

Genetski algoritmi

Primena

Primena genetskih algoritama

- Obučavanje neuronskih mreža
- Traženje najkraćeg, najbržeg i najekonomičnijeg puta (GPS rutiranje)
- Problem trgovačkog putnika
- Strategija igara
- Raspoređivanje procesa, *time scheduling*
- Estimacija parametara sistema
- Optimizacija upita nad bazom podataka
- Permutacioni problemi (bin-packing)
- Podešavanje parametara regulatora (PID, Fuzzy, ...)
- Optimizacija sistema za podršku u odlučivanju
- Traženje optimalne putanje na terenu (ski staza, put za motorna vozila)
- Itd.

Uspešnost genetskih algoritama

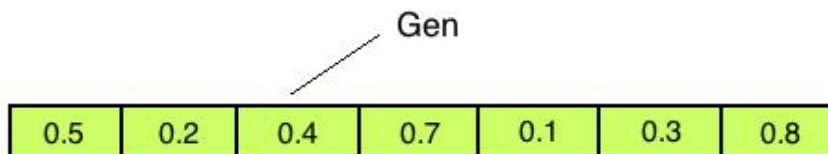
- + Sposobni da odrede položaj globalnog optimuma u prostoru sa više lokalnih ekstrema. Klasične determinističke metode se uvek kreću ka lokalnom optimumu (pri čemu on može biti i globalni, ali to se ne može odrediti iz rezultata).
- + GA ne zavise o jednoj početnoj tački.
- - Gradijentne metode sa sigurnošću nalaze lokalni ekstrem sa željenom preciznošću što nije slučaj sa GA.
- + Moguće je primeniti gradijentnu metodu od početne tačke koja je dobijena sa GA.

Mapiranje Priroda - Računar

PRIRODA	RAČUNAR
Individua (jedinka)	Rešenje problema
Populacija	Skup rešenja
Prilagođenost	Kvalitet rešenja
Hromozom	Kod rešenja
Gen	Deo koda rešenja
Ukrštanje i mutacija	Operatori pretrage
Prirodna selekcija	Ponovna upotreba dobrih (pod-) rešenja

Kodiranje

- ❑ Ovom transformacijom se tačke parametarskog prostora predstavljaju nizovima (stringovima) ili matricama.
- ❑ Šema kodiranja predstavlja način transformacije specifičnog znanja vezanog za određeni problem u okvire genetskog algoritma, što ujedno predstavlja i korak koji će direktno (ključno) uticati na uspešnost genetskog algoritma.
 - ❑ Binarno i sivo (Gray) kodiranje;
 - ❑ Realni brojevi, diskretne vrednosti (najčešće nizovi ali mogu biti N-dimenzionalne matrice u opštem slučaju, npr. kod kodiranja koordinata putanje)
 - ❑ Permutacioni problemi - svaku permutaciju treba kodirati na jedinstven način (bin-packing problem, problem trgovačkog putnika, ...)



Generisanje početne populacije

☐ Veličina populacije

- ☐ Veća populacija daje više genetskog materijala a samim tim i veće šanse da se nađe optimalno rešenje, ali usporava izvršavanje algoritma;
- ☐ Prednost manje populacije je brzina;
- ☐ Da li je bolje manja populacija kroz više generacija ili veća populacija kroz manji broj generacija? Najbolje je negde između?
- ☐ Promenljiva veličina populacije kroz generacije? Zašto da ne?

