

SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET

PROJEKT (UVOD U UMJETNU INTELIGENCIJU)
LJUBAVNI LABIRINT

Student:

Ivona Mustapić-Jogun

Split, veljača, 2018.

SADRŽAJ

UVOD.....	3
OPĆENITO O LJUBAVNOM LABIRINTU	4
PRETRAŽIVANJA	5
ITERATIVNA PRETRAGA PO DUBINI	5
PRETRAGA PO ŠIRINI	5
PRETRAGA PO DUBINI.....	5
POHLEPNI ALGORITAM	6
A* ALGORITAM	6
ALGORITAM JEDNOLIKOG TROŠKA (DIJKSTRA).....	6
ALGORITAM VALNE FRONTE	6
TABU ALGORITAM.....	7
ALGORITAM STOHAISTIČKOG PENJANJA UZBRDO	7
ZAKLJUČAK	8
LITERATURA	12

UVOD

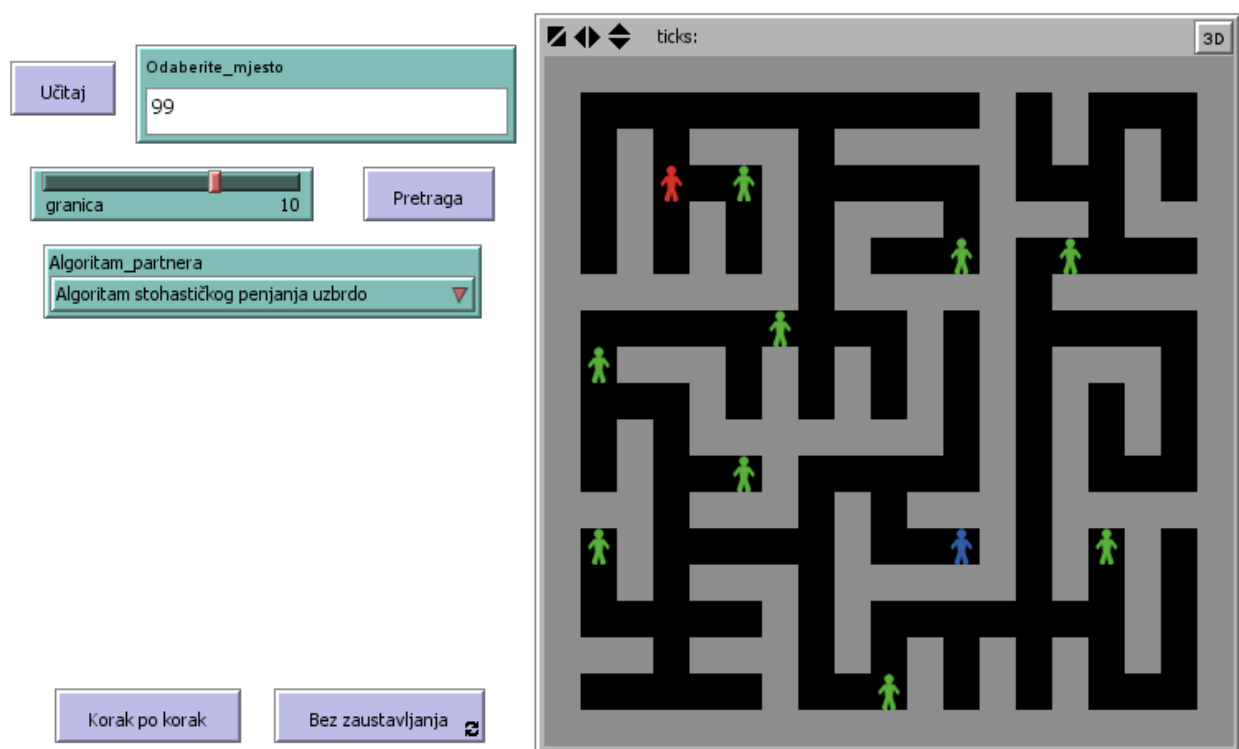
Inteligencija je vrlo općenita mentalna sposobnost, koja između ostalog, uključuje sposobnost zaključivanja, planiranja, rješavanja problema, apstraktno mišljenje, razumijevanje kompleksnih ideja, brzo učenje i učenje na temelju iskustva. Ne obuhvaća samo učenje iz knjiga, usku akademsku vještinu ili elegantno rješavanje testova. Inteligencija reflektira širu i dublju sposobnost razumijevanja našeg okruženja – opažanja, shvaćanja smisla u stvarima ili odlučivanja o tome što napraviti.

