



Klasterovanje tačaka korišćenjem genetskog algoritma

Projekat u okviru kursa
Računarska inteligencija

Opis problema

- Za dati skup tačaka $\text{points}(n\text{-dimenzionog prostora})$ i zadati broj klastera K potrebno je pronaći adekvatne centre klastera kojim će se početni skup tačaka podeliti na odgovarajuće klastere.

Implementacija



Obrada ulaznih podataka


Unos ulaznih podataka je omogućen na dva načina:

- 1) Slučajnim generisanjem tačaka iz odgovarajućih intervala
- 2) Čitanjem fajlova sa zadate putanje. Fajlovi su formatirani na odgovarajući način (svaki red predstavlja jednu tačku u n-dimenzionom prostoru ili csv format)



Implementacija jedinke


- Svaka jedinka u genetskom algoritmu biće predstavljena kao lista dmenzije K gde svaki od elemenata liste je lista dimenzije n (n je dimenzija prostora u kome vršimo klasterovanje) i predstavlja centar za jedan od K klastera.
- Inicijalna populacija se generiše slučajnim izborom p (p -veličina populacije) tačaka iz skupa points koji je zadat na početku.

- 
- **Fitness** funkcija je zadana kao suma kvadratnih rastojanja(SSE) od tačaka od odgovarajućeg centra klastera kome tačka pripada.
 - Rastojanja se računaju euklidski.
 - $SSE = \sum_{i=1}^n (x_i - x^-)^2$
 - Pored navedene funkcije koju treba optimizovati (SSE), implementirana su i rešenja korišćenjem silhouette score-a i davies bouldin score-a.



Implementacija selekcije, ukrštanja i mutacije

- **Selekcija** je implementirana kao turnirska sa veličinom turnira 5
- Korišćeno je jednopoziciono **ukrštanje**, pozicija se bira iz intervala $[0, \text{duzina_jedinke})$ i pritom korišćenom implemetacijom neće postojati problem nepoželjnog ponašanja u kome je jedinka podeljena tako da različite koordinate iz iste tačke(gena) pripadnu različitim potomcima

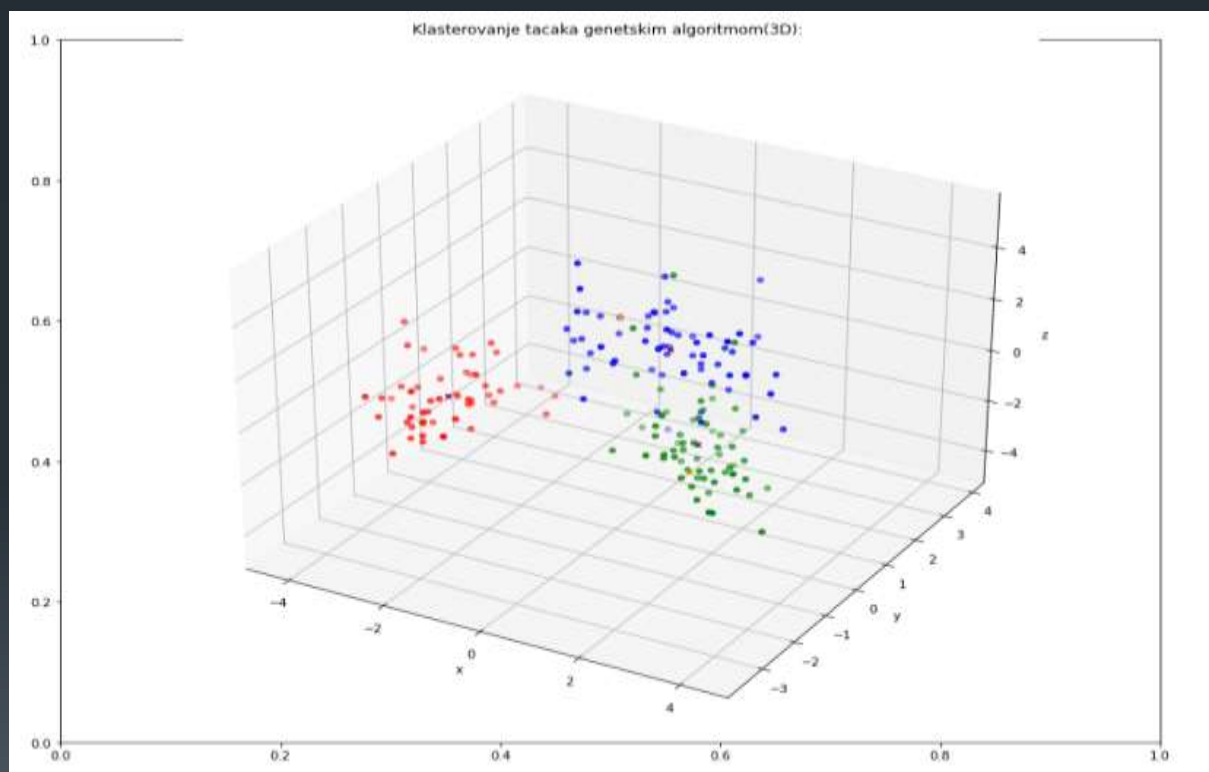
- 
- **Mutacijom** se vrši ekspolaracija prostora pretrage. Upotrebljena je ideja dodavanja/oduzimanja slučajno generisane vrednosti (iz intervala $[0,0.5]$) svakoj koordinati posmatrane tačke ukoliko je slučajno generisana vrednost manja od `MUTATION_RATE` koji je 5% (0.05).