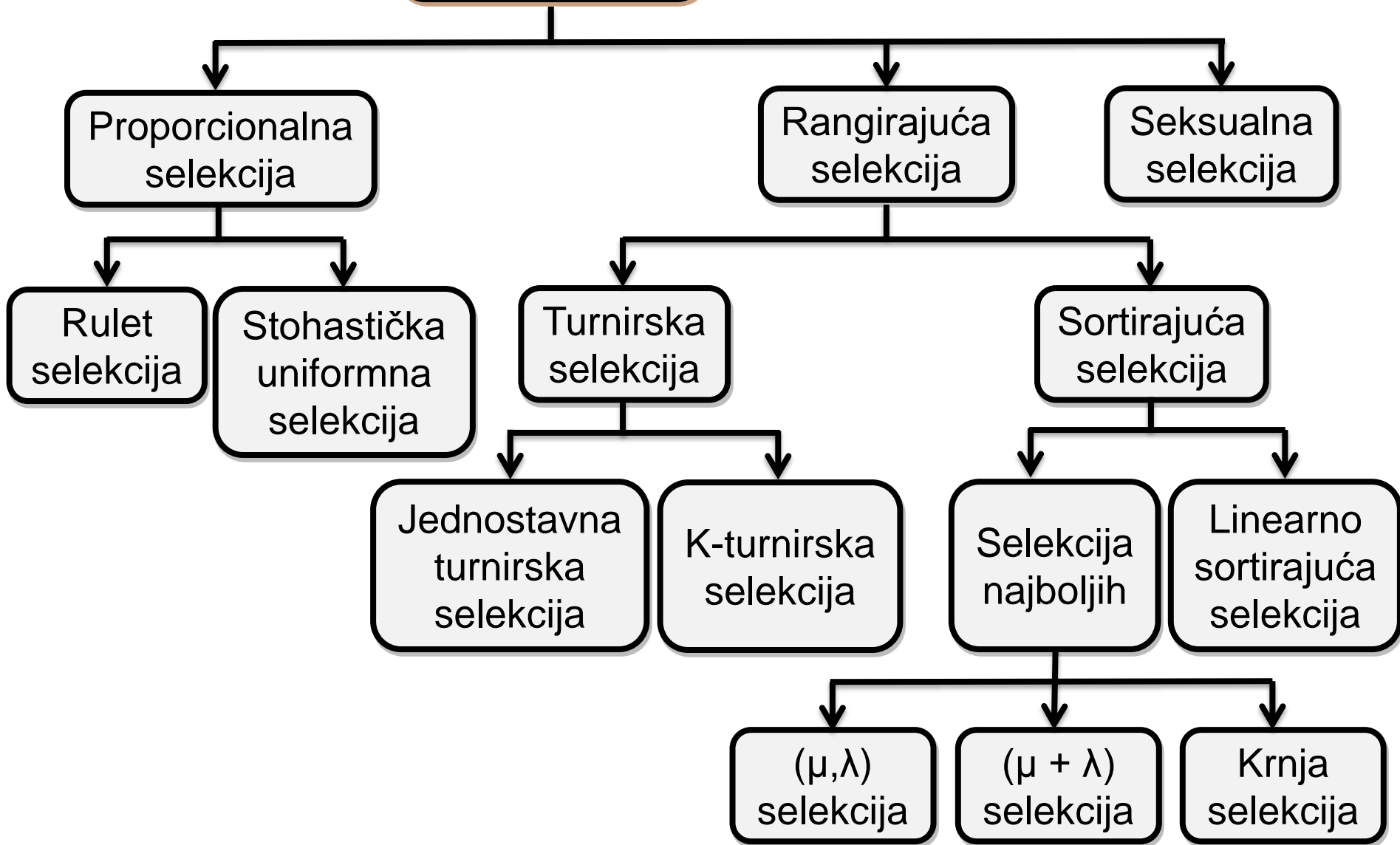
☐ Princip generisanja

- ☐ Slučajno (najčešće)
- ☐ Uniformno raspoređene jedinke u prostoru rešenja (prednost je što uniformno pokriva prostor pretrage, a mana što je ovo često veštačka tvorevina gena). PROBLEM!
- ☐ U početnu populaciju treba uključiti rešenja dobijena drugim optimizacionim algoritmima, ako su dostupna.

Prilagođenost (procena pogodnosti)

- Prilagođenost jedinke je u stvari vrednost kriterijumske funkcije.
- Kriterijumska funkcija se često naziva i kriterijum optimalnosti (Fitness function)
- Određivanje dobre kriterijumske funkcije je ključno za uspešnost optimizacije