Umjetna inteligencija je područje koje je posvećeno proučavanju računalnog modela inteligentnog ponašanja. Zajedničko svim područjima umjetne inteligencije je izrada agenata ili strojeva koji imaju odlike inteligentnog ponašanja; rješavanje problema, predstavljanje znanja, zaključivanje, učenje, percepcija i interpretiranje.

Inteligentni agenti sadrže određeni stupanj inteligencije (razinu zaključivanja i učenja odnosno sposobnosti prihvaćanja korisnikovih izjava o ciljevima i obavljanje zadataka koji se pred njih postave). Inteligentni agenti su agenti koji prate naše ponašanje te su temeljem njega u stanju naučiti kako mi rješavamo probleme i slično. Kad su to naučili oni mogu takve zadatke obavljati samostalno uz povremene provjere vlasnika kod pojave novih ili nejasnih situacija.

OPĆENITO O LJUBAVNOM LABIRINTU

Ljubavni labirint je „igra“ koja Vam omogućava pogled u budućnost. Precizno određuje hoćete li u životu pronaći ljubav svog života ili pak u tome nećete biti uspješni. Upisivanjem broja 1-100 u input „Odaberite_mjesto“ birate lokaciju. Na button „Učitaj“ pojavljuje se odabrana lokacija te vi u uložite crvene osobe, Vaš partner u uložite plave osobe i određen broj Vaših „neprijatelja“ koji Vam smanjuju živote ukoliko se sretnete. Imate ukupno 3 živote, a potreban Vam je barem 1 ako želite da Vas partner pronađe. Chooser „Algoritam_partnera“ Vam daje mogućnost da Vi odaberete algoritam pretraživanja, a button „Pretraga“ koristi odabir na sreću (slučajan odabir). Slider „granica“ služi za odabir godina koje ste spremni izdvojiti za traženje partnera (dubina kod ograničene pretrage po dubini). Nakon svih odabranih stavki možete pritisnuti button „Bez zaustavljanja“ kako bi odmah vidjeli svoju sudbinu ili možete pritisnuti na button „Korak po korak“ kako bi vidjeli i put do partnera, ukoliko Vas je pronašao. Na kraju će se pojaviti ili poruka „Pronasli ste ljubav svog života!“ ili poruka „Niste pronašli ljubav svog života!“.



PRETRAŽIVANJA

Tehnike pretraživanja u računalnoj znanosti su strategije koje traže rješenje u prostoru pretrage. Rješenja ili ciljna stanja mogu ponekad biti objekti, ciljevi, pod-ciljevi ili put do tražene točke. Pretraživanje je postupak rješavanja zadataka umjetne inteligencije. Pretraživanje uključuje traženje puta od početnog stanja do ciljnih stanja kroz prostor stanja unutar kojeg se zadatak rješava.

U ovom projektu je korišteno 8 pretraživanja: iterativna pretraga po dubini, pretraga po širini, pretraga po dubini, pohlepni algoritam, A* algoritam, algoritam jednolikog troška (Dijkstra), ograničena pretraga po dubini, algoritam valne fronte, tabu algoritam i algoritam stohastičkog penjanja uzbrdo.

ITERATIVNA PRETRAGA PO DUBINI

Pretraživanje u dubinu koje u prvom koraku ide do dubine 1, a ako rješenje nije nađeno odbacuju se čvorovi i započinje ponovno pretraživanje u drugoj iteraciji do dubine 2. Ako rješenje i tada nije nađeno započinje ponovno pretraživanje u dubinu do dubine 3 itd. slijedno do neke granične dubine n . Algoritam je u jednoj iteraciji identičan algoritmu za pretraživanje u dubinu. Taj se algoritam prekida kada je postignuta dubina d , ako cilj nije pronađen. Uklanjaju se svi čvorovi iz reda, d se povećava za jedan i postupak ponovno započinje.

