor een willekeurig element. Het zou logisch zijn om willekeurig te kiezen tussen meerdere cellen (of kandidaten) die allen een minimaal aantal kandidaten bevatten (of een minimaal aantal keer voorkomen). Het zou echter kunnen voorkomen dat in een puzzel op een bepaald moment er maar één cel met een minimaal aantal kandidaten, en hierbinnen maar één kandidaat dat een minimaal aantal keer voorkomt bestaat, terwijl deze kandidaat in deze cel niet correct is. In dit geval zou het programma 'vastlopen' en telkens opnieuw alleen dit pad doorlopen. Dit kunnen we voorkomen door de selectie van een specifieke kandidaat binnen en geselecteerde cel willekeurig te laten, dus door alleen de cel 'slim' te selecteren. Het is dan wel zo dat we alsnog systematisch maar een deel van de opties verkennen en een deel van de paden uitsluiten. We sluiten hiermee echter nooit de oplossing zelf uit, alleen bepaalde volgorden van invullen. Een mens verkleint hun opties ook op basis van wat zij

logisch vinden, dus dit lijkt niet zo erg.

Wanneer het programma de graaf verkent, is het natuurlijk belangrijk om eigenschappen van de verkende paden bij te houden. Met name zal het programma het aantal succesvolle en doodlopende paden en hun lengtes bij moeten houden.

Samengevat verkent het programma de graaf dus door optioneel naked en/of hidden singles als triviaal in te vullen en vervolgens een volledig willekeurige kandidaat, of een kandidaat binnen een cel met een minimaal aantal kandidaten in te vullen. Dit herhaalt het programma waarbij het het aantal succesvolle en doodlopende paden en hun lengtes bijhoudt.

Aan de hand van deze eigenschappen heeft Daan van Berkel een programma ontwikkeld dat hier te vinden is [1].

4.4.1 Vergelijking Programma

Om de werking van het programma enigszins toe te lichten zal ik het relateren aan een aantal quotes van Andrew Stuart over hoe hij de moeilijkheidsgraad van een sudoku bepaald [6].

> 'I have ignored guessing as a strategy. This is because it is important to have a benchmark and guessing might short-cut a problem or it might hopelessly confuse a potential solution. My suspicion is that many puzzles which are accused of being easier or harder than the published grade have skipped some logic steps and good or bad hunches have been used. This will effect the outcome of the perceived difficulty.'

Ik kies in het programma strategische gokken, maar laat hierbinnen juist ruimte voor willekeur zodat bij het herhaaldelijk oplossen van dezelfde sudoku een soort representatief gemiddelde kan worden genomen. Hierom wil ik dus specifiek strategieën als van linksboven naar rechtsonder vermijden.

> 'However, there are a number of useful pointers that help one to tackle the grading issue. Firstly, if, for example, there are ten squares which can be solved quite independently of each other then this puzzle is clearly easier – at that point, than a different puzzle where each solution relies on you getting the previous ones in a strict sequence. There is a metric of difficultly, therefore, to be gleaned purely by counting the opportunities to solve at all stages of the game. The eye and the mind can only cope with one opportunity and if it is seized, a number is placed, then the board needs to be re-checked to see the knock on effect. 'Bottlenecks' occur if there are few or only one chance to make a correct deduction and these make a more difficult puzzle.'

Vergelijkbaar kan een sudoku op een moment in verhouding veel correcte strategische gokken bevatten (hopelijk analoog aan veel mogelijkheden voor het vinden van antwoorden met (makkelijke) strategieën). Dan zouden we vanaf dat moment verwachten veel verschillende paden naar de oplossing te vinden in het programma (een breed diagram, met letterlijk weinig 'bottlenecks'). Als dit juist weinig is verwachten we dat relatief weinig van de verkende paden naar een oplossing leiden (en er dus veel doodlopen).

> 'Each strategy can be ranked by how hard they are to spot in real life, how often they are needed and how much damage they do (number of eliminations). There are many ways to gauge the usefulness of a given strategy. If scores and weights are carefully given to them we can get a tally from the whole solve route – a new metric of difficulty.'

Analoog kan een strategische gok veel trivialen opleveren en dus een hoog 'number of eliminations' hebben. Dit zou dan resulteren in een korter pad.

Ik denk dat de 'slimme' manier waarop het de gokken kiest enigszins lijkt op de strategieën van naked candidates, wat veel voorkomt in makkelijkere sudoku's, maar weinig lijkt op de complexere strategieën. Hierom verwacht ik dat dit programma makkelijker onderscheid kan maken tussen sudoku's aan de makkelijke kant van het spectrum.

5 Experimenten

5.1 Opzet

Om gegevens te verzamelen over in hoeverre het beschreven programma [1] gebruikt kan worden om de moeilijkheidsgraad van sudoku's in te schatten, is het nodig om een aantal experimenten uit te voeren.

Natuurlijk is het belangrijk om het programma toe te passen op een spectrum aan moeilijkheidsgraden. Omdat Hodoku het meest transparant is in het bepalen van de moeilijkheidsgraad, en sudoku's op een aangegeven niveau kan genereren heb ik besloten om de sudoku's hiermee te generen. Ik wil hierin minimaal 2 sudoku's van ieder niveau (waarvan er 5 zijn) meenemen, en het spectrum aan mogelijke scores zo goed mogelijk afdekken. ook wil ik het programma op de 'onoplosbare' sudoku van Arto Inkala loslaten. De uiteindelijk gebruikte sudoku's, hun score en niveau is terug te vinden in de Appendix 8.1.

Naast de sudoku's is er ook een variabele in de vorm van de opties van het programma. Ik ben hierbij het meest geïnteresseerd om verschillen in de resultaten tussen het 'slimme' en willekeurige gokken te zien. Ook is er de mogelijkheid om naked en hidden singles wel of niet als triviaal te beschouwen. Dit levert dus vier verschillende methodes op:

- Methode 1, naked en hidden singles gratis, willekeurige selectie
- Methode 2, naked en hidden singles gratis, slimme selectie
- Methode 3, niks gratis, willekeurige selectie
- Methode 4, niks gratis, slimme selectie

Ik schat in dat het voldoende zal zijn om per methode en sudoku het programma 5000 pogingen te laten doen.