Selekcija

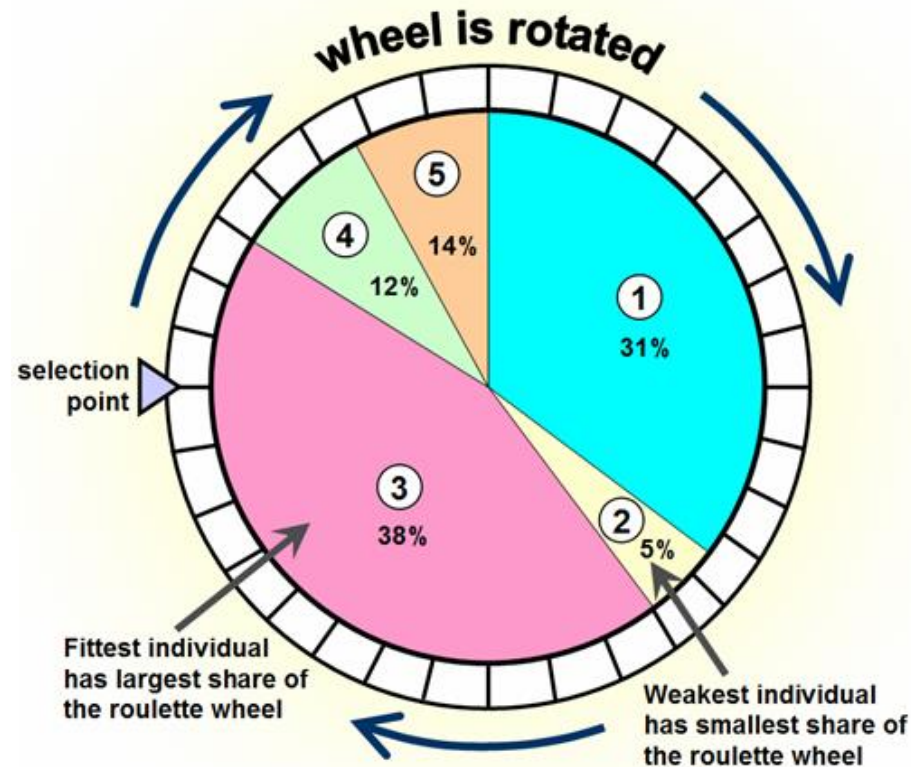
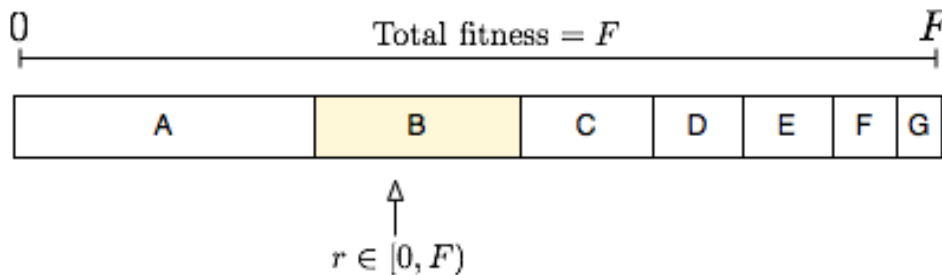


Rulet selekcija

- **Fitness proportionate selection (FPS)**, also known as **roulette wheel selection (RWS)**
- Verovatnoća da će jedinka biti selektovana za ukrštanje je proporcionalna njenom stepenu prilagođenosti

