

Rješavanje problema usmjeravanja vozila korištenjem neuronskih mreža

Tin Jukić

Opis problema VRP

- Jedan od najčešće rješavanih optimizacijskih problema
- Cilj: posjetiti sve korisnike uz najbolju cijenu
- Rješenje problema: put kojim vozila moraju obići korisnike
- Upotreba: dostavne službe
- Sastoji se od:
 - Čvorova – čvor izvorište (jedan ili više njih) i čvorovi korisnici
 - Vozila (jednakih svojstava)
- Varijante problema VRP:
 - VRP s vremenskim prozorom (VRPTW)
 - VRP s ograničenim kapacitetom vozila (CVRP)
 - VRP s više izvorišta

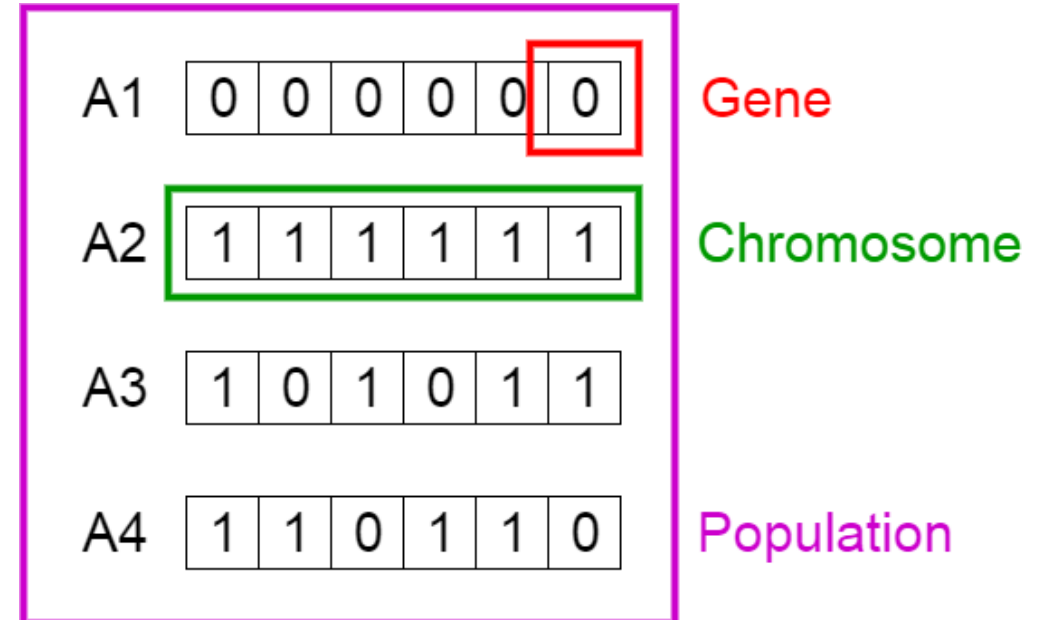
Korištena varijanta VRP-a

- CVRP
- Solomonova instanca
- 25 identičnih vozila, svako kapaciteta 200
- Potrebno obići 100 korisnika
- Algoritam pokušava maksimalno iskoristiti kapacitet svakog vozila te pritom prijeći minimalnu udaljenost
- Korištena neuronska mreža učena genetskim algoritmom

Genetski algoritam (GA)

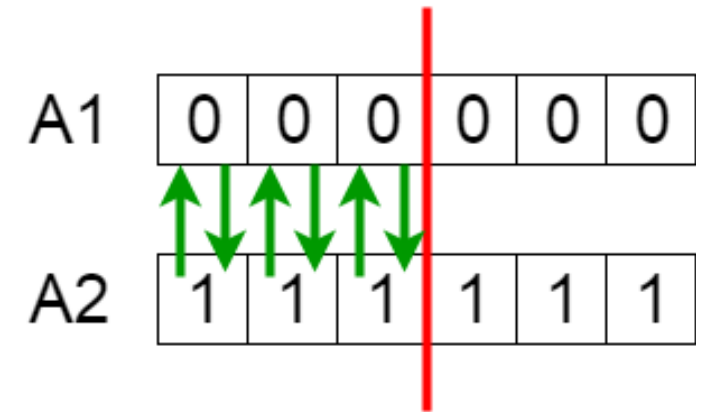
Genetski algoritam (GA)

- Heuristička metoda optimiranja
- Algoritam inspiraciju pronalazi u istraživanju biologa Charlesa Darwina
- Temelj: prirodna selekcija i preživljavanje najboljih jedinki
- Preživjele jedinice daju „potomke” za iduću generaciju
- Temeljna jedinica: jedinka