Parametri genetskog algoritma

- Veličina populacije je 300, broj generacija 100 i elitizmom čuvamo 20% najboljih jedinki generacije.

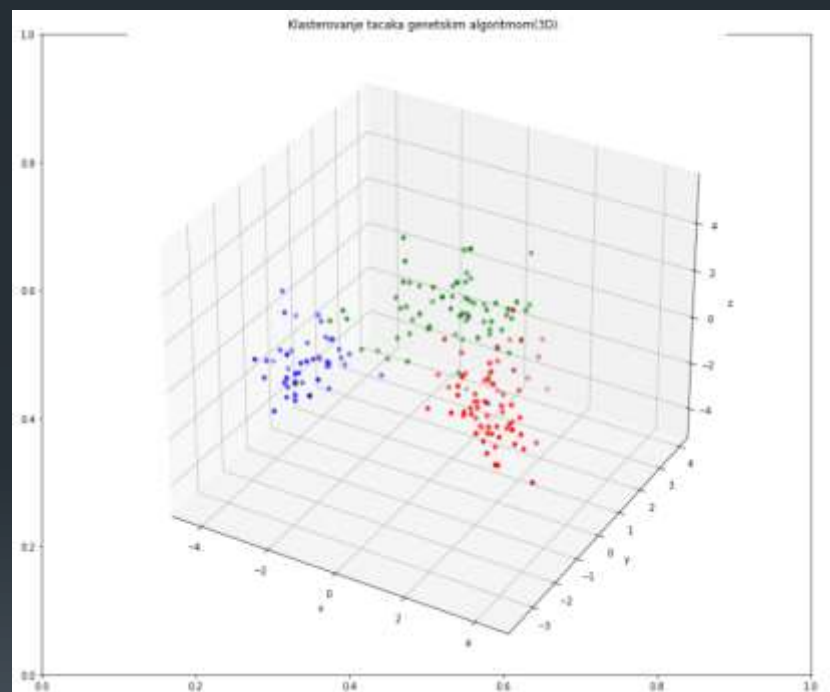
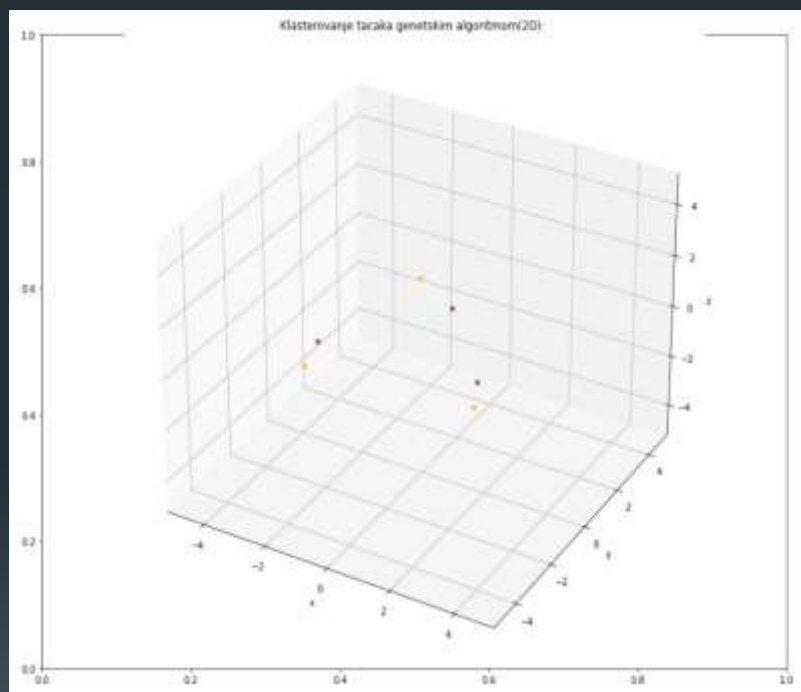
Rezultati:

- Kao poredbeni algoritam koristimo K means algoritam jer on uglavnom daje optimalna resenja za posmatrani problem klasterovanja.



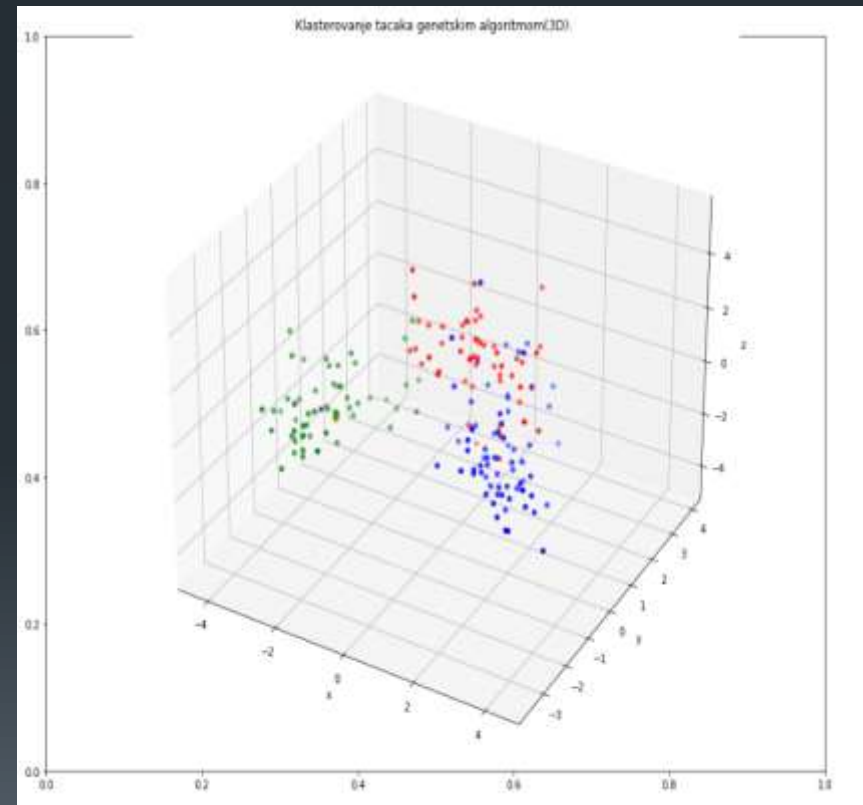
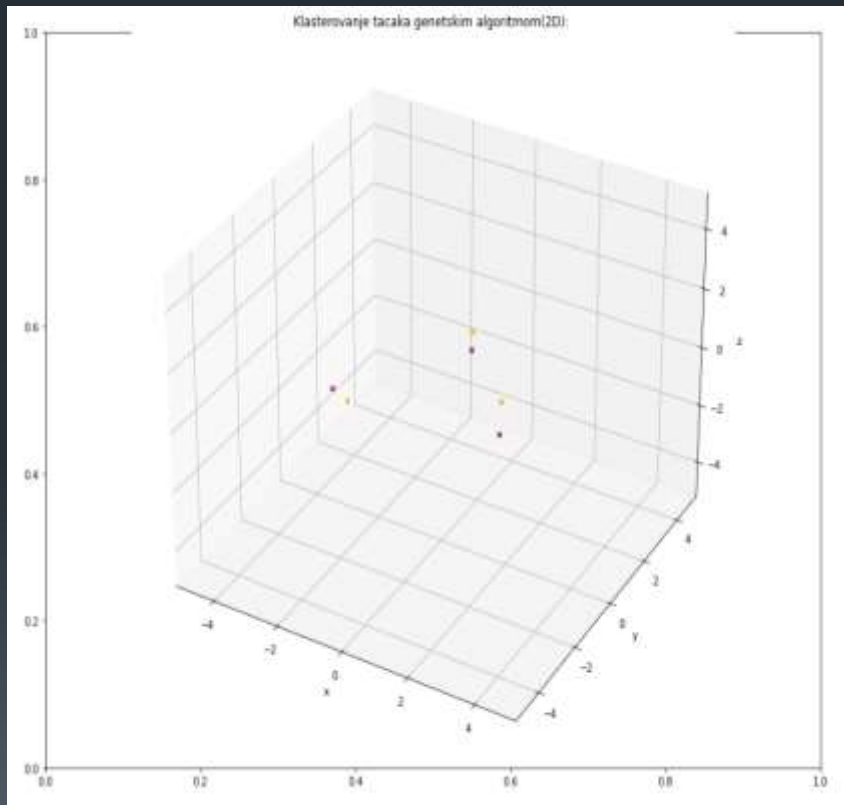
Fitness vrednost dobijena kod poredbenog K means algoritma je 0,00078251.