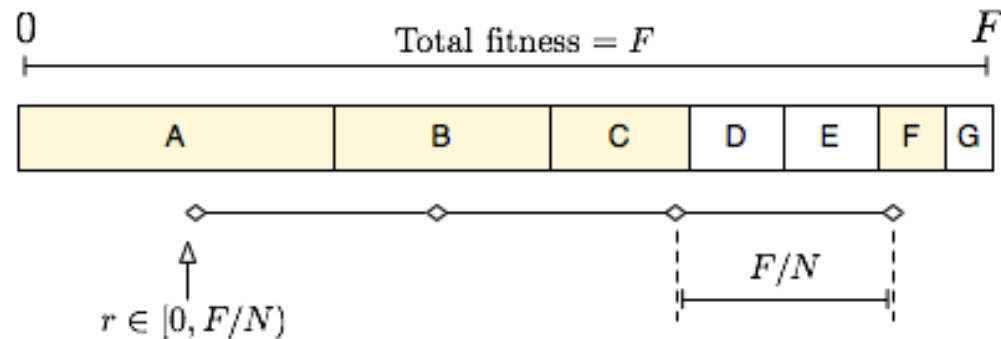
$$p_i = \frac{f_i}{\sum_{j=1}^N f_j}$$

N je broj jedinki u populaciji



Stohastička uniformna selekcija

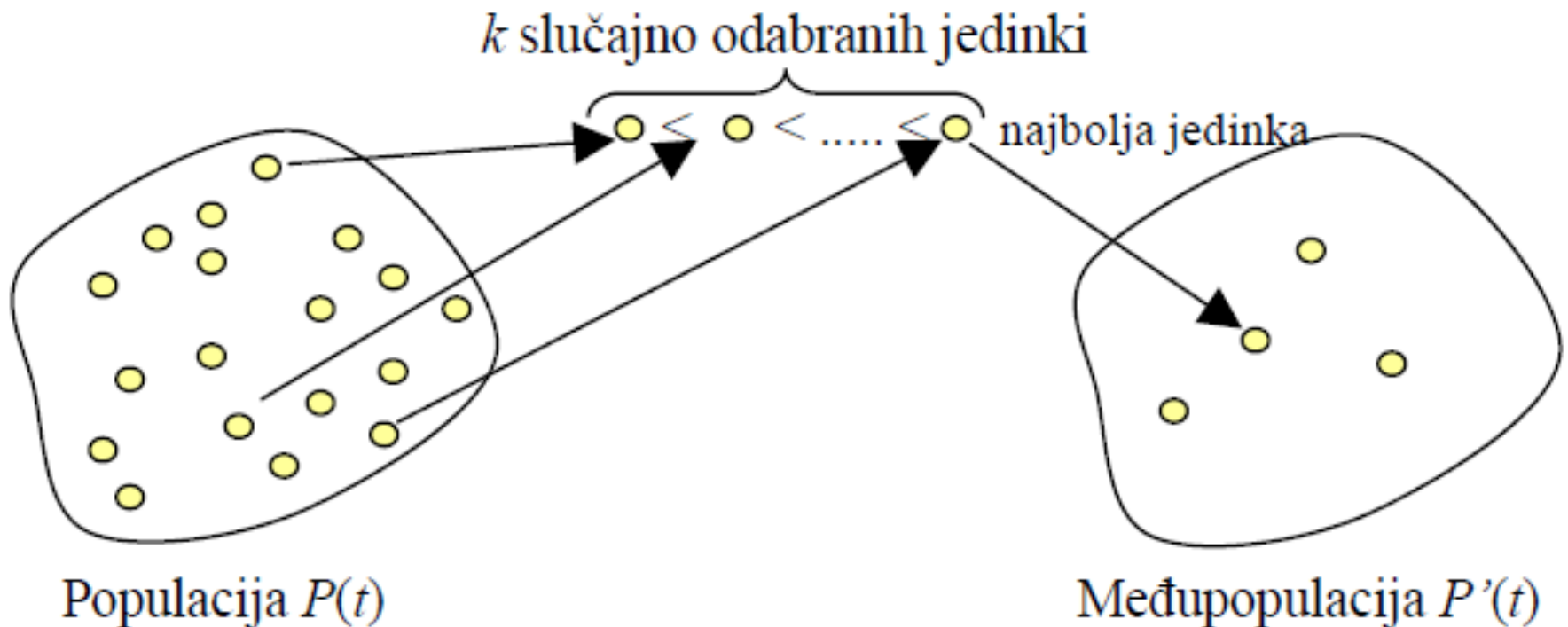
- Stochastic Universal Sampling (SUS) - razvijena od Rulet selekcije
- Baker, James E. (1987). *"Reducing Bias and Inefficiency in the Selection Algorithm"*. Proceedings of the Second International Conference on Genetic Algorithms and their Application. Hillsdale, New Jersey: L. Erlbaum Associates: 14–21.
- Kod ruleta se N roditelja bira tako što se generiše N slučajnih brojeva na intervalu $[0, F)$, a kod SUS-a se generiše samo jedan slučajan broj na intervalu $[0, F/N)$ pa se N roditelja bira tako što se ova vrednost rasporedi N puta na čitavom opsegu **sa jednakim međusobnim rastojanjem (uniformno raspoređeni)**.



- Slabijim jedinkama se daje šansa da budu izabrane (smanjuje se nefer priroda ruleta).

K-turnirska selekcija

- Sprovodi se više „turnira“ među nekoliko (K) jedinki izabranih slučajno iz populacije
- Na turniru, između slučajno odabranih K jedinki pronalazi se najbolja (pobednik turnira - jedinka sa najboljom pogodnošću). N turnira daje N roditelja.