Genetski algoritam (GA)

- 5 faza:
 1. Početna populacija (nasumično generirana)
 2. Funkcija dobrote – ocjenjuje pojedinu jedinku
 3. Selekcija – uzimaju se najbolje jedinke (najveća vrijednost funkcije dobrote)
 4. Križanje – dolazi do miješanja gena dvaju roditelja i stvaranja dvije nove jedinke za novu populaciju
 5. Mutacija – promjene u genima jedinke (održavanje različitosti unutar populacije i sprječavanje pojave prerane konvergencije)



Before Mutation

A5: 1 1 1 0 0 0

After Mutation

A5: 1 1 0 1 1 0

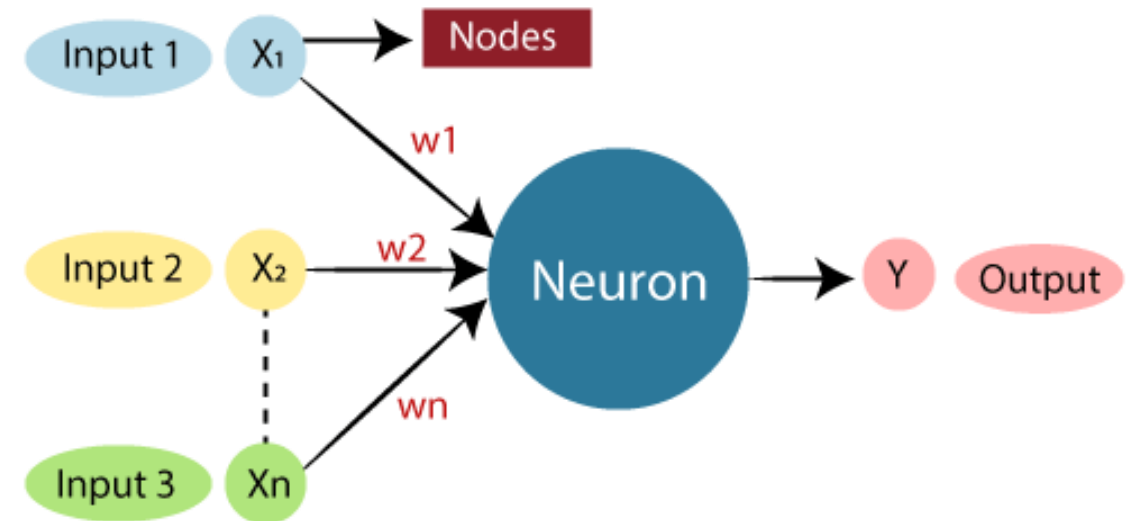
Umjetna neuronska mreža (ANN)

Umjetna neuronska mreža (ANN)

- Model strojnog učenja
- Inspiraciju pronalazi na temelju izgleda i funkcioniranja ljudskog mozga
- Najmanja građevna jedinica: neuron
- Ideja: rješavanje kompleksnog problema se razdvaja na manje probleme, čime se ubrzava njegovo rješavanje
- Neuroni su međusobno povezani

