Tema seminar 5

Ex 1:

Exemplificati algoritmul A* si faceti o comparatie intre el si Dijkstra.

Framework-ul de gandire pentru A*:
Teoretic A* returneaza calea optima obtinuta si in contextul Dijkstra numai ca foloseste ceva in plus care ne ajuta sa muncim mai putin pe parte de cautare a path-ului optim.

Mai exact alg. A* are in plus fata de Dijkstra, un estimator *(aka "a heuristic")* (intre o anume stare sau un anume nod din graf si starea target sau nodul destinatie), deci incearca sa aduca o informatie in plus in relatie cu nodul target, dincolo de ponderile existente in graf.

Acesta "heuristica" este ca un cost aditional intre un nod anume si nodul target care e masurat altfel (distantele euclidiana, manhattan sunt cateva dintre posibilitatile de masurare). Totodata ea nu ar trebui sa supraestimeze costul dat in mod normal prin suma ponderilor intre acel nod si nodul destinatie. Fie il poate subestima sau poate fi exact valoarea data de suma ponderilor, altfel A* nu ar putea returna calea optima.

Cum functioneaza A*:
Pe parte de cost search (deci pornim cu contextul de cautare din Dijkstra), A* va folosi si acea "heuristica" in sensul ca adunam la costul path- ului (sumaa ponderilor) pana la acel nod (parte comuna cu Dijkstra) si informatia in plus ("heuristica") legata de costul estimat al acelui nod fata de nodul target, (lucru specific in A*).
Acesta suma ne va da practic un estimator final al nodului in cauza in relatie cu nodul destinatie (sau un scor A* al path-ului acelui nod in relatie cu nodul destinatie).

Asadar avem graful urmator:

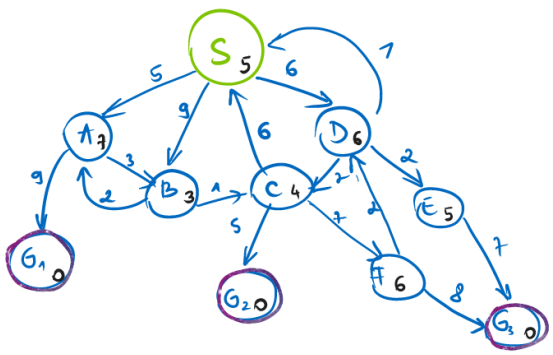

Figure 1: Graful initial ponderat avand si costul heuristic asociat fiecarui nod
Sursa figura: prelucrare proprie

Cu observatiile:

- Ne propunem sa vedem calea optima fata de oricare din nodurile detinatie si odata ajunsi.
- Am considerat mai multe noduri destinatie (costul euristic per fiecare nod din graf fiind vazut in relatie cu oricare din nodurile target, sub conditia celor mentionate mai sus, astfel incat optimul prin A* sa poate fi vazut). De exemplu, costul euristic (**scris in negru**) din D, adica 6, nu putea fi mai mare decat suma ponderilor dintre D si oricare dintre nodurile target (cost path-ul lui D).
- De altfel costul euristic al nodurilor target este 0 fiindca suntem deja la nivelul targetului si algoritmul se va opri cu parcurgerea pe acea cale.

Descriere pasi pentru A*:

  - Avem o coada cu prioritati de noduri vizitate si pornim cu nodul S (start) care va fi marcat ca fiind si vizitat avand asociat costul sau euristic.
  - Costul total euristic recalculat cu fiecare inaintare de nod este scris in rosu.(e cost path-ul pana la acel nod + euristica nodului).