5.2 Verwachtingen

Zoals ook omschreven bij het bespreken van de graaf, verwacht ik met name dat de verhouding tussen succesvolle en gefaalde paden en de lengte van het kortste succesvolle pad indicatoren van de moeilijkheidsgraad van een sudoku. Specifiek verwacht ik dat een makkelijke sudoku in verhouding meer succesvolle paden oplevert, en een korter succesvol pad gevonden kan worden dan bij een moeilijke sudoku. Hierna verwacht ik dat ook de verdeling van de lengtes van de succesvolle paden aangeeft dat en sudoku makkelijker is wanneer deze zich concentreert rond kortere paden.

Ik verwacht dat methode 2 in verhouding meer succesvolle paden zal opleveren dan methode 1 omdat, zoals besproken, de slimme selectie gokken maakt met een hogere kans om correct te zijn en dit zal leiden tot een hogere proportie

succesvolle paden. Om dezelfde reden verwacht ik dat methode 4 meer succesvolle paden op zal leveren dan methode 3. Wel zou het zo kunnen zijn dat er een bepaalde kandidaat in een cel bestaat die veel eliminaties oplevert wat dus mogelijk een kort pad oplevert (wanneer naked en hidden singles gratis zijn). Als dit echter plaatsvindt in een cel met een niet minimaal aantal kandidaten zal het slimme programma dit niet (meteen) uitproberen. Hierdoor zou het dus kunnen gebeuren dat methode 1 kortere succesvolle paden vindt dan methode 2.

Om dit te illustreren heb ik de graaf voor een 4×4 sudoku gedeeltelijk uitgewerkt. Vanwege het formaat kan ik deze graaf hier niet weergeven. Een afbeelding ervan is op deze pagina te vinden. Hierbij leek het erop dat alle 4×4 sudoku's opgelost kunnen worden met alleen naked en hidden singles. Omdat deze graaf dus anders triviaal zou zijn, heb ik het invullen van naked en hidden singles hier dus wel als een stap geteld, niet gratis. De toestanden en pijlen weergeven in zwart/blauw geven toestanden en keuzes aan die het programma met slimme methodes zou kunnen doorlopen (methode 4), deze zou het willekeurige programma (methode 3) natuurlijk ook kunnen doorlopen. De rode en groene toestanden en pijlen geven keuzes aan die alleen het willekeurige programma zou kunnen doorlopen. Naked singles heb ik gemarkeerd met een achtergrondkleur, wanneer deze aanwezig zijn, wat hier altijd het geval is, zal het slimme programma altijd kiezen om een naked single in te vullen. De vroegste toestanden die 'triviaal' op te lossen zijn, omdat ze alleen nog naked singles bevatten, zijn omcirkeld.
We zien dat het slimme programma alleen naked singles invult, en hierdoor alleen correcte antwoorden invult, en dus ook alleen correcte paden oplevert. We zien in de rode toestand dat het willekeurige programma een fout antwoord heeft ingevuld en er een lege cel overblijft, dit pad loopt dus dood. De slimme methode zal dus meer correcte paden opleveren dan de willekeurige methode.
We zien ook dat de slimme methode pas na 4 stappen een 'triviaal' op te lossen toestand tegenkomt, terwijl de willekeurige methode dit al na 3 stappen kan. In dit geval zijn er geen gratis stappen, dus is ieder pad naar de oplossing even lang, namelijk het aantal in te vullen cellen. Het valt echter voor te stellen dat in een versie van het programma waar wel gratis stappen bestaan, het willekeurige programma ook eerder een triviale toestand, en dus een korter succesvol pad kan vinden dan het slimme programma.

Omdat, zoals besproken, het gratis maken van naked en hidden singles de doorzochte graaf enigszins vereenvoudigd door paden te verkorten en een aantal fouten voorkomt, verwacht ik dat methode 1 en 2 meer en kortere succesvolle paden zullen vinden dan methode 3 en 4 respectievelijk. Vanuit ervaringen met eerdere versies van het programma verwacht ik dat specifiek methode 3 bijzonder weinig succesvolle paden op zal leveren.

5.3 Bespreking

Een uitgebreide tabel met alle resultaten is te vinden in de Appendix 8.1.

Methode 3
Er is te zien dat methode 3 alleen gefaalde paden opleverde, hierom laat ik deze verder buiten beschouwing. Dit is logisch omdat dit programma iedere lege cel, waarvan er meestal meer dan 50 zijn, met een gok moet invullen. Hierdoor moet het dus erg veel opeenvolgende, willekeurige gokken maken, wat natuurlijk maar met extreem kleine kans een oplossing oplevert.

Methode 1 en 2
We zien dat dat, zoals verwacht, methode 2 meer succesvolle paden oplevert dan methode 1. Ook zien we bij de Inkala puzzel duidelijk terug dat methode 2 langere paden oplevert dan methode 1.