- Rezultati dobijeni izvršavanjem GA(SSE):
- Veličina populacije 100, broj generacija 20, dobijena fitness vrednost **0.00057494**

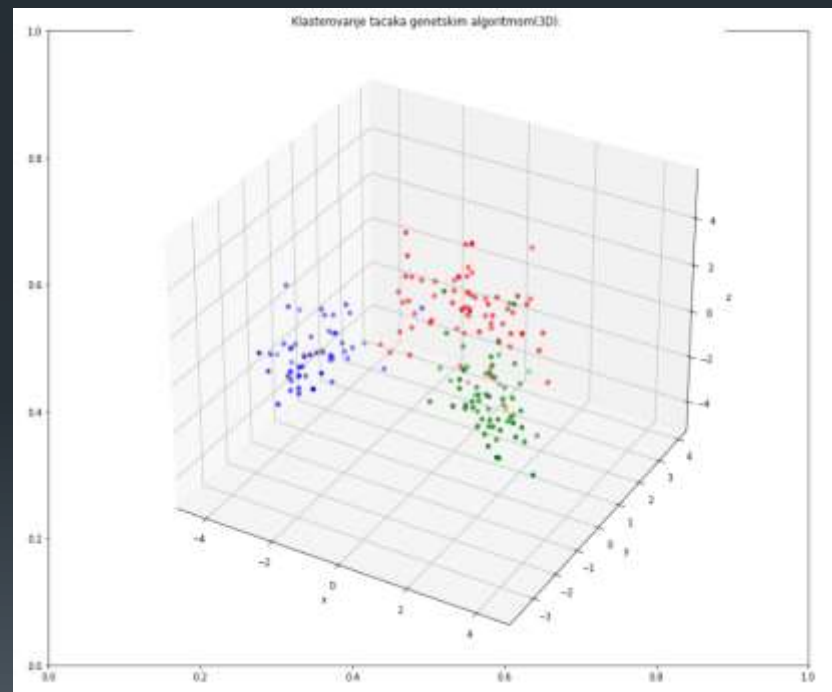
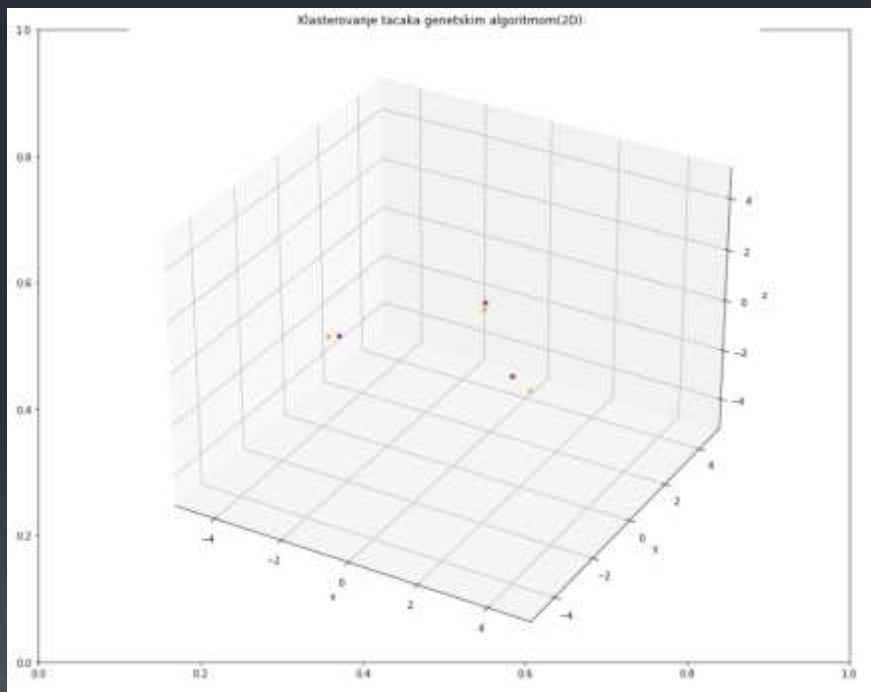


*ljubičaste tačke označavaju centroeide Kmeans algoritma ,
narandžaste tačke označavaju centre klastera GA.