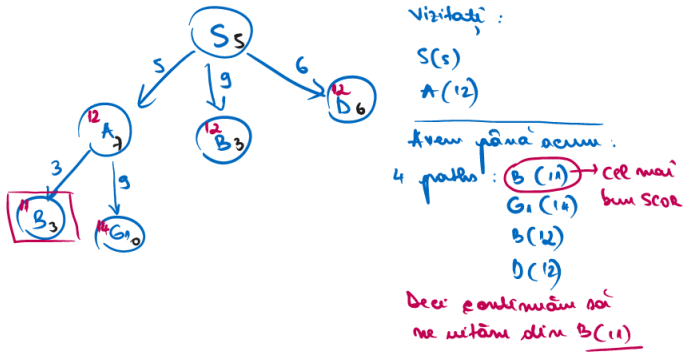

Figure 2: Graful parcurs pana la vecinii nodului A

  - din B(11) putem merge in A dar ne va da un cost estimat de 13 (cost path-ul + euristica = (5 + 3 + 2) + 3 care este mai mare decat ceea ce avem deja pe vizitati si nu are sens sa il luam in calcul si atunci mergem pe C.
  - Mergem mai departe cu nodul B care are 9 + 3 = 12 cost estimat dar a fost deja vizitat cu un cost estimat de 11 deci nu are sens si va fi inchis acest path.
  - Ne uitam la D si la nodurile catre care putem ajuneg de aici. Nu ne intoarcem in S fiindca nu are sens fiind deja vizitat mai optim. Dar putem explora C si E.

  - Vezi in figura 4.
  - Din C are sens sa exploram G2 si F.Avem pe C adaugat in vizitati si vedem caile posibile in acest punct dein C(13), G2(15), F(21) si E(13). C va fi un nod inchis fiind vizitat mai optim si mergem pe vizitarea lui E iar din E putem ajunge in G3.
  - In acest moment avem toate path- urile iar cel optim se atinge in G2 avand costul estimat minim.
  - putem reface path-ul tinand cont de un label de predecesori asociat ficarui nod astfel incat sa afisam si path-ul otpim de noduri.

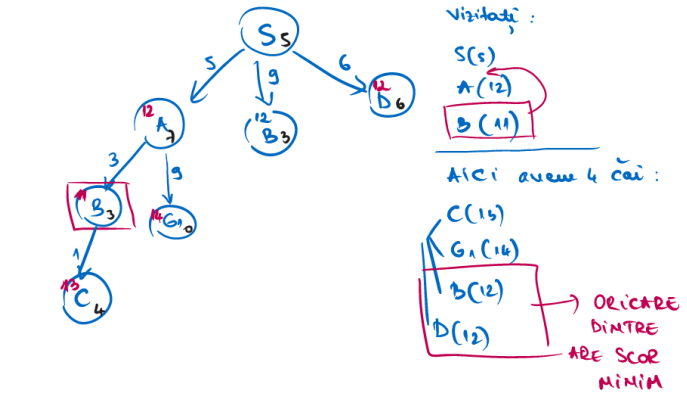

Figure 3: Graful parcurs pentru nodurile B si D

– O observatie aici este legata de euristica lui lui S care pare sa nu fi fost chiar in regula fiind 5 iar path-ul de cost minim a dat 13.
– Deci putem regandi costul nodului S astfel incat algoritmul sa vada mai rapid calea.

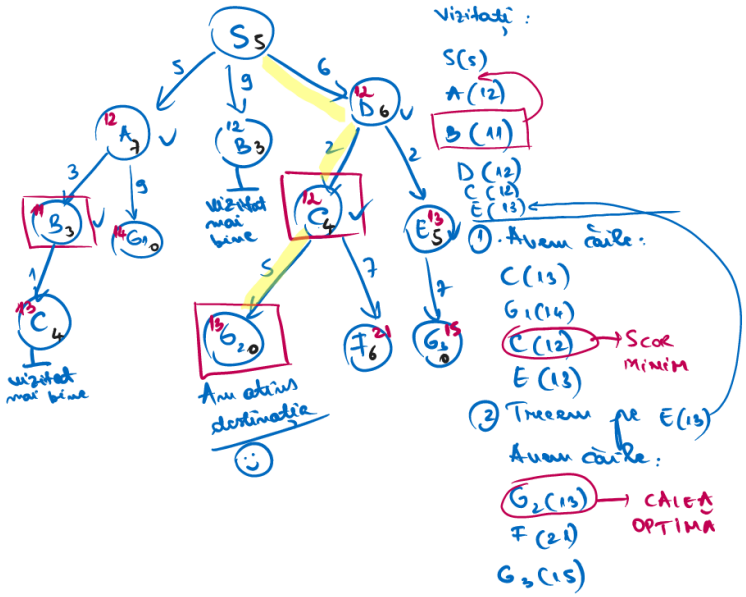

Figure 4: Graful parcurs pentru descoperirea caii optime
Sursa figura: prelucrare proprie