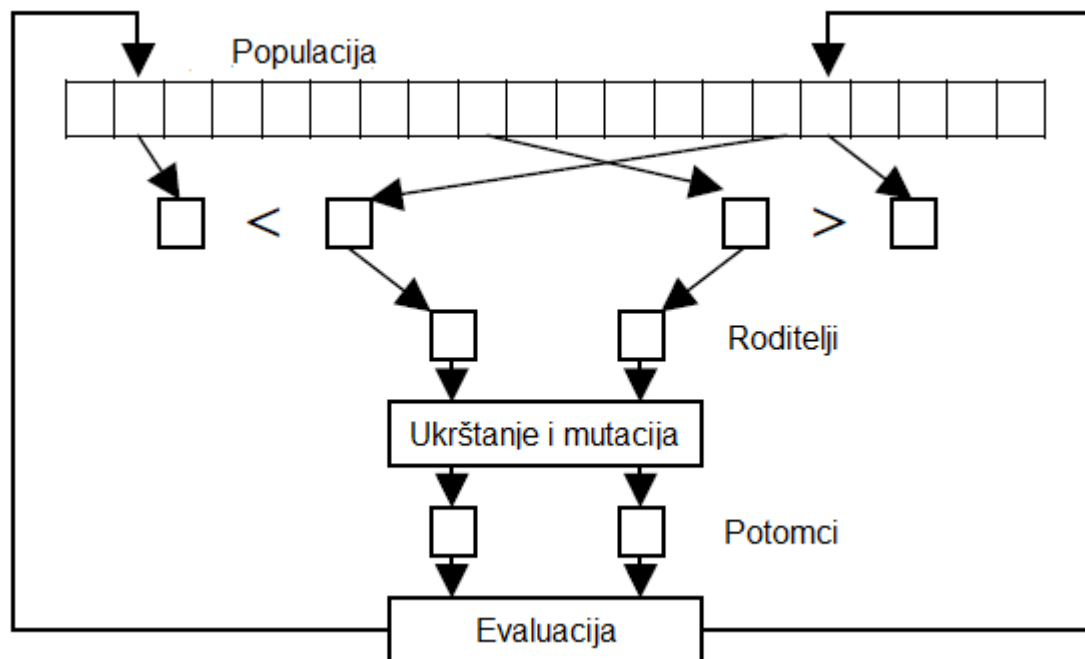


K-turnirska selekcija

- Menjanjem veličine turnira se lako podešava pritisak selekcije. Što je K veće, slabiji pojedinci imaju manje šanse da budu izabrani.
- **$K=1$ najslabijih jedinki nemaju nikakve šanse** (jedan od načina da najlošije jedinke ne budu nikad odabrane)
- Ako je $K = 1$, selekcija je ekvivalentna slučajnom izboru.
- Prednosti:
 - Brzina - pogodnost se računa samo za odabrane jedinke i nema sortiranja cele populacije.
 - Radi na paralelnim arhitekturama
 - Pritisak selekcije se lako podešava

Jednostavna turnirska selekcija

- Specifičan oblik binarne turnirske selekcije ($K = 2$)
- U jednom koraku, slučajnim postupkom se bira dva para jedinki. U svakom paru se lošije jedinke eliminišu. Ukrštanjem boljih jedinki se generiše dvoje potomaka, koji se potom mutiraju i evaluiraju. Taj par novostvorenih jedinki nadomešćuje eliminisane jedinke.
- Na taj način genetski algoritam sa jednostavnom turnirskom selekcijom u istom koraku obavlja i selekciju i reprodukciju.