PRETRAGA PO ŠIRINI

Metoda pretraživanja u širinu (eng. breadth-first search) obilazi čvorove grafa redoslijedom kojim su generirani iz čvora prethodnika, odnosno koristi se strategija u kojoj stariji čvorovi imaju prednost. To znači da će se prvo ispitati svi čvorovi jedne razine, a tek će se nakon toga prijeći na ispitivanje čvorova na sljedećoj razini. Najprije se ispituje početni čvor. Ukoliko on nije ujedno i ciljni, generiraju se svi njegovi sljedbenici te se za svakog provjerava je li upravo taj čvor, ciljni čvor. Ne pronađe li se ciljni čvor, postupak se nastavlja za svaki čvor na sljedećoj razini.

PRETRAGA PO DUBINI

Za razliku od pretraživanja u širinu, metoda pretraživanja u dubinu (eng. depth-first search) ne obilazi čvorove po razinama, nego najprije obilazi sve sljedbenike nekog čvora, a tek nakon što se dođe do dna grafa, odnosno do čvora koji više nema nasljednika, pretraživanje se usmjerava na sljedeći čvor na istoj razini. Ovdje se prednost daje čvorovima koji su generirani u najnovije vrijeme. Kod ove je metode potrebna opreznost jer postoji mogućnost postojanja zatvorenih petlji u grafu, pa bi se moglo dogoditi da se pretraživanje zavrti u takvoj petlji. Naime, moguć je scenarij u kojem je nasljednik nekog čvora čvor koji je već ispitan negdje na višoj razini. Pretraživanje bi se u tom slučaju usmjerilo na već posjećeni čvor, a algoritam bi se ponašao kao da se otišlo razinu niže. Jasno je da će se postupak periodički ponavljati (u terminima grafova ovo se svojstvo naziva svojstvo napuhavanja), te nikad neće dovesti do rješenja. Jedno od mogućih rješenja upadanja u zamke ovakvog tipa je ograničavanje dubine do koje će algoritam

pretraživati čvorove. Kad se dostigne ograničavajuća dubina, pretraživanje se usmjerava na čvor na koji bi se usmjerilo u slučaju da je na tom mjestu stvarno bio posljednji čvor u grani grafa.

POHLEPNI ALGORITAM

Pohlepno pretraživanje (*greedy search*) je slično metodi uspona na vrh, samo što ne napredujemo sistematski dok ne dosegneмо maksimalnu dubinu, već idemo u onaj čvor koji od svih otvorenih čvorova ima najmanju vrijednost heurističke funkcije. Kod metode uspona na vrh, najprije se poslože svi novi čvorovi koji se otvaraju i izabere onaj sa najmanjom vrijednosti koja se dodaje u red na početak reda. Kod pohlepnog traženja nakon što se takav čvor dodaje poslože se svi članovi reda i uzme onaj koji ima najmanju vrijednost heurističke funkcije. Ovakvo pretraživanje spada u postupke najboljeg prvog pretraživanja (Best First Search).

A* ALGORITAM

Kombinacija dosadašnjih algoritama sa dodatkom brisanja određenih putova koji ne zadovoljavaju određene uvjete. Postupak zovemo „brisanje redundantnih puteva“ (*redundant path deletion*) i on ovisi samo o akumuliranoj cijeni, a ne o estimiranim vrijednostima dolaska do cilja. Algoritam je skoro isti kao algoritam jednolikog troška proširen estamacijom, osim što se dodaje dio brisanja određenih putova. Drugim riječima kazano, A* algoritam izbjegava skupe puteve.

ALGORITAM JEDNOLIKOG TROŠKA (DIJKSTRA)

Na svakom koraku širi se čvor koji ima najmanju akumuliranu cijenu (zbroy onoga što se do sada potrošilo – zbroj prijeđenih kilometara do čvora u kojem se trenutno nalazimo).

OGRANIČENA PRETRAGA PO DUBINI

Pretraživanje u dubinu po principima pretraživanja u dubinu, ali samo do određene dubine, a nakon toga se ide u sljedeću granu do iste dubine. Algoritam je u jednoj iteraciji identičan algoritmu za pretraživanje u dubinu. Taj se algoritam prekida kada je postignuta dubina d , ako cilj nije pronađen. Uklanjaju se svi čvorovi iz reda, d se povećava za jedan i postupak ponovno započinje.