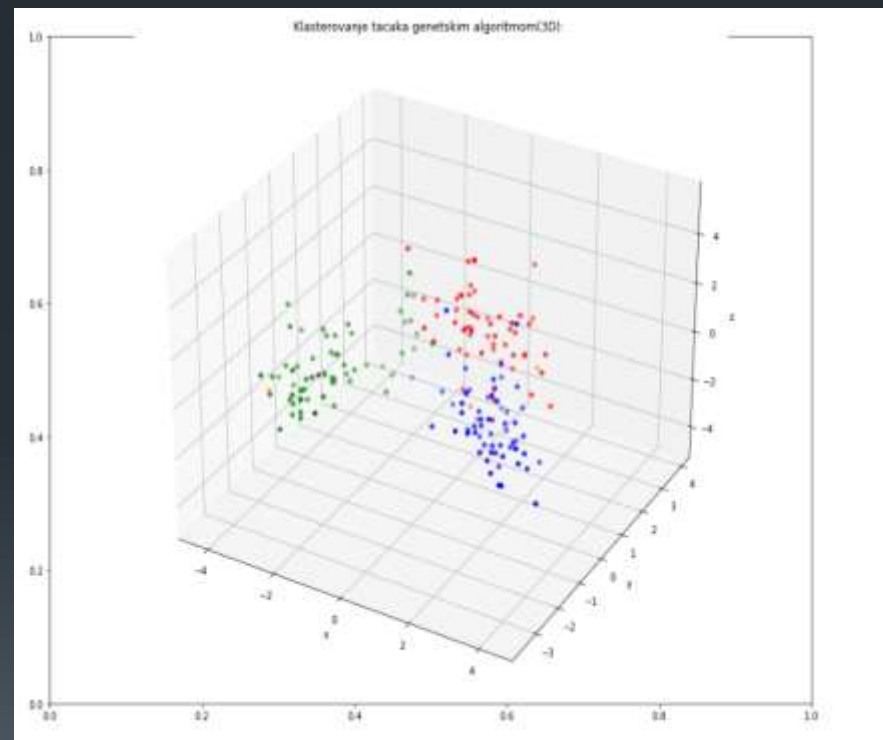
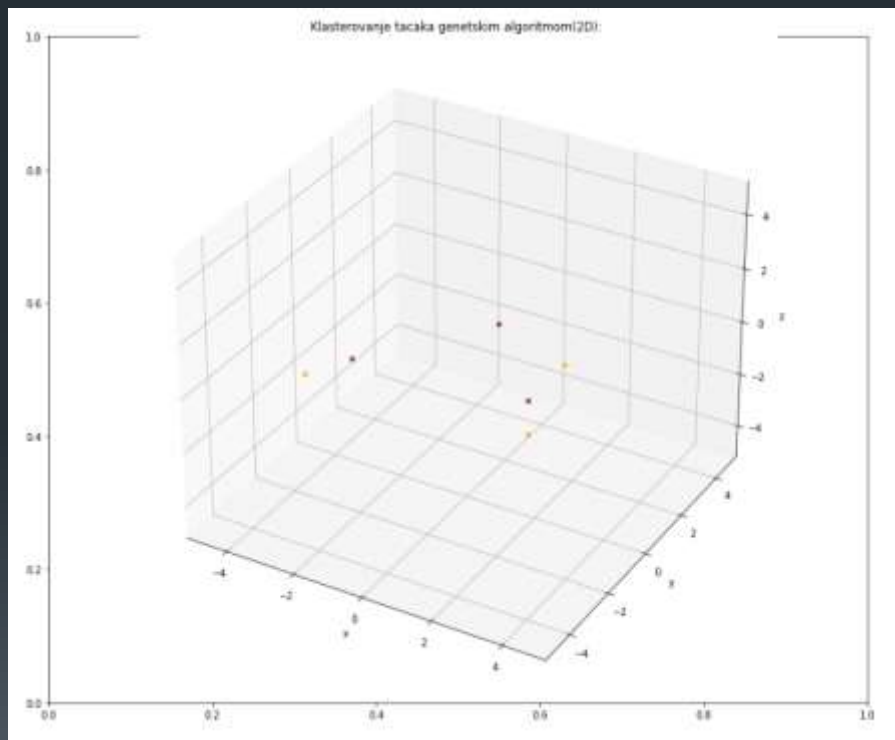
- Rezultati dobijeni izvršavanjem GA(SSE):
- Veličina populacije 200, broj generacija 50, dobijena fitness vrednost **0.000622776**.