Methode 4
Methode 4 levert de minste succesvolle en de langste paden op, de resultaten zijn hierom slecht vergelijkbaar met de andere methodes, toch zijn er wel een aantal interessante patronen te zien. Alle succesvolle paden hebben hierbij vanzelfsprekend een lengte gelijk aan het aantal lege cellen in de puzzel. De doodlopende paden hebben allen een lengte van minimaal 2 groter dan het aantal naked singles dat aan het begin van de puzzel achtereenvolgend ingevuld kunnen worden. Wanneer er naked singles zijn zal het programma deze vanzelfsprekend eerst invullen voor het gaat gokken, dus het is logisch dat alle paden langer dan het aantal naked singles zijn. Het zou echter wel mogelijk moeten zijn om in de eerste stap hierna een fout te maken. Ik geloof dat dit ook gebeurt, maar dat het programma deze fout pas een stap later kan herkennen.
Ook opvallend is dat in de verdeling van de lengtes van de doodlopende paden bij deze methode vaak meerdere pieken te zien zijn, wat bij methode 1 en 2 niet het geval is. Ik geloof dat dit te maken heeft met een soort periodiciteit in het aantal ontstane naked singles, bij een aantal opeenvolgende gokken kunnen steeds meer fouten ontstaan, maar uiteindelijk ontstaat er ook een nieuwe lading naked singles die dan ingevuld kunnen worden waarbij er geen nieuwe fouten ontstaan, maar nog wel een klein aantal zichtbaar worden.
Deze methode geeft ook erg verschillende resultaten voor de twee easy puzzels. Dit heeft ermee te maken dat Easy 1 voornamelijk gebruik maakt van naked singles, terwijl Easy 2 vooral gebruik maakt van hidden singles. Zoals gezegd vult het programma naked singles altijd correct in, maar blijkbaar heeft het veel meer moeite met hidden singles. Toch is dit een 'makkelijke' strategie, dus zouden we willen dat het 'slimme' programma hier niet tegenaan zou lopen. Dit zou een aanleiding kunnen zijn om de werking van het 'slimme' programma te heroverwegen.

Lengtes (methode 1 en 2)
De Easy puzzels zijn triviaal op te lossen met naked en hidden singles en worden

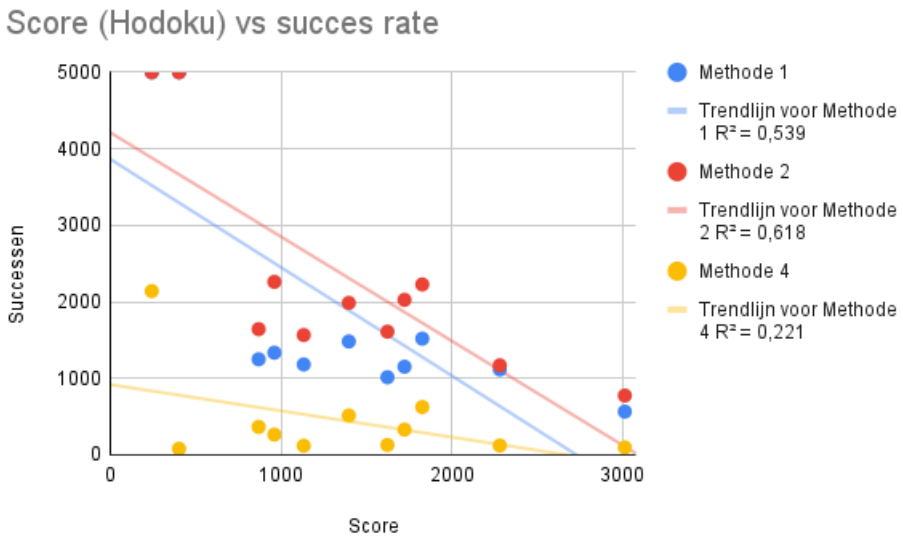
daarom altijd binnen 0 stappen opgelost zoals verwacht.
Wanneer we de Inkala puzzel buiten beschouwing laten, zien we, tegen verwachtingen in, dat alle andere puzzels binnen 1 stap opgelost kunnen worden. Dit betekent dus voor iedere van deze puzzels, dat deze slechts één extra hint verwijderd zijn van een triviaal (met alleen naked en hidden singles) oplosbare puzzel. Dit kan dus blijkbaar niet erg goed gebruikt worden als maatstaaf voor de moeilijkheidsgraad van een puzzel. Wel zou het interessant zijn om te kunnen zien wat hier de doorslaggevende extra hint is.
Bij de Inkala puzzel zien we zoals gezegd dat methode 2 langere paden oplevert dan methode 1. Zelfs met methode 1 is het kortste succesvolle pad van lengte 2, langer dan bij alle andere puzzels. De Inkala puzzel is een bijzonder geval van een extreme puzzel in de zin dat deze niet met de bekende strategieën opgelost kan worden, maar toch een uniek bepaalde oplossing heeft. Dit doet vermoeden dat een kortst succesvolle pad van een lengte groter dan 1 misschien wel een indicatie kan zijn voor een erg extreem oplosbare puzzel, of een niet met strategieën oplosbare puzzel.

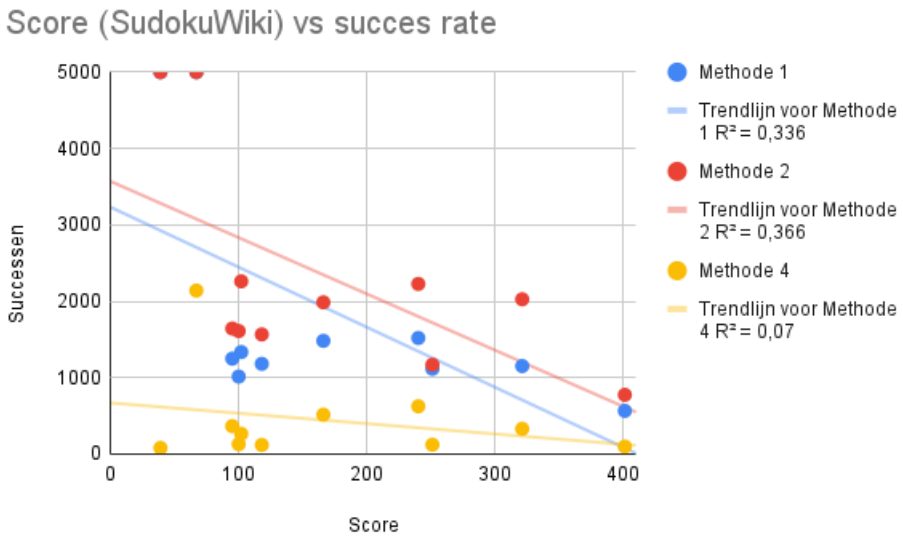
Verhoudingen
Zoals gezegd bleek de lengte van het kortste succesvolle pad een slechte indicator van de moeilijkheidsgraad. Om concreet de samenhang tussen de moeilijkheidsgraad en het aantal succesvolle paden te bekijken, is deze informatie hier kort samengevat.