ALGORITAM VALNE FRONTE

Algoritam valne fronte je jedan od osnovnih, ali i moćnijih algoritama. Može se podijeliti u tri glavna koraka. To su stvaranje diskretne mape prostora, ispunjavanje matrice i pomicanje agenta do cilja. Prvi korak je stvaranje diskretne mape u kojoj se agent kreće. Na toj mapi označujemo sav prazan prostor, prepreke koje se nalaze u prostoru, početnu i ciljnu poziciju. U drugom koraku moramo ispuniti matricu kako bi imali put od agenta do cilja. Matrica se ispunjava tako da počnemo od cilja kojeg označimo s vrijednosti 1, a susjedna polja s 2. Koja su mu susjedna polja ovisi o tipu susjedstva u okruženju.

TABU ALGORITAM

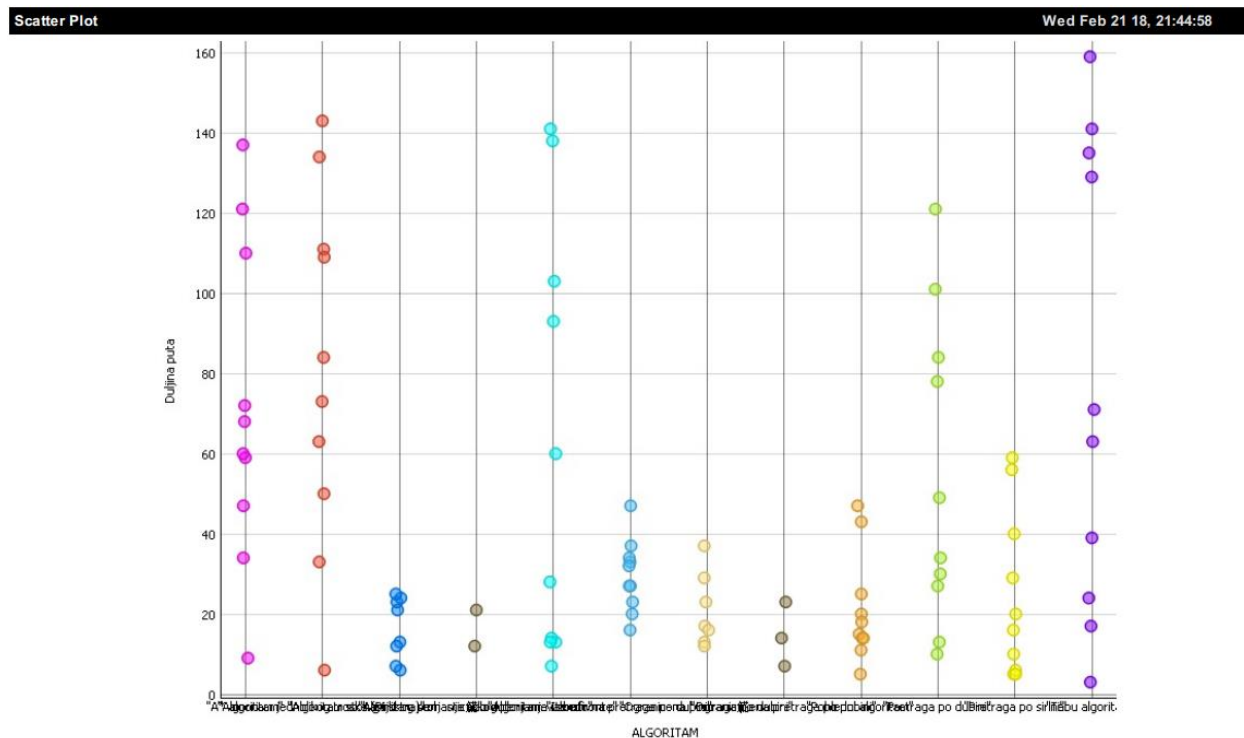
Osnovna ideja ovog tipa heurističkog algoritma je zamijeniti dopustivo rješenje X novim, najboljim rješenjem Y iz njegove okoline, koje je također dopustivo. Treba biti pažljiv jer moglo bi se dogoditi da u sljedećem koraku zamijenimo Y s X pa bismo se počeli vrtjeti u krug, što nikako ne želimo. Zato se izgrađuje "tabu-lista" u kojoj se nalaze neka rješenja koja algoritam ne smije odabrati u sljedećem koraku. U tabu-listi ne nalaze se sva dopustiva rješenja koja je algoritam u nekom koraku posjetio, već samo ona posjećena u zadnjih L iteracija (uobičajeno je $L = 10$). Za potpuno onemogućavanje ciklusa, L bi trebao biti jako velik, no tada ne bi bilo mnogo dozvoljenih odabira i bilo bi ih teško naći. Time bi se program usporio, što je u suprotnosti s onim što se heurističkim algoritmima nastoji postići: dopušta se lošiji rezultat, ali se omogućava veća brzina izvođenja programa.