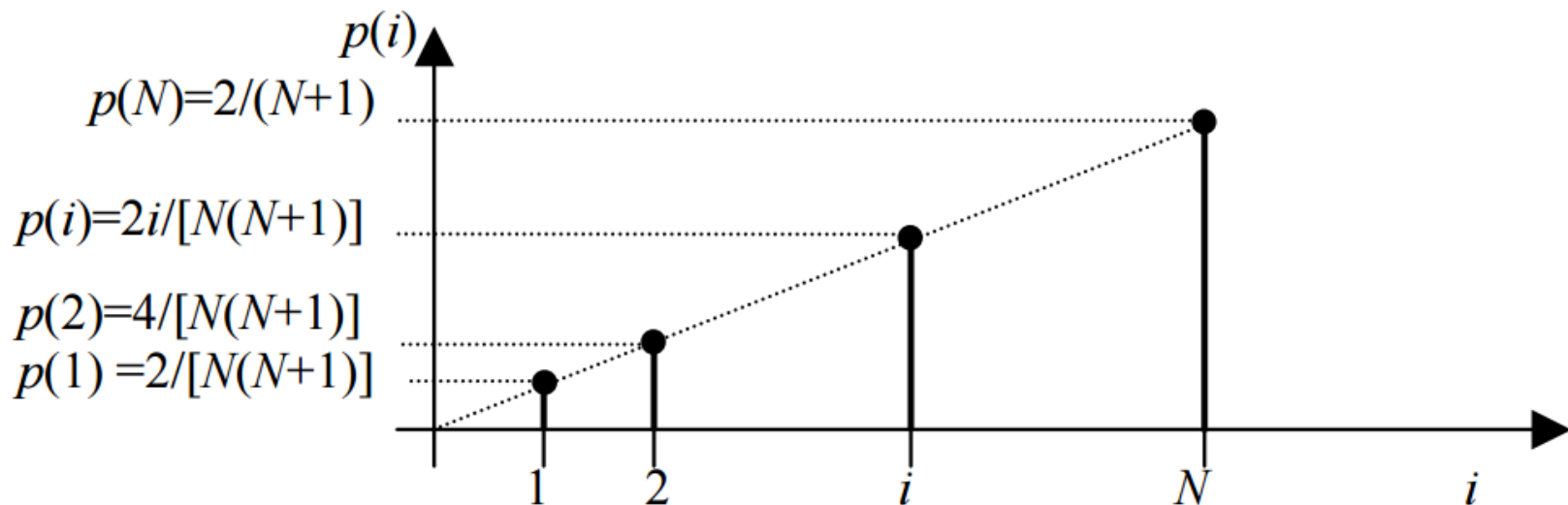


Linearno sortirajuća selekcija

- Kod rangirajućih selekcija verovatnoća izbora ne zavisi direktno od fitnesa, već od položaja jedinke u poretку jedinki sortiranih po fitnesu.
- Kod linearno sortirajuće selekcije verovatnoća je proporcionalna rangi, odnosno poziciji jedinke u poretку sortiranom po pogodnosti.

$$p(i) = \frac{i}{\sum_{i=1}^N i} = \frac{2i}{N(N+1)}, \quad i = 1:N$$

,gde najbolja jedinka ima indeks N , a najgora 1



Selekcija najboljih

- Bira se unapred zadati broj najboljih jedinki
- Tri vrste:
 - $(\mu + \lambda)$ selekcija
 - (μ, λ) selekcija
 - Krnja selekcija
- Razlikuju se prema skupu jedinki iz kojeg se biraju najbolje. Mogu se birati iz skupa samo roditelja, roditelja i potomaka ili samo potomaka.
- Prvo se slučajnim postupkom biraju roditelji, zatim se njihovi potomci (sa ili bez roditelja) sortiraju i preživljava samo određeni broj najboljih jedinki.
- μ – veličina populacije roditelja
- λ – veličina populacije potomaka

Selekcija najboljih

- **$(\mu + \lambda)$ selekcija** – slučajno se bira μ roditelja. Ukrštanjem se stvara λ potomaka. Iz skupa roditelja i potomaka se zatim najboljih μ jedinki bira za sledeću generaciju. Postupak se ponavlja sve dok se ne popuni nova generacija s N novih jedinki, odnosno N/μ puta.
- **(μ, λ) selekcija** – isto se slučajno bira μ roditelja i njihovim ukrštanjem se stvara λ potomaka. Međutim, potomaka je više od roditelja ($\lambda \geq \mu$) i μ najboljih potomaka se bira za narednu generaciju.
- Kod obe metode se odvija rekombinacija (selekcija + ukrštanje)
- **Krnja selekcija** bira n najboljih jedinki i kopira ih N/n puta u bazen za ukrštanje.

Seksualna selekcija

- **Seksualna selekcija** (koristi se u specijalnom slučaju kada je populacija podeljena na dve populacije – mužjake i ženke). Svrha ovoga je definicija ko može sa kim da se ukršta i pod kojim uslovima. Jedan od načina da se sprovede multikriterijumska optimizacija, jer može da se računa jedna pogodnost za pravo na ukrštanje a druga za verovatnoću da će doći do ukrštanja (oponaša prirodni proces).
- **Ukrštanje sa maticom** – simulira razmnožavanje pčela kod kojih jedna jedinka (matica) sa najboljim fitnessom učestvuje u svim ukrštanjima sa ostalim jedinkama (trutovima).

Ukrštanje

- U jednoj tački, dve tačke, N tačaka i uniformno.
 - Ne može da se koristi u slučajevima kada tako dobijeni potomci ne pripadaju kodiranom parametarskom prostoru, npr. kod permutacionih problema.
 - Ukrštanje u jednoj ili dve tačke je preporučljivo pri većim populacijama, jer veća populacija ima i veću raznolikost šema. S druge strane, uniformno ukrštanje se koristi pri manjim populacijama, da populacija ne bi postala homogena.
- Ograničeno ukrštane u N tačaka, npr. kod rutiranja putanja (samo neke jedinice mogu da se ukrštaju i to samo na nekim mestima)
- Ukrštanje permutacija
 - Problem je što se permutacije sastoje od niza indeksa i kod običnog ukrštanja može da se desi da se kod potomka jave dva ista indeksa na različitim mestima što više nije validna permutacija (nije deo parametarskog prostora)
 - Varijanta 1 - PMX (Partially Matched Crossover)
 - Varijanta 2

PMX (Partially Matched Crossover)

- Prva varijanta ukrštanja permutacija
- Prvi korak je obično ukrštanje u N tačkaka (2 tačke na primeru)

Roditelj 1	1 4 7		3 8 2		5 6
Roditelj 2	3 6 2		8 1 7		4 5
Potomak 1	1 4 7		8 1 7		5 6
Potomak 2	3 6 2		3 8 2		4 5

- U drugom koraku se proveriti da li se neki indeksi ponavljaju kod deteta (dovoljno proveriti kod jednog deteta), pa se na tim mestima izvrši dodatno ukrštanje.