	Hodoku		SudokuWiki		Methode		
Naam	Niveau	Score	Niveau	Score	1	2	4
Easy 1	Easy	242	Moderate	67	5000	5000	2140
Easy 2	Easy	402	Gentle	39	5000	5000	80
Medium 1	Medium	868	Tough	95	1249	1643	365
Medium 2	Medium	960	Tough	102	1335	2261	265
Hard 1	Hard	1132	Tough	118	1181	1566	119
Hard 2	Hard	1396	Very Hard	166	1482	1986	515
Unfair 1	Unfair	1622	Tough	100	1014	1610	129
Unfair 2	Unfair	1722	Very Hard	321	1152	2026	330
Extreme 1	Extreme	1826	Very Hard	240	1519	2227	625
Extreme 2	Extreme	2280	Very Hard	251	1116	1170	122
Extreme 3	Extreme	3012	Very Hard	401	565	775	97
Inkala	Extreme	21342	-	-	11	22	2

Hierin is het nogmaals duidelijk dat de moeilijkheidsgraad subjectief is, alhoewel de indelingen van Hodoku en SudokuWiki grofweg overeen komen, zijn er ook veel verschillen te zien.
Omdat de Inkala puzzel extreem moeilijk is, en SudokuWiki hier geen moeilijkheidsgraad van kan bepalen, zullen we deze buiten beschouwing laten.

Gebaseerd op deze data kunnen we met behulp van Google Spreadsheets de volgende spreidingsdiagrammen maken:

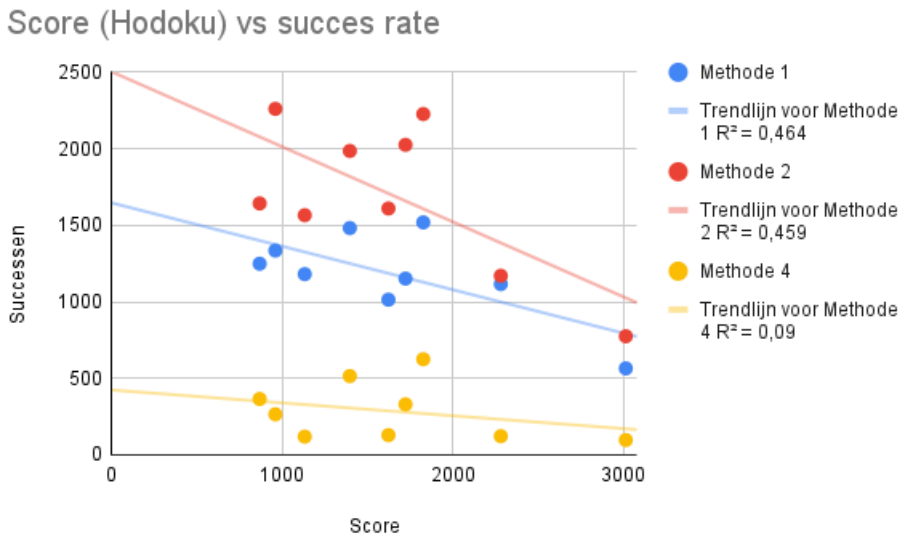


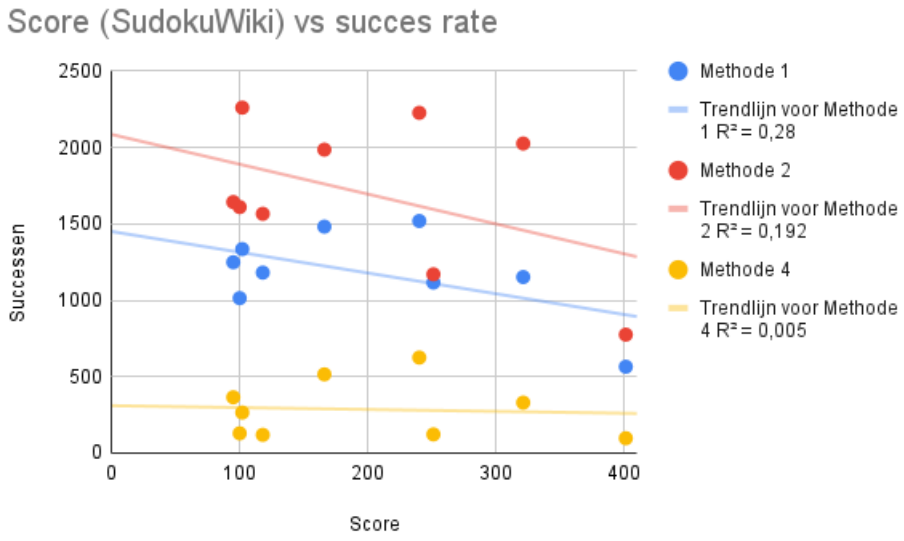


Hierin is de trendlijn bepaald op basis van lineaire regressie, waarbij R de correlatiecoëfficiënt en R^2 de bijbehorende determinatiecoëfficiënt is.
Voor Hodoku hebben methode 1 en 2 een redelijk hoge R^2, dit impliceert een sterk verband. Dit wordt echter sterk beïnvloed door extremen, zoals de resultaten van de twee easy sudoku's, deze waren als enige triviaal op te lossen en kregen daarom een extreem veel successen. We zien meteen dat er sprake is van een veel lagere R^2, en dus een zwakker verband met methode 4 en met de scores

volgens SudokuWiki.