ALGORITAM STOHAISTIČKOG PENJANJA UZBRDO

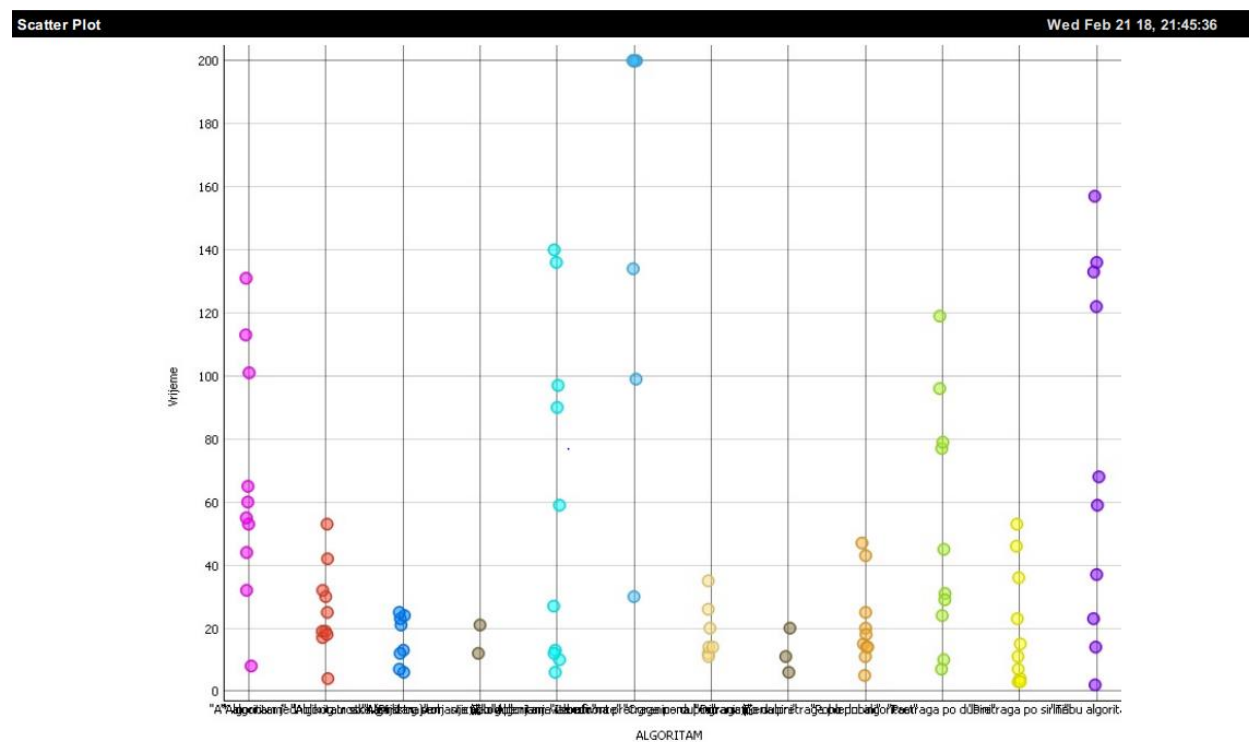
Strategija algoritma stohastičkog penjanja uzbrdo je biranje slučajnog susjeda za rješenje i njegovo prihvatanje samo ako to dovodi do poboljšanja. Ne bira uvijek susjedno stanje sa najvećom vrijednošću.

