README

Rotaru Razvan Paul
341C3

Avantaje si dezavantaje:

- BFS – exploreaza toate “drumurile” posibile pana la prima stare finala insa este cea mai lenta strategie
- Uniform cost search – exploreaza si dezvolta drumruile conform unui minim cost de translatie intre stari
  – poate omite drumuri ce vor fi mai profitabile in viitor dar pornesc cu stari costisitoare
  – timpul de executie de obicei mare
- DFS – exploreaza rapid arborele de stari, oprindu-se la prima solutie, nu ofera siguranta unei solutii optime
- DFS limitat – exploreaza solutiile pana la un anumit nivel de adancime
  – poate gasi o solutie rapida dar in general nu va fi una optima
- Iterative Deepening search – apeleaza DFS limitat cu o adancime maxima din ce in ce mai mare pana va obtine prima solutie
- Greedy best-first search – se foloseste de o euristica pentru a determina care dintre starile viitoare este cea mai profitabila
  – va gasi solutia cea mai satisfacatoare conform functiei euristice
  – depinde de functie daca solutia va fi sau nu optima
  – implementarea mai complexa
- A* – determina cea mai buna stare pentru tranzitie folosind atat o euristica cat si o functie care estimeaza costul pentru parcurgerea drumului pana in acea stare
  – timp de executie si complexitate ridicate
  – solutia obtinuta este satisfacatoare in cele mai multe cazuri
- Hill-Climbing search – parcurge spatiul de stari pana la o stare cu un cost mai mic ca cele viitoare, obtine primul segment dintr-o solutie posibil optima
  (algoritmul doreste sa maximizeze costul care este format din o euristica si functia de cost al tranzitiilor din starea de star pana in cea curenta, functia din urma va creste de la o stare la alta iar algoritmul se va opri repede fara a gasi o solutie problemei)

Efecte ale optimizarii:

- Optimizare: Prevenirea revizitarii unei stari
- Timpul de executie a scazut drastic dupa momentul aplicarii acesteia pentru toate strategiile de cautare, mai putin cele ce implicau DFS

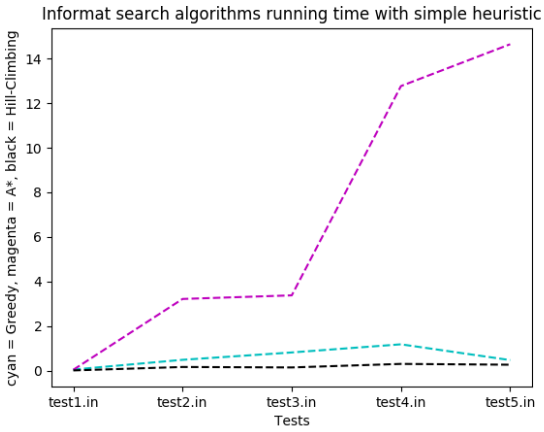
Functii euristice:

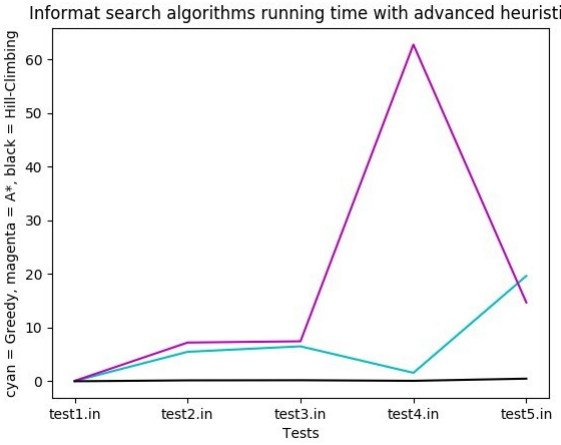
- Functia simple_h are ca scop alegerea actiunilor de tip dropoff si pickup, apoi va prioritiza tranzitiile catre nord, est, sud si vest, asadar taxiul va tinde sa se miste in deosebi spre nord si est

