Search Algorithms in AI - A Search*

Egzonit Demhasaj
Fakulteti i Shkencave Matematiko-Natyrore
Drejtimi: Shkenca Kompjuterike
Universiteti i Prishtinës
Deçan, Kosovë 51000
egzonit.demhasaj@student.uni-pr.edu

Gabriel Kolaj
Fakulteti i Shkencave Matematiko-Natyrore
Drejtimi: Shkenca Kompjuterike
Universiteti i Prishtinës
Klinë, Kosovë 32000
gabriel.kolaj@student.uni-pr.edu

***Abstrakt*— Për të përafruar rrugën më të shkurtër në situatat e jetës reale, si në harta, lojëra ku mund të ketë shumë pengesa, ne mund të konsiderojmë një rrjetë 2-dimensionale që ka disa pengesa dhe ne fillojmë nga një qelizë burimore (me ngjyrë të kuqe) për të arritur drejt një qelize qëllimi (me ngjyrë të gjelbërt). Ekzistojnë algoritme të ndryshme të kërkimit që na ndihmojnë në lidhje me Path Finding, e cila ka gjetur një aplikim të madh në kuadër të fushës së Inteligjencës Artificiale. Ne mund të gjejmë praktikisht rrugën më optimale nga një burim në një destinacion duke shtuar kosto të cilat do të përfaqësonin kohën, paratë etj. A* është një nga algoritmet më të njohura për të gjitha arsyet e duhura. Në këtë punim, le të zbulojmë arsyejen pse.**

Fjalët Kyçe: Inteligjenca Artificiale (ang. Artificial Intelligence), Algoritmet e Kërkimit (ang. Search Algorithms), Algoritmet e Painformuara (ang. Uninformed Search Algorithms), Nyja (ang. Node), Funksioni i Vlerësimit (ang. Evaluation Function), Heuristic.

I. Çfarë është Algoritmi i Kërkimit ??

Të lëvizim nga njëri vend në tjetrin është një detyrë që ne njerëzit e bëjmë pothuajse çdo ditë. Ne përpiqemi të gjejmë rrugën më të shkurtër që na mundëson të arrijmë destinacionet tona më shpejtë dhe ta bëjmë të gjithë procesin e udhëtimit sa më efikas që të jetë e mundur. Në kohët e vjetra, testonim dhe gabonim me rrugët që kishim në dispozicion dhe duhej të mendonim se cila rrugë ishte më e shkurtër ose më e gjatë.

Tani, kemi algoritme që mund të na ndihmojnë të gjejmë rrugët më të shkurtra praktikisht. Ne vetëm duhet të shtojmë kosto (**kohë**, **para** etj.) në grafikët ose hartat dhe algoritmi na gjen rrugën që duhet të marrim për të arritur destinacionin tonë sa më shpejt të jetë e mundur. Shumë algoritme u zhvilluan gjatë viteve për këtë problem dhe **A*** është një nga këto algoritmet më popullore.

II. A* Search

A* Search është një algoritëm i njohur pathfinding i përdorur në një gamë të gjerë aplikimesh, nga videolojërat tek robotikët. A* Search është një algoritëm i informuar i kërkimit (ang. Informed Search Algorithms), ku përpos njohurive nga definimi i problemit, ky algoritëm ka edhe njohuri specifikie në lidhje me problemin e kërkimit. Ky algoritëm kombinon karakteristikat e **Best First Search** dhe **Greedy Best First Alogrithm**, për të na treguar rrugën më efikase se si të mbërrijmë nga nyja fillestare (ang. initial state) tek nyja e destinacionit. A* Search, ashtu sikurse Best First Search, funksionon në bazë të një funksioni të vlerësimit (ang. **Evaluation Function**) **f(x)**, i cili është shuma e dy funksioneve të tjera, të cilat po i shënojmë me **g(x)** dhe **h(x)**. g(x)- na paraqet koston (mund të reprezentojë kohën, gjatësinë e rrugës, paratë ose diçka të tillë) e rrugës nga nyja ku jemi aktualisht deri tek nyja e radhës, kurse h(x)- na paraqet një funksion heuristik, i cili na tregon se sa larg nga nyja e destinacionit jemi, nga nyja ku ndoshemi aktualisht. Pra, mund të themi se A* Search është pak a shumë një variant i ngjashëm me **Dijkstra's Algorithm** (Uniform Cost Search), vetëm se A* Search përdorë edhe funksioni heuristik h(x), përpos funksionit g(x).

Që algoritmi të gjenerojë rezultatin e saktë, funksioni i vlerësimit duhet të jetë **admissible**, që do të thotë se ai kurrë nuk e mbivlerëson koston për të arritur në nyjen e qëllimit.