Weglaten van de easy puzzels geeft:

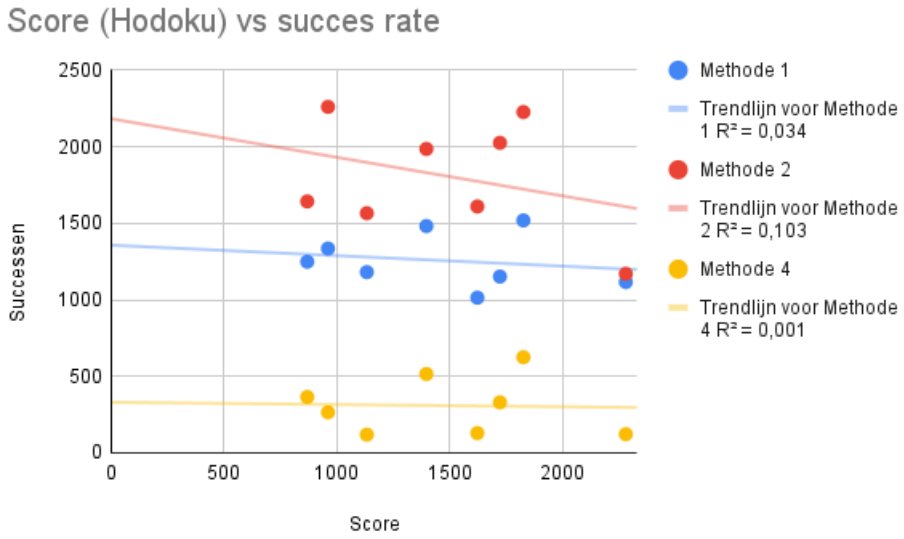


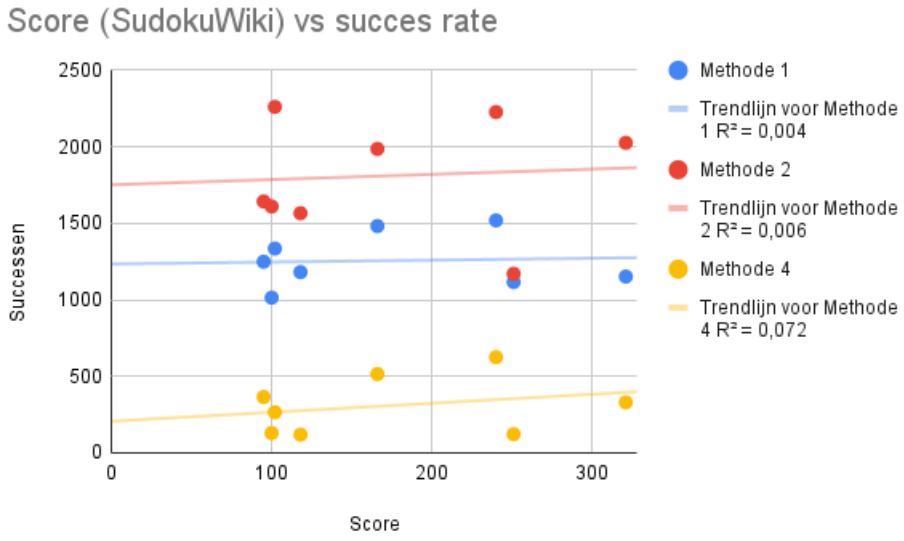


Hierin is bij Hodoku met methode 1 en 2 sprake van een iets zwakker verband, maar nog steeds redelijk sterk. De moeilijkste sudoku levert echter ook een wat extreem datapunt. Er valt te discussiëren over of het terecht zou zijn om deze

data achterwege te laten. Echter, de meeste sudoku's vallen natuurlijk ergens in het midden van het spectrum, dus natuurlijk zijn we het meest geïnteresseerd in de samenhang bij deze resultaten.

Wanneer we ook de moeilijkste sudoku weg zouden laten, zien we het volgende:





Hier blijft er voor elke methode, behalve methode 2 bij Hodoku, slechts een erg

zwak verband over, en ook bij methode 2 bij Hodoku is er slechts sprake van een zwak tot matig verband. We zien dat dat voor deze 'middenmoot' dat het verband tussen de moeilijkheidsgraad en het aantal succesvolle paden hoogstens matig is.

Bij elk van deze overwegingen is het belangrijk om mee te nemen dat we werken met een zeer beperkt aantal datapunten, en hierdoor dus met weinig zekerheid uitspraken over de relaties kunnen doen. Wel zien we telkens dat methode 2 de hoogste succes 'rate' heeft (meeste succesvolle paden) en ook de hoogste R^2, dus het sterkste verband met de moeilijkheidsgraad.

Mogelijk is er natuurlijk sprake van een niet-lineair verband. Door grofweg een aantal verbanden uit te proberen lijkt logaritmische regressie het sterkste verband op te leveren. Dit zou mogelijk veroorzaakt kunnen worden door het opeenvolgend maken van keuzes, hierdoor zouden we de kans op een succes kunnen benaderen als het product van de eerder besproken naïeve kansen, bijvoorbeeld $\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot ...$ wanneer er telkens een cel met twee kandidaten ingevuld wordt. Om hier een uitspraak over te doen is zeker verder onderzoek nodig.

5.4 Conclusie

Zoals verwacht levert methode 2 in verhouding de meeste succesvolle paden, daarna methode 1, methode 4 en methode 3 achtereenvolgens. Ook kan methode 1 in enkele gevallen kortere succesvolle paden vinden dan methode 2 zoals verwacht.
Tegen mijn verwachtingen in lijkt de lengte van het kortste succesvolle pad echter weinig betekenisvol omdat dit voor alle 'normale' sudoku's 1 is. Wel zou een langere lengte mogelijk een indicatie kunnen zijn voor een 'onoplosbare' sudoku.
Ook tegen mijn verwachting in lijkt het aantal successen niet consistent af te nemen naarmate de puzzel moeilijker wordt. Zoals gezien in de spreidingsdiagrammen lijkt er hierbij over het algemeen de sterkste samenhang tussen het aantal successen van methode 2 en de score volgens Hodoku te bestaan. Zeker voor de gematigde sudoku's lijkt dit verband echter slechts zwak tot matig. Vanwege de kleine omvang van dit experiment is het hiermee niet mogelijk om met redelijke zekerheid een definitieve conclusie te trekken.

