Computationele Intelligentie: deeltoets 1

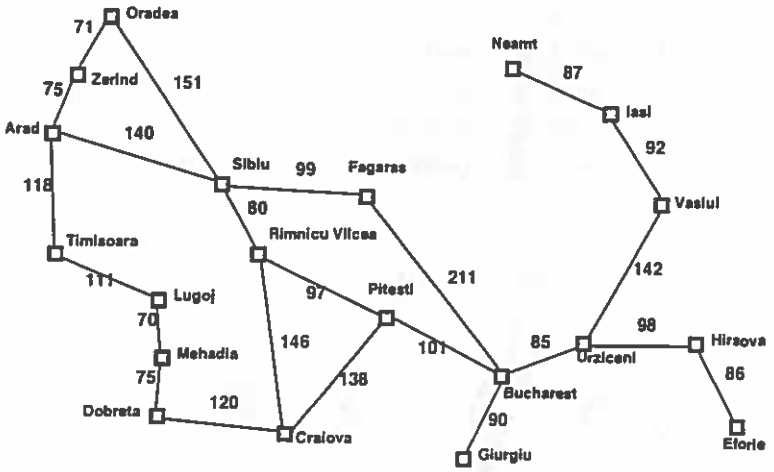
26 mei 2016, **8.30 – 10:30 uur.**

Dit is een gesloten boek tentamen. Rekenmachines (en telefoons) zijn niet toegestaan. Bij elke opgave is vermeld hoeveel punten met de respectievelijke onderdelen van de opgave te verdienen zijn. In totaal kunnen 50 punten verkregen worden.

Opgave 1

(Totaal = 10 ptn.: 1: 5 ptn., 2: 5 ptn.)

Beschouw een kaart met steden en wegen zoals weergegeven in de volgende figuur.



Straight-line distance to Bucharest	
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Het doel is een kortste pad te vinden van de stad Oradea naar Boekarest. In de kaart staan bij de wegen de afstanden aangegeven. In de tabel staan de afstanden naar Boekarest in vogelvlucht.

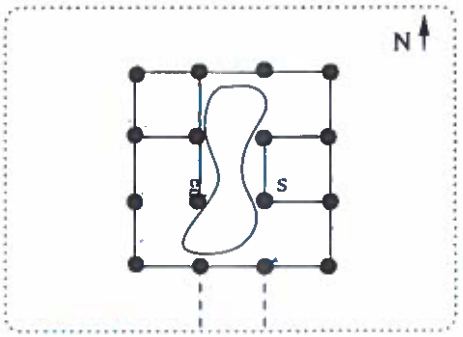
1. Geef de zoekboom die gegenereerd wordt door het A^*-zoekalgoritme met de afstand in vogelvlucht als heuristische functie.

2. Geef de zoekboom die gegenereerd wordt door het Depth-First Branch-and-Bound algoritme. De successors van de knopen worden doorlopen in wijzerszin, vertrekkend van bovenaan (= 12 uur).

Opgave 2

(Totaal = 10 ptn.: 1: 2 ptn., 2: 2 ptn., 3: 2 ptn., 4: 2 ptn., 5: 2 ptn.)

Beschouw een park met een vijver, zoals afgebeeld in Figuur 1. Rondom de vijver lopen paden, die in de figuur weergegeven zijn met getrokken lijnen. De dichte cirkeltjes in de figuur geven speciale punten weer, zoals hoekpunten, kruispunten en begin- of eindpunten. Een wandelaar wil in het park van punt s naar punt g lopen. Hij loopt via de paden van punt naar punt. Noem $h(n)$ de hemelsbrede afstand in de **horizontale** richting van n naar g, zonder rekening te houden met obstakels. Veronderstel dat de knopen geëxpandeerd worden



Figuur 1: Een vijver in een park

in de volgorde: **zuid-west-noord-oost**. Veronderstel dat in de zoekboom geen toestanden worden opgenomen die al op het pad van de wortel naar de huidige toestand voorkomen.

Geef voor de onderstaande zoekalgoritmen met heuristische functie $h(n)$ steeds de lengte van het (eerste) gevonden pad en het aantal toestanden dat wordt geëxpandeerd:

1. Best-first search.
2. A-algoritme met evaluatiefunctie $f(n) = g(n) + h(n)$ met $g(n)$ gelijk aan de lengte van het gevolgde pad van s naar n.
3. Hill-climbing algoritme.
4. Lokale beam-search met $k = 2$.
5. Tabu search met de lengte van de taboe-lijst resp. gelijk aan 2.

Opgave 3

(Totaal = 10 ptn.: 1: 3 ptn., 2: 7 ptn.)

Beschouw het constraint satisfaction probleem $CSP = (V, D, C)$ met

- $V = \{V_1, V_2, V_3\}$;
- $D = \{D_i \mid i = 1, \ldots, 3\}$ met $D_1 = D_2 = D_3 = \{1, 2, 3, 4\}$;
- C bevat de volgende niet-universele constraints: $V_1 > V_2$; $V_1 > V_3$; $V_2 \neq V_3$;.

1. Geef alle redundante tupels in de constraints van het probleem.
2. Reduceer het probleem CSP tot een equivalent pijlconsistent probleem CSP' door toepassing van het arc-consistency-3 (AC-3) algoritme. Geef hierbij de opeenvolgende waarden van de constraint-set, de onderzochte constraint, en het effect op het domein.

Opgave 4

(Totaal = 20 ptn.: 1: 4 ptn., 2: 4 ptn., 3: 4 ptn., 4: 4 ptn., 5: 4 ptn.)

1. Hoe kan je in simulated annealing een goede begintemperatuur bepalen ? Gegeven een oplossing T en zijn successor S; hoe bereken je de acceptatie kans bij simulated annealing voor een minimalisatie probleem ?

2. Hoe wordt een variantprobleem gedefinieerd bij heuristisch zoeken ? Wat is het nut van variantproblemen ?

3. Bij het 8-koninginnen probleem plaatsen we koningingen in oplopende volgorde van de rijen op de volgende vakken: (1, 1), (2, 8), (3, 6), (4, 2), en (5, 7) waarbij (i,j) de i^e rij en j^e kolom voorstellen. We gebruiken het Maintaining Arc Consistency (MAC) algoritme om het constraint probleem dynamisch te reduceren. Wat wordt de initiële constraint verzameling voor het MAC algoritme nadat we op de 5e rij de koningin in kolom 7 hebben geplaatst ?

4. Beschouw een zoekboom T met een vertakkingsfactor b en diepte d: Hoeveel ruimte gebruikt dynamische depth-first search voor het opslaan van het front op diepte d ?

5. Beschouw 2 oplossingen voor het kleuringsprobleem voor een graaf met 12 knopen en 3 kleuren:

   $G1 : k_1 = g, k_2 = r, k_3 = b, k_4 = b, k_5 = r, k_6 = g, k_7 = r, k_8 = b, k_9 = r, k_{10} = b, k_{11} = g, k_{12} = r.$

   $G2 : k_1 = g, k_2 = g, k_3 = g, k_4 = g, k_5 = g, k_6 = b, k_7 = b, k_8 = b, k_9 = r, k_{10} = r, k_{11} = r, k_{12} = r.$

   Genereer een nieuwe oplossing door toepassing van de Greedy Partitioning Crossover (GPX) operator.