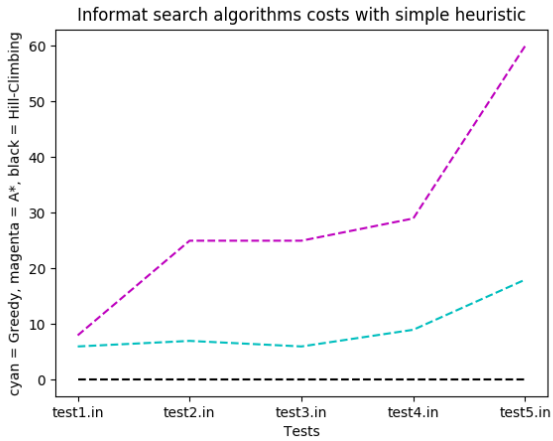
- Functia h prioritizeaza actiunile de dropoff si pickup, apoi considera ideale tranzitiile in care taxiul este angajat intr-o cursa, in caz ca nu are client se va indrepta catre cel mai apropiat

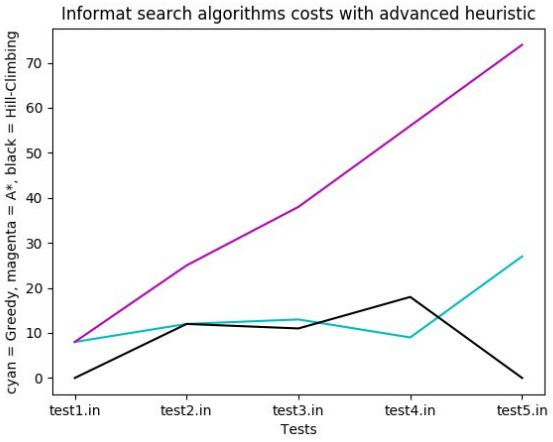
PS: este ora 22:52, am terminat README-ul si am aranjat graficele frumos, acum am observat ca exista si testul test6.in, nu mai am timp sa refac README-ul, voi reface graficele si le voi lasa in arhiva, super sorry :D

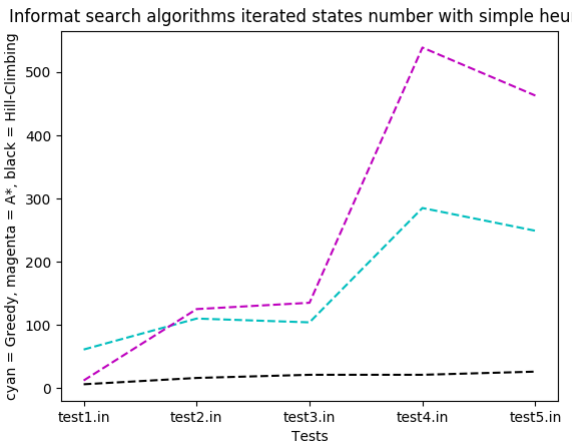
Rezultate si grafice pentru strategiile informate

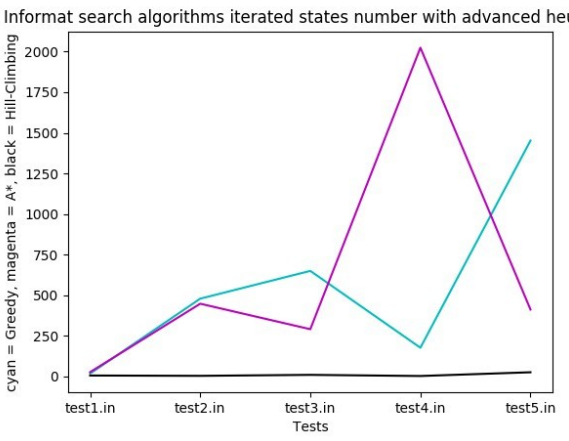












Rezultate si grafice pentru strategiile neinformate