5.5 Vervolg

Een logische vervolgstap zou zijn om de omvang van het experiment te vergroten en meer sudoku's te bekijken om zo een sterkere uitspraak over het verband tussen het aantal succesvolle paden en de moeilijkheidsgraad te kunnen doen. Hierbij zou er ook verder onderzocht kunnen worden of dit verband mogelijk van een niet-lineaire vorm is.
Verder zou het programma mogelijk verder aangepast kunnen worden zodat het met 'slimme' gokken beter om kan gaan met hidden singles, niet alleen naked singles zoals omschreven in de bespreking van methode 4. Een methode hiervoor

was al overwogen in Paragraaf 4.4, namelijk om na het selecteren van een cel met een minimaal aantal kandidaten, een kandidaat te selecteren die het minst voorkomt. Dit zou echter kunnen veroorzaken dat het programma vastloopt. Mogelijk zou het programma tegelijkertijd een afweging kunnen maken tussen een cel met een minimaal aantal kandidaten of een kandidaat die een minimaal aantal keer voorkomt. Als het aantal keer dat een kandidaat voorkomt lager is dat het minimale aantal kandidaten in een cel, zou het programma deze kandidaat in een willekeurige cel waarin deze voorkomt, kunnen invullen. Dit dus in tegenstelling tot het huidige programma, wat altijd eerst een cel selecteert.
Ook zou er verder onderzocht kunnen worden of het hebben van een kortste succesvolle pad van lengte groter dan 1 met methode 1 daadwerkelijk een indicator is voor het 'onoplosbaar' of extreem moeilijk zijn van een sudoku. Hiervoor zouden er natuurlijk meer onoplosbare en normale sudoku's onderzocht moeten worden. Als het programma hiervoor aangepast wordt, zou er ook gekeken kunnen worden welke extra hint doorslaggevend is en tot een triviale puzzel leidt.
Zoals genoemd, vermoed ik na er veel gezien te hebben dat 4×4 sudoku's altijd oplosbaar zijn met enkel naked en hidden singles. Ik heb echter nog niet voldoende informatie kunnen vinden om dit te bewijzen, maar dit zou mogelijk ook een interessant onderwerp kunnen zijn.

6 Dankwoord

Ten eerste wil ik mijn begeleider, dr. Wieb Bosma, bedanken voor al zijn hulp, maar ook zijn begrip en geduld. Door zijn begeleiding heb ik me verder kunnen verdiepen in- en vorm kunnen geven aan mijn interesse in sudoku's. Ook wil ik Daan van Berkel bedanken voor het maken van het programma waarmee ik heb geprobeerd de moeilijkheidsgraad te bepalen en voor zijn bijdrage aan onze discussies.
Daarnaast wil ik graag mijn ouders bedanken omdat zij me altijd hebben gesteund en aangemoedigd. In het bijzonder wil ik mijn moeder en oma bedanken omdat zij mij hebben geïntroduceerd aan sudoku's. Ook wil ik graag mijn vriend bedanken voor al zijn steun en hulp.

7 Referenties

[1] Daan van Berkel. Sudoku solver, 2023. [Online; accessed 31-juli-2023].

[2] Bernhard Hobiger. Solving techniques, 2023. [Online; accessed 5-augustus-2023].

[3] Narendra Jussien. *A to Z of Sudoku*. ISTE, 2007.

[4] Gary McGuire, Bastian Tugemann, and Gilles Civario. There is no 16-clue sudoku: solving the sudoku minimum number of clues problem via hitting set enumeration. *Exp. Math.*, 23(2):190–217, 2014.

[5] Jason Rosenhouse and Laura Taalman. *Taking Sudoku seriously. The math behind the world's most popular pencil puzzle*. Oxford: Oxford University Press, 2011.

[6] Andrew C Stuart. Sudoku creation and grading. *Mathematica*, 39(6):126–142, 2007.

[7] Andrew C Stuart. Sudokuwiki, 2023. [Online; accessed 5-augustus-2023].

8 Appendix

8.1 Resultaten

Geordend op methode en puzzel.
De Inkala puzzel is een bijzonder geval van een extreme puzzel.
Ieder experiment bestond uit 5000 pogingen van het beschreven programma om de puzzel op te lossen.
In methode 1 waren naked singles en hidden singles 'gratis' en selecteerde het programma willekeurige cellen, in methode 2 waren naked singles en hidden singles ook gratis en selecteerde het programma cellen met het minste aantal kandidaten, in methode 3 was niks gratis en selecteerde het programma willekeurige cellen en in methode 4 was niks gratis en selecteerde het programma cellen met het minste aantal kandidaten.
Methode 3 is niet op elke puzzel toegepast.