Neuron

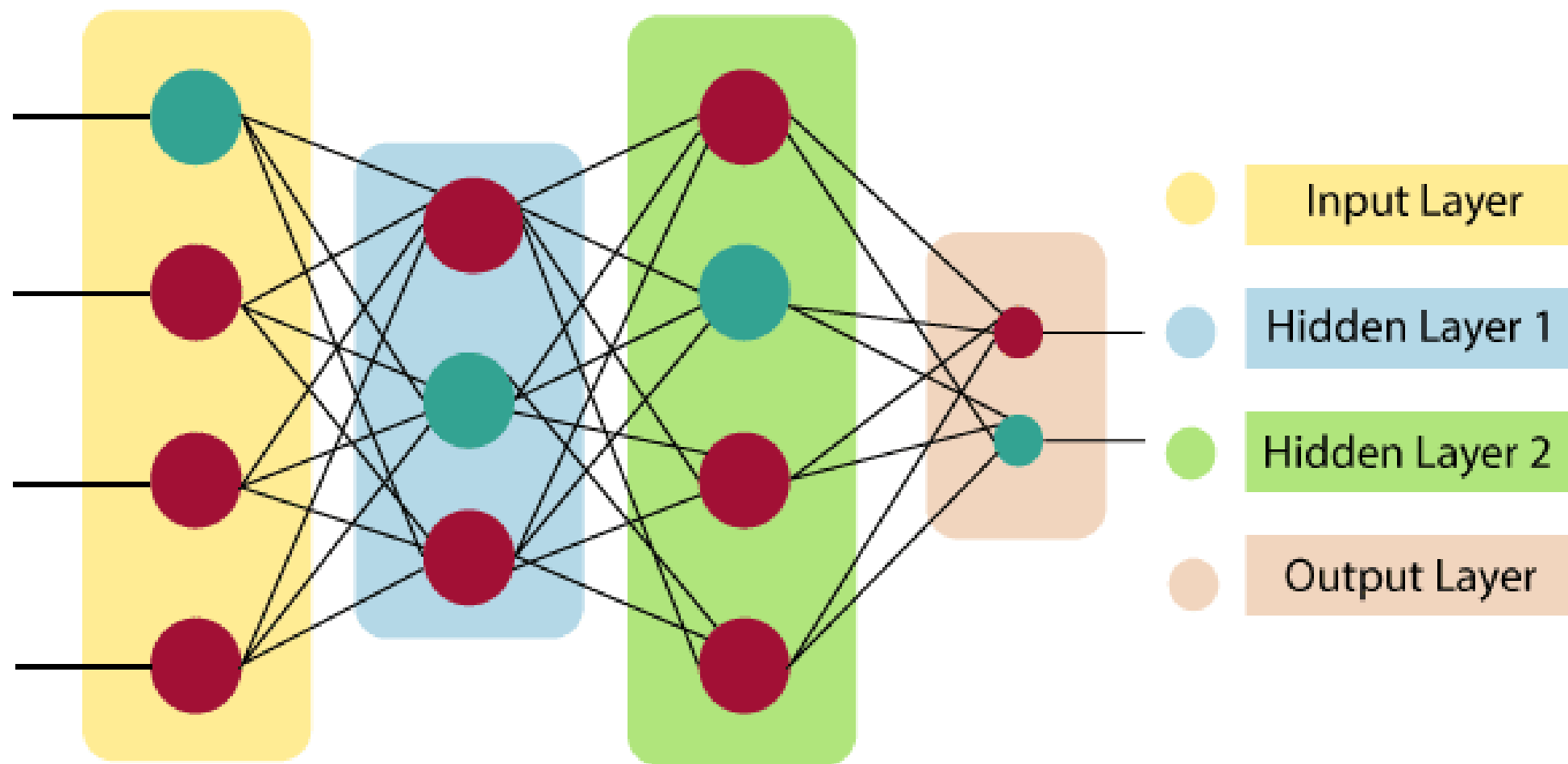
- Najmanja građevna jedinica kod neuronskih mreža
- Neuron se sastoji od:
 - Ulaza – primanje informacija (ekvivalentno podražajima kod živih organizama) i težina
 - Funkcije aktivacije – određuje izlaz neurona (ReLU, linearna funkcija...)



Slojevi

- Svaka neuronska mreža sastoji se od slojeva
- Broj slojeva nije ograničen (najčešće ovisi o problemu koji se pokušava riješiti), a svaki sloj sadrži određeni broj čvorova (neurona)
- Pojedini slojevi neuronske mreže ne moraju imati isti broj čvorova
- 3 vrste slojeva (hijerarhijski gledano):
 1. Ulazni sloj (input layer) – prima podatke, ništa ne računa
 2. Skriveni sloj (hidden layer) – ne vidi se, obrađuje podatke koji su dovedeni na ulaz te se obrađeni podaci prosljeđuju na izlaz (funkcija aktivacije)
 3. Izlazni sloj (output layer) – daje konačni rezultat koji se dobije kao rezultat posljednjeg skrivenog sloja