A* zgjeron rrugët që tashmë janë më pak të shtrenjta duke përdorur këtë funksion:

$$f(x) = g(x) + h(x)$$

ku :

- **f(x)** : kostoja totale e përllogaritur e rrugës nga nyja e fillimit deri në nyjen e destinacionit ;
- **g(x)** : kostoja për të arritur deri në nyjen e radhës x ;
- **h(x)** : kostoja për të arritur deri në nyjen e destinacionit nga nyja ku ndodhemi aktualisht x.

III. Pse A* Search Algorithm ??

Pra, çfarë është saktësisht algoritmi A*? Është një algoritëm i avancuar BFS që kërkon së pari rrugët më të shkurtra se sa rrugët më të gjata. A* është optimal si dhe një algoritëm i plotë (ang. complete algorithm).

Çfarë dua të them me **Optimal** dhe **Complete**? Optimale nënkupton se A* është e sigurt se do të gjejë koston më të vogël nga burimi deri në destinacion, kurse Complete do të thotë se do të gjejë të gjitha shtigjet që janë në dispozicion për ne nga burimi deri në destinacion dhe nëse ekziston një zgjidhje, ai do të na e gjejë atë.

Pra kjo e bën A* algoritmin më të mirë apo jo? Epo, në shumicën e rasteve, po. Por A* është i ngadalshëm dhe gjithashtu hapësira memorike që kërkon është shumë e madhe, pasi që kursen të gjitha shtigjet e mundshme që janë në dispozicion për ne. Kjo bën që algoritmet e tjera më të shpejta të kenë një përparësi mbi A* por megjithatë është një nga algoritmet më të mira të kërkimit.

Atëherë, pse të zgjedhim A* në vend të algoritmeve të tjera më të shpejta?

Le t'ju përgjigjen grafikët e mëposhtëm. Kemi marrë Algoritmin e Dijkstra-s dhe algoritmin A* për krahasim.

Këtu mund të shihni se Algoritmi i Dijkstra-s gjen të gjitha rrugët që mund të merren pa gjetur apo ditur se cila është më optimale për problemin me të cilin po përballemi. Kjo rezulton në punën e paoptimizuar të algoritmit dhe llogaritjeve të panevojshme.

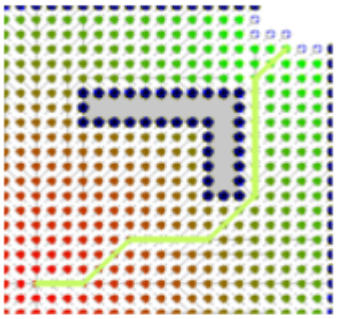

Fig.1. Algoritmi i Dijkstra-s

Nga ana tjetër, algoritmi A* gjen rrugën më optimale që mund të marrë nga burimi në arritjen e destinacionit. Ajo e di se cila është rruga më e mirë që mund të merret nga gjendja e tanishme dhe si duhet të arrijë në destinacion.

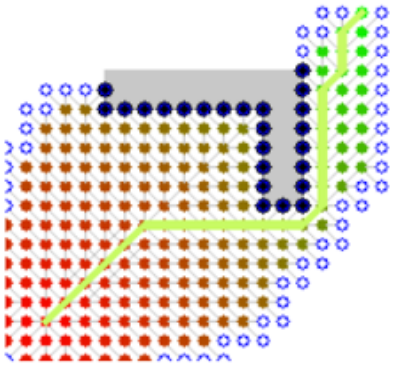

Fig.2. Algoritmi A*

IV. Demonstrimi i A* me një shembull

Tani që ju e dini pse ne zgjedhim A*, le të kuptojmë pak nga teoria rreth saj pasi është thelbësore për t'ju ndihmuar të kuptoni se si funksionon ky algoritëm.

A*, siç e dimë të gjithë tani, përdoret për të gjetur rrugën më optimale nga një burim në një destinacion. Ajo optimizon rrugën duke llogaritur distancën më të vogël nga njëra nyje në tjetrën.