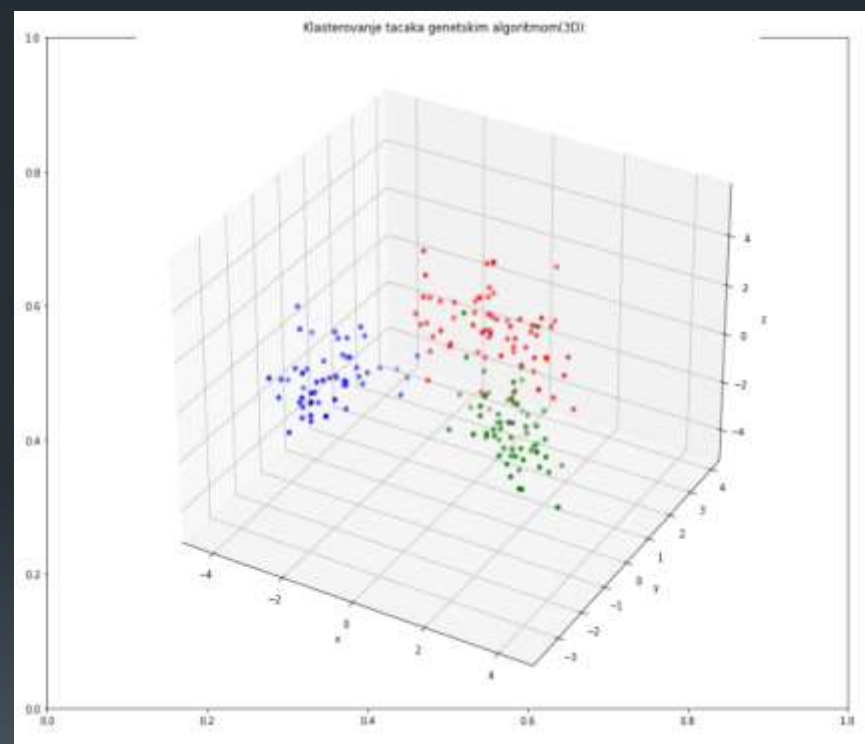
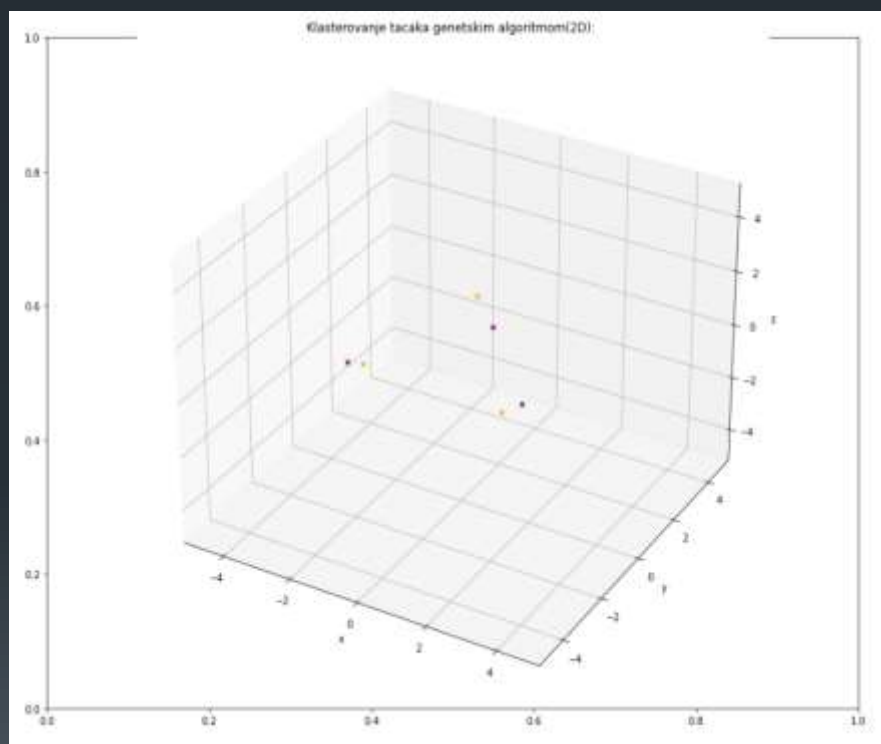
- Rezultati dobijeni izvršavanjem GA(SSE):
- Veličina populacije 300, broj generacija 100, dobijena fitness vrednost **0.000631328**.