Dete 1	3 4 2		8 1 7		5 6
Dete 2	1 6 7		3 8 2		4 5

PMX (Partially Matched Crossover)

1. Select the substring

Parent 1	2	1	4	3	7	6	9	8	5
Parent 2	9	4	1	2	5	3	8	6	7



2. Exchange substrings

Proto-offspring 1	2	1	1	2	5	3	9	8	5
Proto-offspring 2	9	4	4	3	7	6	8	6	7



4. Legalize the offspring

Offspring 1	6	4	1	2	5	3	9	8	7
Offspring 2	9	1	4	3	7	6	8	2	5



3. Mapping relationship

1 ↔ 4
2 ↔ 3 ↔ 6
5 ↔ 7

Druga varijanta ukrštanja permutacija

- Nasumično se odabere određeni broj indeksa gena
- Na tim indeksima se preuzmu geni jednog roditelja
- Utvrdi se koji su geni iskorišćeni i oni neće biti uzeti u razmatranje u drugom roditelju
- Preostali geni se uzimaju od drugog roditelja u redosledu u kojem su poređani i popunjavaju se s leva na desno na preostala mesta u jedinki koja se formira
- Ukrštanjem dva roditelja nastaje samo jedna jedinka

Roditelj 1 1 4 7 3 8 2 5 6

Roditelj 2 3 6 2 8 1 7 4 5

Dete 6 8 7 3 1 2 4 5

Mutacija

- Negacija kod binarnog kodiranja (0 u 1 i obrnuto)
- Realne vrednosti (sabiranje sa malim slučajno generisanim brojem)
- Izmena koordinata putanje (krive u N-dimenzionalnom prostoru u opštem slučaju). Moguće su samo takve izmene koje neće dovesti do prekida putanje.
- Mutacija pregrupisavanja - zamena gena (na primer kod permutacionih problema gde se ne smeju pojavljivati novi geni i svaki gen može da se pojavi tačno jednom).

Pre mutacije	1 4 5 7 8 2 3 6
Posle mutacije	1 8 5 7 4 2 3 6

- ✓ Kod realno-kodiranih jedinki se može koristiti veći stepen mutacije jer se kod njih, za razliku od binarno-kodiranih, povećava nivo moguće pretrage prostora rešenja, a da se ne utiče negativno na karakteristike konvergencije.

Zašto je mutacija važna?

- Mutacijom se pretražuje parametarski prostor i upravo je mutacija mehanizam za **izbegavanje lokalnih optimuma**.
- Ako cela populacija završi u nekom od lokalnih optimuma, jedino se slučajnim pretraživanjem parametarskog prostora pronalazi bolje rešenje. Dovoljno je da jedna jedinka (nastala mutacijom) bude bolja od ostalih, pa da se u nekoliko narednih generacija, sve jedinke "*presele*" u prostor gde se nalazi bolje rešenje.

Varijacije operatora mutacije

- ❖ Postoje mnoge varijacije operatora mutacije. Na primer:
 - ❖ Naklonjenost mutacije jedinkama sa manjim fitnessom da bi se proširilo polje pretrage, a istovremeno očuvale informacije kod onih sa većim fitnessom, ili
 - ❖ Parametrizacija mutacije tako da frekvencija mutacije opada sa konvergencijom populacije.
- ❖ Dodavanje novih jedinki u populaciju
 - ❖ Od kreiranja početne populacije na slučajan način može mnogo da zavisi uspešnost algoritma. Paralela smanjenju stepena mutacije kroz generacije može da bude zamena najlošijih (ili najstarijih) jedinki novim jedinkima generisanim na slučajan način. U ovom slučaju broj jedinki koji se zamenjuje opada kroz generacije i time se postepeno prelazi iz slučajne pretrage u optimizaciju genetskim algoritmom.