Neuronska mreža

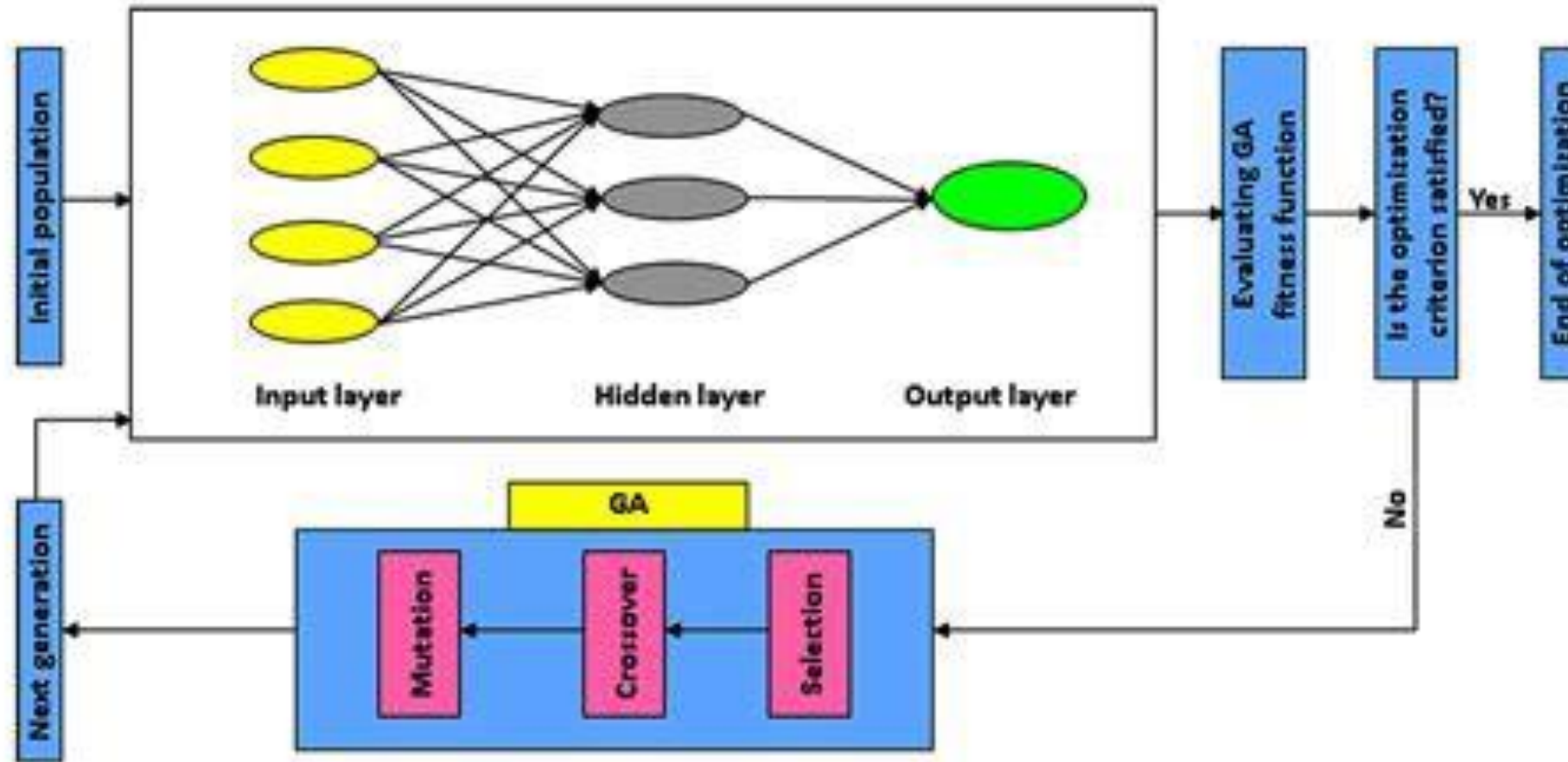


Umjetna neuronska mreža potpomognuta genetskim algoritmom

Umjetna neuronska mreža potpomognuta genetski algoritmom

- Problem postavljanja težina kod neuronskih mreža
- Rješenje: upotreba genetskog algoritma za postavljanje težina
- Stvara se populacija neuronskih mreža, koja sadrži različite vrijednosti težina – početna populacija genetskog algoritma
- Prilikom izračuna težina neuronskih mreža, uzimaju se najbolje neuronske mreže te se pomoću njih računaju nove težine
- Postupak se ponavlja, dok se ne dođe do zadanog maksimalnog broja iteracija

Umjetna neuronska mreža potpomognuta genetskim algoritmom



Korištena metoda

- Umjetna neuronska mreža potpomognuta genetskim algoritmom
- Razlozi:
 - Jednostavna mogućnost mijenjanja težina pojedinih čvorova
 - Lakše učenje
 - Lakša implementacija rješenja za problem
- Omogućen lakši pronalazak optimalne (ili što bolje) rute za zadani problem

Opis implementacije

Uniform cost search (UCS)

- Korišten za implementaciju funkcije dobrote
- Algoritam koji na jednostavan način određuje koji čvor obići
- Odluku o obilasku čvora donosi samo na temelju cijene puta do određenog čvora -> traži put s najmanjom cijenom
- U ovoj implementaciji UCS algoritma, cijena se računa kao maksimalni kapacitet uz minimalni put koji se treba prijeći do čvora

Algoritam za korišteni UCS

odaberi čvor i

provjeri možeš li ga obići i je li već obiđen

ne:

odaberi drugi čvor

da:

izračunaj udaljenost i zahtjev za njega

odaberi čvor j

provjeri možeš li ga obići i je li obiđen

ne:

odaberi drugi čvor

da:

izračunaj udaljenost i zahtjev za njega

ako je za čvor j udaljenost manja, a zahtjev veći:

postavi na mjesto čvora i čvor j

ima još čvorova za obići

da:

vrati se na početak

ne:

vrati čvor i kao rezultat

Gdje bi se koristila neuronska mreža?