De gebruikte puzzels zijn:
Easy 1 - 000000320140020000260800004000008250004309600051700000500003091000090068098000000
Easy 2 - 032000070401000000000700460100458009900000007300697005013002000000000203040000790
Medium 1 - 5100000000000540079040120603000000200089231000500000060801307091004900000000000051
Medium 2 - 003000004850600030000830060000040970000206000014070000060093000070008095100000700
Hard 1 - 508004000000000240940060000000059004190603028600820000000300860310000000000200903
Hard 2 - 400020300000000800000008049300007014087010930210900007840700000005000000009030006
Unfair 1 - 460800007050630000000000080009060002386000479700080500090000000000058090600007028
Unfair 2 - 010309400096700023300000000000830700020000800070460000000000069600023400026030 50
Extreme 1 - 000000009050000023207008004000400006078692350100003000300700602760000040800000000
Extreme 2 - 000000402307000000000590030502700040004306200060009703050021000000000501409000000
Extreme 3 - 39510006000000000028003650000060005080600030702000700000796001200000000060004975
Inkala - 800000000003600000070090200050007000000045700000100030001000068008500010090000400

	Length	Easy 1		Easy 2		Medium 1		Medium 2		Hard 1		Hard 2		Unfair 1		Unfair 2		Extreme 1		Extreme 2		Extreme 3		Inhuman	
		Succeeded	Gefaalt	Succeeded	Gefaalt	Succeeded	Gefaalt	Succeeded	Gefaalt	Succeeded	Gefaalt	Succeeded	Gefaalt	Succeeded	Gefaalt	Succeeded	Gefaalt	Succeeded	Gefaalt	Succeeded	Gefaalt	Succeeded	Gefaalt	Succeeded	Gefaalt
Methode 1	Totaal	5000	0	5000	0	1249	3751	1335	3665	1181	3819	1482	3518	1014	3986	1152	3848	1519	3481	1116	3884	565	4435	11	4989
	0	5000		5000																					
	1					966	1051	1145	696	818	840	1264	1420	776	164	947	566	1323	1450	804	642	276	740		
	2					250	745	171	845	316	964	206	518	109	765	157	1658	179	635	264	948	202	851	2	86
	3					31	544	19	637	45	812	11	474	35	717	44	705	17	419	45	780	78	670	3	375
	4					1	435		485	2	512	1	357	4	687	4	530		351	3	560	7	575	2	661
	5					1	337		362		309		285		523		391		225		367	2	494	2	707
	6						256		254		160		183		429		252		169		253		366	2	700
	7						166		147		102		122		264		136		96		145		260		625
	8						103		102		64		66		169		91		64		96		168		537
	9						67		59		26		34		109		49		31		43		123		429
	10						29		30		12		30		74		42		15		24		52		294
	11						15		17		9		21		43		14		16		10		46		224
	12						8		16		4		3		23		6		6		12		23		141
	13						8		7		1		4		7		3		2		3		8		90
	14						4		3						5		4		1				4		55
	15						3		1		1				3		1		1				3		29
	16								3						3						1		1		20
	17										1		1										1		6
	18						1				2				1										8
	19																								1
	20								1																
	21																								1
Methode 2	Totaal	5000	0	5000	0	1643	3357	2261	2739	1566	3434	1986	3014	1610	3390	2026	2974	2227	2773	1170	3830	775	4225	22	4978
	0	5000		5000																					
	1					972	777	2012		824		1518	759	1088		1654		1937	1177	566	211	291	1193		
	2					593	982	224	835	677	1212	438	1097	415	2300	310	1234	262	867	381	1038	169	456		
	3					75	760	23	723	62	860	26	615	87	514	56	848	28	250	168	1068	205	727		
	4					3	397	2	491	3	599	2	273	20	303	6	432		174	51	757	87	704		26
	5						230		277		328		137		148		236		112	3	392	21	494	1	388
	6						117		164		191		67		72		102		82	1	196	2	285	6	805
	7						47		97		133		35		27		61		53		96		157	6	1079
	8						29		53		54		14		11		26		31		35		81	6	961
	9						5		43		46		4		7		21		15		21		56	1	729
	10						7		24		9		6		3		7		5		7		26	2	445
	11						2		12		2		3		2		2		4		6		15		249
	12						1		5				2		1				1		3		11		124
	13								8				1		1		3		2				4		68
	14								3				1				1						4		43
	15								2														1		22
	16								1								1						1		11
	17								1						1										6
	18																								6
	19																								1
	20																								1
	21																								1
	22																								
	23																								
	24																								2
	25																								1
	26																								
	27																								1
Methode 3	Totaal	0	5000			0	5000			0	5000			0	5000	0	5000			0	5000				
	0																								
	1		197				112														306				
	2		323				220				156				165		97				391				
	3		375				279				296				301		231				440				
	4		462				399				365				398		315				451				
	5		569				459				486				528		438				464				
	6		554				531				499				541		511				551				
	7		522				557				555				578		602				476				
	8		474				563				548				526		602				435				
	9		399				502				521				518		530				393				
	10		346				397				432				419		439				323				
	11		253				318				344				328		377				244				
	12		209				216				245				212		302				162				
	13		127				171				195				176		213				119				
	14		73				96				135				114		146				93				
	15		57				59				99				81		80				76				
	16		29				40				50				49		52				31				
	17		17				17				24				23		35				19				
	18		13				11				12				17		21				14				
	19		5				4				6				8		12				3				
	20		1								3				7		5				1				
	21		1				5				4				7		1				3				
	22										2						1								
	23		1								1				1										
	24																				1				
	25														1										
	26																								
	27														1										
	28														1										
Methode 4	Totaal	2140	2860	80	4920	365	4635	265	4735	119	4881	515	4485	129	4871	330	4670	625	4375	122	4878	97	4903	2	4998
	0																								
	1																								
	2										231														
	3				52						164				88										
	4				80		296		2		168				91		16						247		
	5				82		170		33		101				134		82						144		41
	6				112		143		95		73				129		89						94		67
	7				145		153		75		73				111		75				511		133		33
	8				124		161		73		147				133		105				241		157		40
	9				118		133		123		129				140		131		203		207		97		86
	10				144		121		95		150				121		161		64		429		97		180
	11				191		107		123		154				93		192		46		321		74		197
	12				230		119		145		130				112		324		121		262		89		88
	13				225		88		190		134				103		281		197		175		70		112
	14				190		86		187		170				107		252		240		168		66		108
	15				142		92		216		163		32		147		218		119		134		85		105
	16				115		78		195		201		121		174		167		88		83		113		200
	17				98		114		231		174		183		169		197		75		80		91		180
	18				70		125		197		164		202		168		209		44		102		79		184
	19				67		200		220		187		211		191		168		32		74		104		250
	20				63		125		516		172		169		196		194		46		75		70		236
	21				76		157		186		147		159		170		154		39		73		98		267
	22				96		112		179		152		204		185		159		65		75		151		259
	23				123		105		130		140		253		204		141		51		100		130		231
	24				138		77		129		143		276		246		126		54		112		151		267
	25				158		81		116		141		279		250		107		67		109		136		247
	26				117		73		93		141		265		236		109		61		147		164		210
	27				121		117		97		134		246		159		125		89		126		186		215
	28				95		139		99		141		255		156		111		158		148		168		196
	29				115		168		85		102		252		134		109		153		119		215		159
	30				131		194		90		111		214		139		85		199		90		185		141
	31		516		172		151		94		100		208		98		105		276		102		210		111
	32		240		178		156		106		91		149		62		89		205		94		171		99
	33		123		145		160		97		99		124		55		57		289		84		147		97
	34		101		146		107		96		76		124		45		49		303		84		162		78
	35		219		134		88		91		55		104		52		52		226		81		159		79
	36		396		201		87		63		53		75		62		46		177		79		154		48
	37		238		118		74		53		54		84		45		50		135		86		123		39
	38		175		195		61		49		36		79		38		29		94		72		121		29
	39		156		89		56		52		21		59		22		32		73		51		83		42
	40		171		44		51		37		29		45		22		30		52		52		45		20
	41		167		36		37		17		14		39		25		11		46		40		38		16
	42		143		27		32		30		6		33		26		15		48		22		23		11
	43		99		20		22		11		3		23		13		8		50		25		18		11
	44		58		5		10		14		6		9		15		10		33		12		40		6
	45		33		1		6		5		2		8		13		7		26		9		8		5
	46		18		1		3		2				2		1		3		6		7		1		9
	47		4				1		2				1				1		5		3				2
	48		1						4												3				
	49																				1				3
	50																								1
	51																								2
	52					365		265																	
	53	2140		80						119				129		330						97			
	54											515						625							
	55																			122					
	56																								
	57																								
	58																								
	59																								
	60																							2	