ZAKLJUČAK

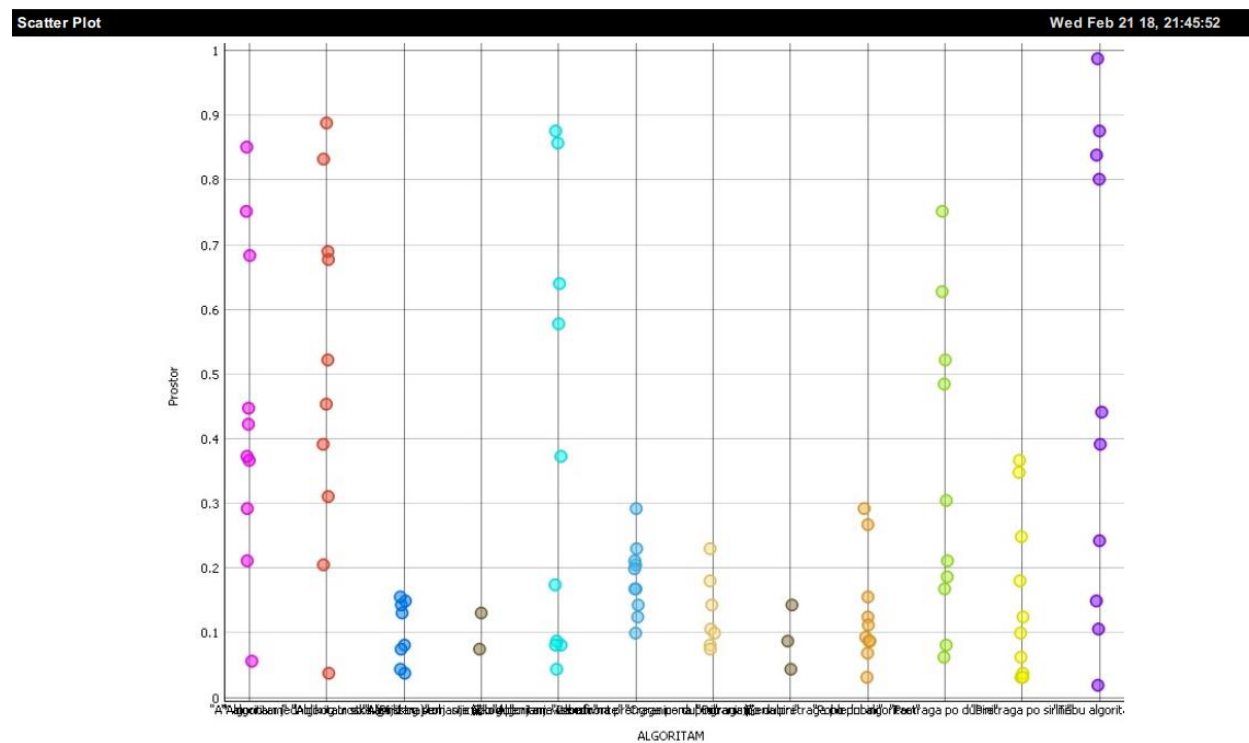
Upotrebljavajući različite algoritme, uočili smo njihove prednosti i mane. Usporedili smo ih pomoću Orange-a te spremili u dokument „izvještaj.pdf“ gdje možemo vidjeti njihovu detaljnu usporedbu. Duljina puta je najveća kod tabu algoritma.



