

Rapportage



Naam: Carlo Keizer
Email: carlo.keizer@student.hu.nl
Klas: TICT-AAI-V1A
Datum: 02-07-2019

Inhoudsopgave

Projectsamenvatting.....	3
Probleembeschrijving	3
Eisen van de oplossing	3
Beschrijving van het algoritme.....	3
Alldifferent.....	3
Tabu search	3
Voor- en nadelen	4
Genomen overwegingen	4
Poster	5
Bronvermelding	6

Projectsamenvatting

In de projectsamenvatting zal een samenvatting plaatsvinden van de belangrijke onderdelen die ik tijdens mijn project heb uitgevoerd.

Probleembeschrijving

Mensen die af en toe eens een keertje een sudoku maken om de tijd te doden, lopen soms wel eens vast bij het maken van een sudoku. Soms kom je dan weer verder, maar soms ook niet. En sommige mensen vinden het erg vervelend als ze dan hun puzzel niet kunnen afmaken of niet verder komen.

Eisen van de oplossing

Hier zal een beschrijving plaatsvinden van de eisen van de oplossing

- Er moet een oplossing gegeven kunnen worden, van een gegeven sudoku.
- Er moeten een hints gegeven kunnen worden, over een gegeven sudoku.
- Er moet gedetecteerd kunnen worden als de gegeven sudoku niet oplosbaar is.
- Een sudoku moet kunnen uitgelezen worden uit een bestand.
- Een sudoku moet uitgelezen kunnen worden uit een tekstregel.

Beschrijving van het algoritme

Voor het oplossen van het sudoku probleem, heb ik gelezen in een document van (Soto, Crawford, Galleguillos, Paredes, & Norero, 2015) dat je hiervoor het beste gebruik kan maken van het alldifferent algoritme in combinatie met het tabu search algoritme. Ik heb deze twee algoritmes hieronder verder toegelicht.

Alldifferent

Het alldifferent algoritme zorgt er voor dat je van de sudoku weet wat de mogelijke waardes zijn van de onbekende in de sudoku. Dit word gedaan door eerst bij alle onbekende alle mogelijke waardes te geven, in dit geval is dat 1 t/m 9. Vervolgens kan je per rij, kolom en blok alle bekende waardes uit de lijst halen met alle mogelijke waardes van de onbekende. Wanneer je dit voor alle rijen, kolommen, en blokken hebt gedaan dan weet je dus wat de mogelijk waardes zijn per vakje. Hier kan je vervolgens weer sudoku strategieën op toepassen, zodat je ook daadwerkelijk waardes kan gaan invullen.

Tabu search

Bij toepassen van het alldifferent algoritme en sudoku strategieën, kan het zijn dat het alldifferent algoritme de mogelijke waardes van de onbekende in de sudoku in meer kan verminderen. Het algoritme loopt dus vast. Hiervoor heb ik gebruik gemaakt van het tabu search algoritme. Wat dit algoritme doet is het maken van een gok in de sudoku. Als later blijkt dat de gemaakte gok fout was, dan word deze gok in een lijst geplaatst zodat het algoritme weet welke gok er niet meer gemaakt hoeft te worden. Met de combinatie van deze twee algoritmes kan je dus tot de oplossing komen van elke sudoku.

Voor- en nadelen

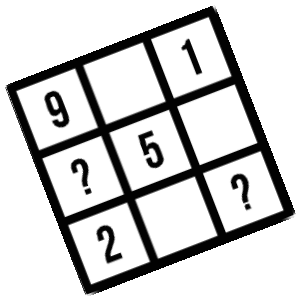
In het zelfde document van (Soto et al., 2015) word het AC3 algoritme in combinatie met het tabu search ook genoemd. Dit algoritme is ook te gebruik voor het oplossen van een sudoku. Een groot nadeel aan dit algoritme is dat het een langzamer algoritme is dan het alldifferent algoritme in combinatie met tabu search. Dat is ook de reden dat ik voor het alldifferent tabu search heb gekozen. Ik heb er verder geen tijd in gestoken om uit te zoeken waarom dit algoritme langzamer is. Want in dat zelfde document van (Soto et al., 2015) staan ook de cijfers waar de twee algoritmes vergeleken zijn. En ik besteed mijn tijd tijdens het project liever aan het project zelf.

Genomen overwegingen

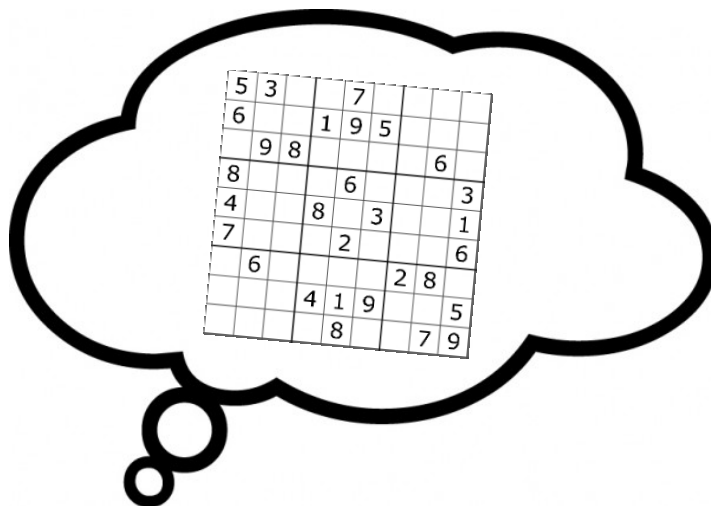
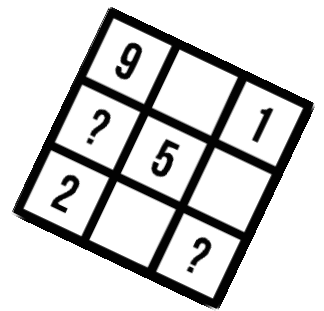
Alle overwegingen die ik heb gemaakt met betrekking tot het algoritme zijn hier beschreven. Om te beginnen heb ik een keuze moeten maken hoeveel strategieën ik ging gebruiken voor het oplossen van de sudoku. Ik heb er voor gekozen om twee simpele strategieën te implementeren. Dit heb ik gedaan omdat ik de twee algoritmes in evenwicht moet houden. Want een oplossing zoeken met strategieën kost veel meer rekenkracht dan het tabu-search algoritme. Het tabu search algoritme maak namelijk meer gebruik van het geheugen. Ook al scheelt het niet veel, en zal het voor een gemiddelde computer niks uitmaken. Het is toch iets waar je over na moet denken bij het maken van een algoritme.

Mijn volgende overweging die ik gemaakt heb, hoe ik het beste een gok kon maken, zodat de kans zo klein mogelijk was dat de gok fout is. Daar was ik eigenlijk vrij snel uit. Hiervoor heb ik een algoritme geschreven die de eerste plek kiest waar zo min mogelijk mogelijkheden zijn om een gok te maken. Want hoe minder mogelijkheden een positie heeft, hoe kleiner de kans is dat je een verkeerde gok maakt. En de kans op onderbrekingen van het oplossen van de sudoku word hierdoor ook kleiner. Waardoor je dus sneller tot een oplossing bent van de sudoku.

Poster



Sudoku oplosser/helper



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



Bronvermelding

Soto, R., Crawford, B., Galleguillos, C., Paredes, F., & Norero, E. (2015). A Hybrid alldifferent-Tabu Search Algorithm for Solving Sudoku Puzzles, 1-11.
<https://www.hindawi.com/journals/cin/2015/286354/>