8.2 Strategieën

8.2.1 SudokuWiki

Volgorde van toepassing SudokuWiki:

- Naked Singles
- Hidden Singles
- Naked Pairs/Triples
- Hidden Pairs/Triples
- Naked/Hidden Quads
- Pointing pairs
- Box/Line Reduction
- X-Wing
- Simple Colouring
- Y-Wing
- Swordfish
- XYZ Wing
- BUG
- X-Cycles
- XY-Chain
- 3D Medusa
- Jellyfish
- Unique Rectangles
- Fireworks
- SK Loops
- Extended Unique Rectangles
- Hidden Unique Rectangles
- WXYZ Wing
- Aligned Pair Exclusion
- Exocet

- Grouped X-Cycles
- Empty Rectangles
- Finned X-Wing
- Finned Swordfish
- Alternating Inference Chains
- Sue-de-Coq
- Digit Forcing Chains
- Nishio Forcing Chains
- Cell Forcing Chains
- Unit Forcing Chains
- Almost Locked Sets
- Death Blossom
- Pattern Overlay Method
- Quad Forcing Chains
- Bowman's Bingo

8.2.2 Hodoku

Volgorde toepassing, level en score Hodoku:

- Full House, Easy, 4
- Naked single, Easy, 4
- Hidden Single, Easy, 14
- Locked Pair, Medium, 40
- Locked Triple, Medium, 60
- Locked Candidates Pointing, Medium, 50
- Locked Candidates Claiming, Medium, 50
- Naked Pair, Medium, 60
- Naked Triple, Medium, 80
- Hidden Pair, Medium, 70
- Hidden Triple, Medium, 100

- Naked Quadruple, Hard, 120
- Hidden Quadruple, Hard, 150
- X-Wing, Hard, 140
- Swordfish, Hard, 150
- Jellyfish, Hard, 160
- Remote Pair, Hard, 110
- Bivalue Universal Grave +1, Hard, 100
- Skyscraper, Hard, 130
- 2-String Kite, Hard, 150
- Turbot Fish, Hard, 120
- Empty Rectangle, Hard, 120
- W-Wing, Hard, 150
- XY-Wing, Hard, 160
- XYZ-Wing, Hard, 180
- Uniqueness Test 1, Hard, 100
- Uniqueness Test 2, Hard, 100
- Uniqueness Test 3, Hard, 100
- Uniqueness Test 4, Hard, 100
- Uniqueness Test 5, Hard, 100
- Uniqueness Test 6, Hard, 100
- Hidden Rectangle, Hard, 100
- Avoidable Rectangle Type 1, Hard, 100
- Avoidable Rectangle Type 2, Hard, 100
- Finned X-Wing, Hard, 130
- Sashimi X-wing, Hard, 150
- Finned Swordfish, Unfair, 200
- Sashimi Swordfish, Unfair, 240
- Finned Jellyfinsh, Unfair, 250

- Sashimi Jellyfish, Unfair, 260
- Sue de Coq, Unfair, 250
- Simple Colors, Hard, 150
- Multi Colors, Hard, 200
- X-Chain, Unfair, 260
- XY-Chain, Unfair, 260
- Nice Loop/AIC, Unfair, 280
- Grounded Nice Loop/AIC, Unfair, 300
- Almost Locked Set XZ-Rule, Unfair, 300
- Almost Locked Set XY-Wing, Unfair, 320
- Almost Locked Set Chain, Unfair, 340
- Franken X-Wing, Unfair, 300
- Franken Swordfish, Unfair, 350
- Finned Franken X-Wing, Unfair, 390
- Finned Franken Swordfish, Unfair, 410
- Forcing Chain, Extreme, 500
- Forcing Net, Extreme, 700
- Brute Force, Extreme, 10000