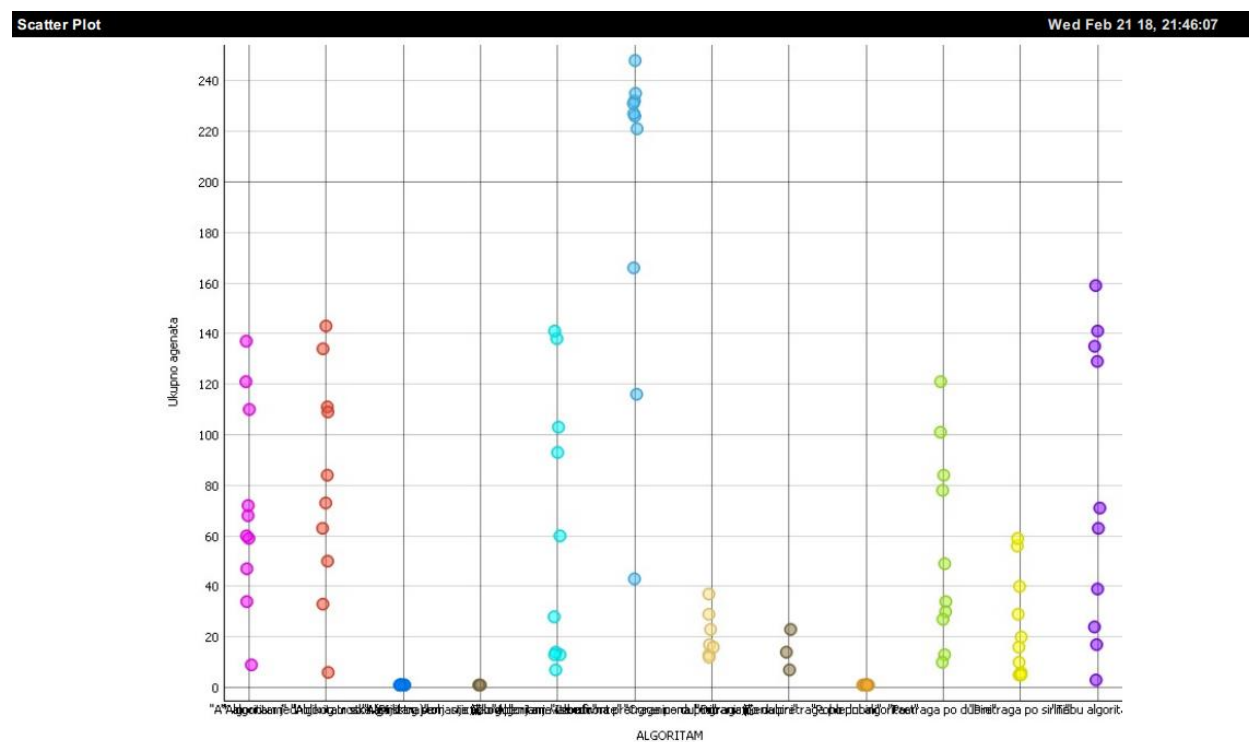
Vrijeme (koraci) je najveće kod iterativne pretrage po dubini.



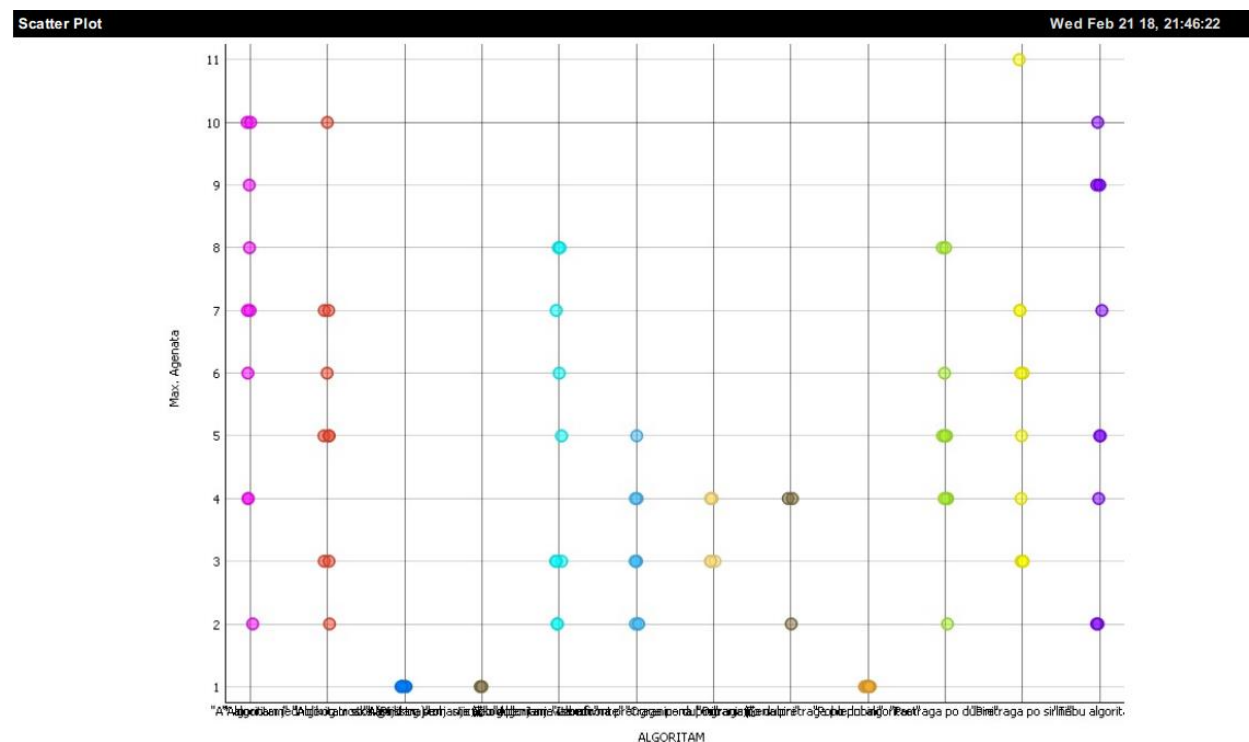
Duljina puta je najveća kod tabu algoritma.