Ekziston një formulë, të cilën e kemi përmendur edhe më herët ku e kemi quajtur **Funksioni Vlerësues** (ang. **Evaluation Function**), që të gjithë ne duhet ta mbajmë në mend pasi është zemra dhe shpirti i algoritmit.

$$f(x) = g(x) + h(x)$$

Rolin e gjithsecilit parametër të funksionit f(x) e kemi kuptuar më herët, si dhe e dijmë arsyejen se pse algoritmi e përdorë këtë funksion vlerësues. Pasi që tani e kemi kuptuar funksionimin e algoritmit A* sipas f(x) ne mund të marrim një shembull dhe ta demonstrojmë atë. Kështu që le të marrim një shembull shumë të thjeshtë :

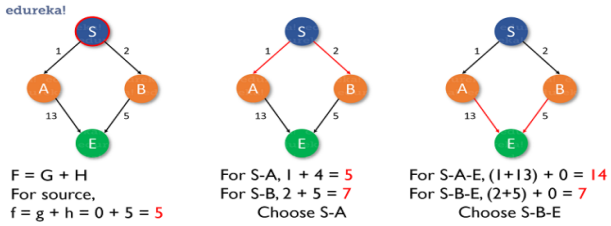

Fig.3. Demonstrimi i zgjidhjes së një problemi me anë të A*

Supozojmë se kemi një grafik të vogël me nyjet: S, A, B, E ku **burimi** është **S** dhe **E** është **destinacioni**.

Mos harro se kostoja për të hyrë në burim dhe në destinacion është gjithmonë **0**.

Vlerat heuristike janë:

S -> 5, A -> 4, B-> 5 dhe E-> 0

Le të përdorim formulën dhe të llogarisim rrugën më të shkurtër nga burimi në destinacion tani.

f = g + h, ku **g** është kosto për të udhëtuar dhe **h** është vlera heuristike.

Për të arritur burimin:

f(S) = 0 + 5 = 5

Rrugët nga S në nyjet e tjera janë:

f(S->A) = 1 + 4 = 5

f(S->B) = 2 + 5 = 7

Pra, ne së pari do të zgjedhim rrugën e **S -> A**, pasi ajo ka koston, përkatësisht vlerën e funksionit vlerësues **f(x)** e ka më të vogël, e pastaj vijojmë rrugën tonë për në destinacion.

Rrugët nga A dhe B në Destinacion:

f(S->A->E) = (1 + 13) + 0 = 14

f(S->B->E) = (2 + 5) + 0 = 7

Pas llogaritjes, tani kemi gjetur se **B** më vonë na ka dhënë rrugën më të shkurtër. Pra, ne ndryshojmë rrugën tonë më të shkurtër për në **S->B->E** dhe kemi arritur destinacionin tonë. Kështu e përdorim formulën për të gjetur rrugën më optimale.

Shpresojmë që demonstrimi i algoritmit A* me këtë shembull kaq të thjeshtë ta bëjë atë më të qartë tek lexuesi.

V. Pseudokodi (Algortimi) i A* Search

Me të kuptuar përdorimin e funksionit vlerësues, le t'i hedhim një sy se si funksionon algoritmi i A* Search:

Së pari krijoni 2 lista të cilat do t'ju ndihmojnë të kuptoni rrugën, le t'i quajmë ato **opened list** dhe **closed list** :

Input: Nje grafë G(V, E) me nyjen fillestare dhe nyjen ku dëshirojmë të shkojmë.

Output: Rruga që kushton më së paku nga fillimi deri tek mbarimi.

Hapat:

```
Inicializo
        open_list = { start }
        clost_list = { }
        g(fillimi) = 0
        h(start) = funksioni_heuristik (start, end)
        f(start) = g(start) + f(start)
while open_list is not empty
        m = Node on to of open_list, with least f
        if m == end
                return
        remove m from open_list
        add m to closed_list
for each n in child(m)
                if n in closed_list
                        continue
cost = g(m) + distance(m, n)
if n in open_list and cost < g(n)
remove n from open_list as new path is better
if n in closed_list and cost < g(n)
        remove n from closed_list
if n not in open_list and n not in closed_list
        add n to open_list
g(n) = cost
        h(n) funksioni heuristik (n, mbarimi)
        f(n) = g(n) = h(n)      return failure
```

VI. REFERENCAT

[1] What is the A* Algorithm and How does it work? Last updated on Nov 25,2020 - edureka!

[2] Patel, A. *Introduction to A**. Retrieved April 29, 2016, from http://theory.stanford.edu/~amitp/GameProgramming/concave1.png

[3] Patel, A. Introduction to A*. Retrieved April 29, 2016, from http://theory.stanford.edu/~amitp/GameProgramming/concave2.png

[4] A* Search. Brilliant.org. Retrieved 15:44, April 19, 2023, from https://brilliant.org/wiki/a-star-search/

[5] A*Search. Code Academy. Hisham Touma. Retrieved 15:44, April 19, 2023, from AI | Search Algorithms | A* Search | Codecademy

[6] A* Search Algorithm. GeekforGeeks. Retrieved 15:44, April 19, 2023, from A* Search Algorithm - GeeksforGeeks.