

Practicumverslag Computatieve Intelligentie

Cornelis Bouter (studentnummer) en Roel Brouwer (3976866)

27 juni 2014

Abstract

Chapter 1

Implementatie algoritmen

1.1 Hill-climbing

1.1.1 Random Restart

1.1.2 Iterated Local Search

Voor het implementeren van *iterated local search* is het nodig geweest de zoekoperator iets aan te passen. De zoekoperator die voor het beklimmen van de heuvel, het *hill-climbing* deel, gebruikt wordt, loopt alle swap-mogelijkheden in een bepaalde volgorde door, en kiest de eerste die een verbetering is. Voor *iterated local search* is een *random walk* nodig. Een *random walk* geeft niets om het antwoord op de vraag of een opvolger-toestand al dan niet een verbetering is. Het moet een willekeurige opvolger-toestand kunnen kiezen, ongeacht de waarde die de evaluatiefunctie aan die toestand geeft. Wanneer de oude implementatie gebruikt zou worden, zou het dan steeds voor de eerste optie kiezen: het swappen van de eerste twee niet-vaste vakjes in het blok in de linksbovenhoek van de sudoku. Dat willen we niet, daarom veranderen we de zoekoperator-functie zodat deze twee willekeurige vakjes (binnen hetzelfde blok - natuurlijk) van plaats verwisseld worden:

```
RandomZoekOperator (Sudoku)
    // g en h representeren het blok waarin het verwisselen wordt
    uitgevoerd.
    g, h = randomInt(0, n) * n;
    // i, j, k en l representeren de twee vakjes die verwisseld
    worden.
    i, j, k, l = randomInt(0, n);
    while ((i,j) en (k,l) representeren hetzelfde vakje of
        (i,j) of (k,l) is een vast vakje)
        // Nieuwe waarden voor i, j, k en l.
        i, j, k, l = randomInt(0, n);
    // Verwissel de genoemde vakjes in de sudoku en return het
```

```
        resultaat  
    return Sudoku.verwissel(g + i, h + j, g + k, h + l)
```

1.2 Constraint Satisfaction

1.2.1 Backtracking

1.2.2 Forward Checking

Chapter 2

Experimentele resultaten

2.1 Hill-climbing

2.1.1 Random Restart

2.1.2 Iterated Local Search

2.2 Constraint Satisfaction

2.2.1 Backtracking

2.2.2 Forward Checking