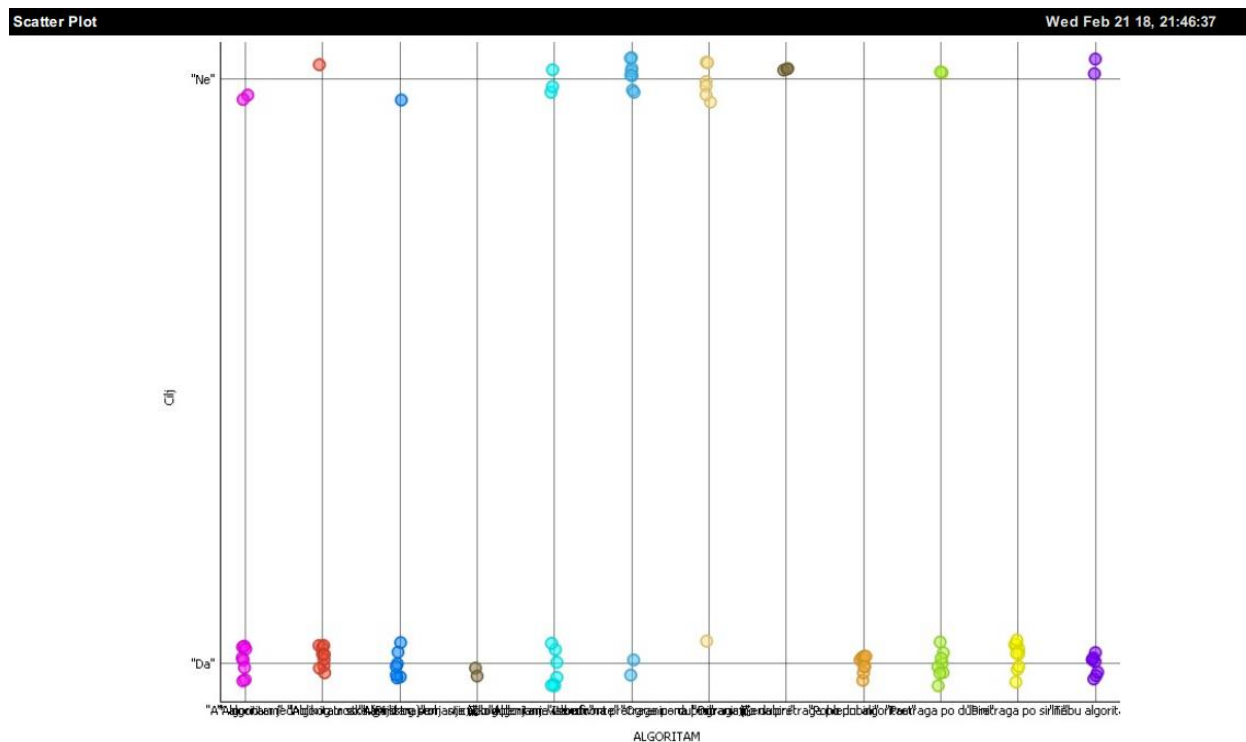
Ukupno agenata je najveće kod iterativne pretrage po dubini.



Maksimalno agenata u određenom vremenu je najveće kod pretrage po širini.



Kod pohlepnog algoritma i kod pretrage po širini momak uvijek dolazi do cure. Stohastičko pretraživanje je razdvojeno u dva stupca (3. i 4.) pa kod njega momak uvijek ne dolazi do cure. Pretraga po dubini je također razdvojena u dva stupca (7. i 8.) pa kod nje nije istina da momak nikad ne dolazi do cilja. Ostali algoritmi (nijedan drugi nije razdvojen na dva dijela) ponekad dođu do cilja, a ponekad ne.



LITERATURA

<https://www.scribd.com/doc/113235454/Uvod-u-umjetnu-inteligenciju-2012>

<http://www.cleveralgorithms.com/nature-inspired/index.html>