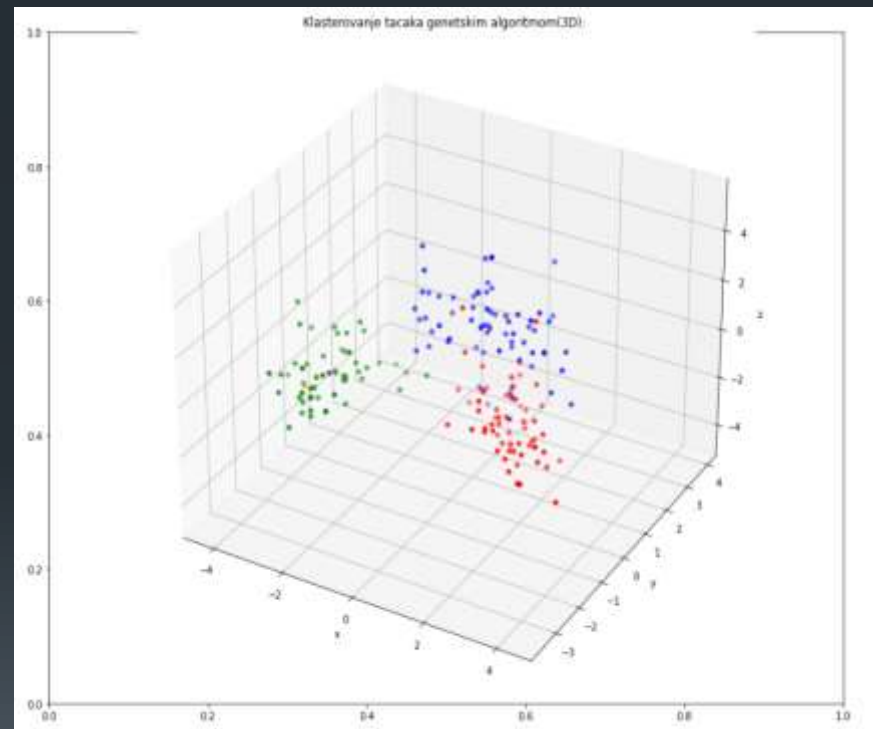
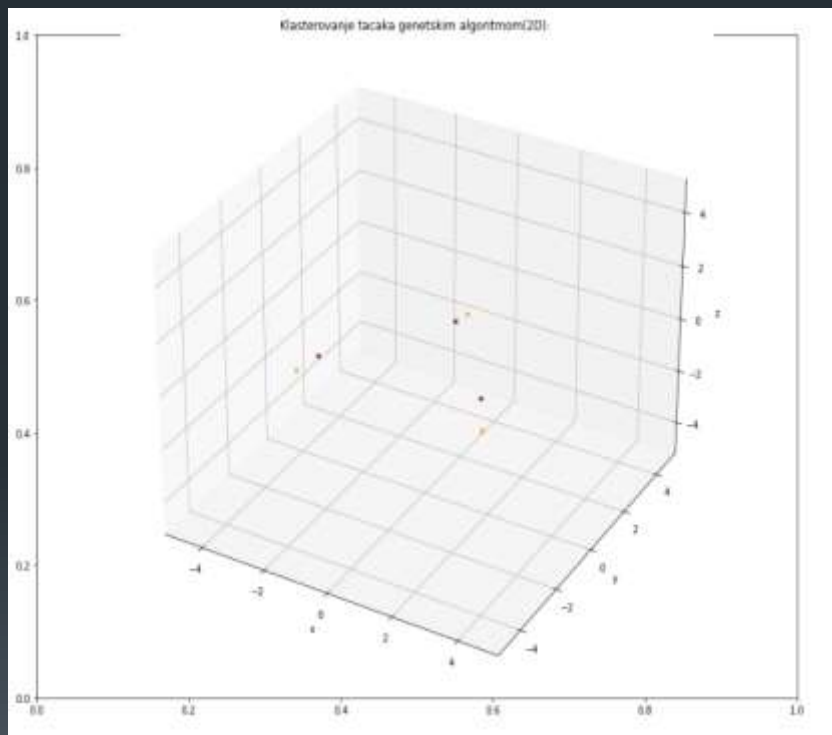
- Vrednost silhouette score-a za K-Mens **0,28485**
- Rezultati dobijeni izvršavanjem GA(silhouette score):
Veličina populacije 100, broj generacija 20, dobijena fitness
vrednost **0,27335**



- Rezultati dobijeni izvršavanjem GA(silhouette score):
Veličina populacije 200, broj generacija 50, dobijena fitness
vrednost **0,28471**



- Rezultati dobijeni izvršavanjem GA(silhouette score):
Veličina populacije 300, broj generacija 100, dobijena
fitness vrednost **0,28599**



Zaključak

- Na osnovu prethodne analize možemo doći do zaključka da povećanjem broja jedinki unutar populacije poboljšava se i kvalitet (fitness vrednost) rešenja samog algoritma i dobijeno rešenje veoma je blizu rešenja uporednog algoritma (K means algoritma) koji u najvećem broju slučajeva daje optimalno rešenje.



Hvala na pažnji

Luka Radenković 59/2018
Ermin Škrijelj 194/2018