- Prilikom izračuna koji čvor je potrebno obići
- UCS pretragu čvorova za obilazak obavlja na „klasičan” način (usporedba dviju brojčanih vrijednosti)
- Posao dobivanja brojčanih vrijednosti za usporedbu se sada prepušta neuronskoj mreži (računa izlaze za dane ulaze)

Algoritam za izračun funkcije dobrote

odaberi čvor i

provjeri možeš li ga obići i je li već obiđen

ne:

odaberi drugi čvor

da:

neuronskoj mreži predaj podatke za čvor i spremi rezultat

odaberi čvor j

provjeri možeš li ga obići i je li obiđen

ne:

odaberi drugi čvor

da:

neuronskoj mreži predaj podatke za čvor i spremi rezultat

ako je za čvor j povoljniji rezultat:

postavi na mjesto čvora i čvor j

ima još čvorova za obići

da:

vrati se na početak

ne:

vrati čvor i kao rezultat

Implementacija umjetne neuronske mreže potpomognute genetskim algoritmom

- Za implementaciju je korišten:
 - Programski jezik Python (brojne biblioteke s gotovim implementacijama: keras i tensorflow)
- Prvi korak: definicija oblika neuronske mreže

```
Ulazni sloj = (funkcija aktivacije: linearna)
Skriveni sloj 1 = (4, funkcija aktivacije: ReLU)
Skriveni sloj 2 = (8, funkcija aktivacije: ReLU)
Izlazni sloj = (funkcija aktivacije: linearna)
Broj rješenja = 3
```
- Idući korak: izrada modela neuronske mreže

Implementacija umjetne neuronske mreže potpomognute genetskim algoritmom

- Definiranje ostalih parametara za genetski algoritam

Broj generacija = 5

Broj roditelja = 2

- Broj kromosoma kojeg sadrži svaka jedinka jednak je broju težina neuronske mreže
- Zadnji korak: postavljanje svih parametara (funkcija dobrote, broj roditelja, broj generacija, inicijalna populacija) i pokretanje instance
- Pomoću funkcije dobrote, genetski algoritam ocjenjuje kvalitetu rješenja kojeg je dala neuronska mreža

Rješenja problema

Rješenje za treće vozilo pri korištenju algoritma UCS (slika gore) i ANN + GA (slika dolje)

```
Vehicle number: 3:  
Traveled distance: 192.2060519316261  
Remaining capacity: 22.0  
Used capacity: 178  
Visited customers: 14  
Route: 78 -> 29 -> 24 -> 55 -> 25 -> 54 -> 75 -> 56 -> 4 -> 39 -> 23 -> 67 -> 99 -> 59
```

```
Vehicle number: 3:  
Traveled distance: 228.30854494941326  
Remaining capacity: 0.0  
Used capacity: 200  
Visited customers: 19  
Route: 5 -> 84 -> 17 -> 18 -> 52 -> 7 -> 19 -> 11 -> 47 -> 36 -> 46 -> 6 -> 8 -> 45 -> 83 -> 60 -> 10 -> 62 -> 88
```


Rješenja problema

	UCS	ANN + GA
Broj korištenih automobila	8	7
Ukupna prijeđena udaljenost	1202,09	1444,71
Ukupni preostali kapacitet	257	89
Ukupni iskorišteni kapacitet	1343	1311
Maksimalni iskorišteni kapacitet	198	200

Zaključak

- Rješavanjem problema usmjeravanja vozila dobije se put kojim vozila trebaju obilaziti korisnike
- Umjetna neuronska mreža potpomognuta genetskim algoritmom daje bolje rezultate za rješavanje iste instance problema nego upotreba algoritma UCS
- Problem i poteškoće: sporo izvršavanje, Python...
- Poboljšanja:
 - Prilagodba parametara neuronske mreže (promjena broja skrivenih slojeva te broja čvorova unutar njih)
 - Mijenjanje parametara genetskog algoritma
 - Eksperimentiranje...

Hvala na pažnji!