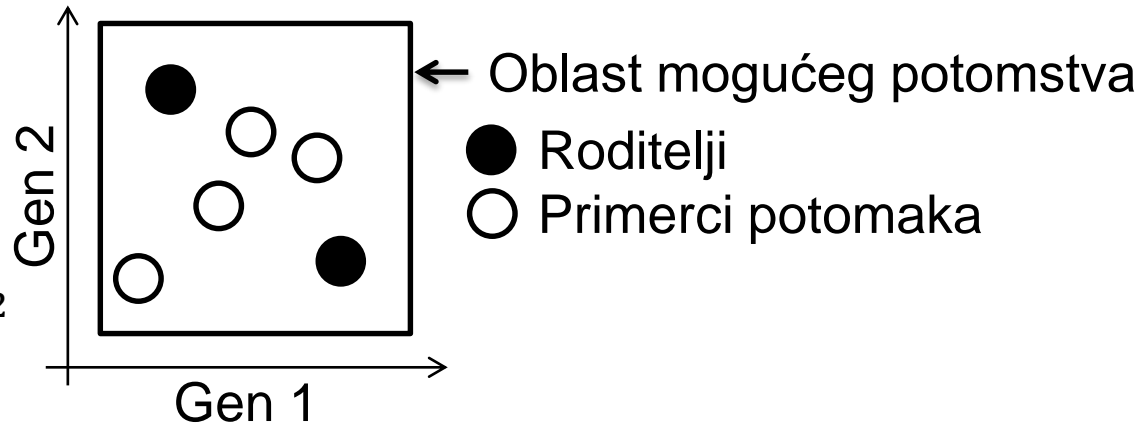
Rekombinacija (ukrštanje + mutacija)

■ Na primer, kod jedinki kodiranih nizom realnih brojeva

■ **Intermedijarna rekombinacija** (*intermediate recombination*) – dobijaju se nove jedinice okolo i između roditelja. α se bira za svaki par roditeljskih gena, najčešće:

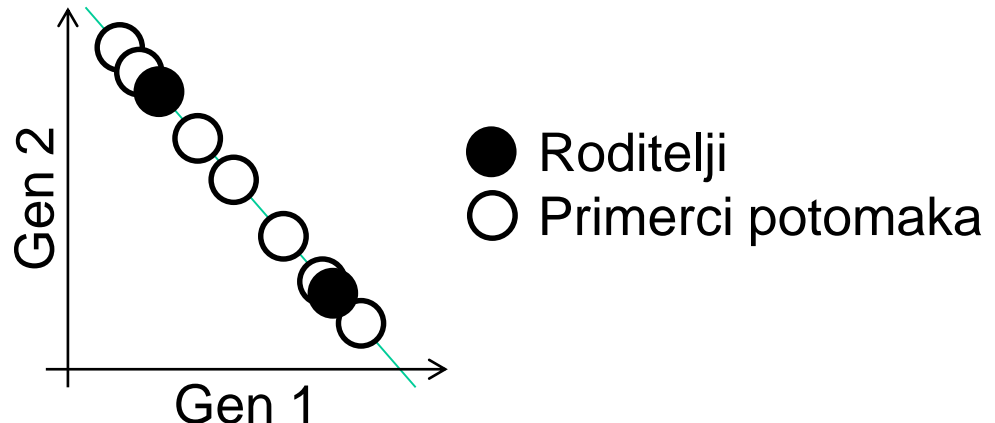
$$\alpha \in [-0.25, 1.25]$$

$$Q_1 = \alpha_1 \times P_1 + (1 - \alpha_2) \times P_2$$



■ **Linearna rekombinacija** (*linear recombination*) – jedna vrednost α u toku rekombinacije

$$Q_1 = \alpha \times P_1 + (1 - \alpha) \times P_2$$



Elitizam

- **Obezbeđuje monotono nerastuću funkciju prilagođenosti najbolje jedinke kroz generacije.**
- Kako odabrati koliko jedinki preneti u novu generaciju? Dve jedinke? Zašto ne 1? 10% populacije? Ima smisla ovu vrednost prilagoditi veličini populacije ali nema smisla preneti previše jedinki (više od 10 je previše).
- Da li koristiti elitizam:
- **+DA+** Jedini način da se najbolja jedinka iz populacije sačuva
- **-NE-** Da bi u svakom koraku evolucije zaštitili najbolju jedinku od bilo kakvih izmena ili eliminacije, potrebno ju je u svakom koraku i pronaći. Pretraživanje ili sortiranje zahteva procesorsko vreme zbog čega se može znatno usporiti genetski algoritam.

Kriterijumi zaustavljanja

- Zadati broj iteracija
- Dostignuta željena vrednost kriterijumske funkcije
- Vreme (dužina trajanja optimizacije)
- Nenapredovanje algoritma - ako se prilagođenost najbolje jedinke nije značajno promenila (*function tolerance*) u zadanom broju generacija (*stall generations*)
 - U ovoj situaciji može da se proba sa nekom promenom u algoritmu, pre nego što se prekine izvršavanje, na primer značajno se uveća stepen mutacije kroz nekoliko generacija zbog uvođenja sveže krvi da bi se posle postepeno smanjivao do stare vrednosti.

U korist i protiv GA

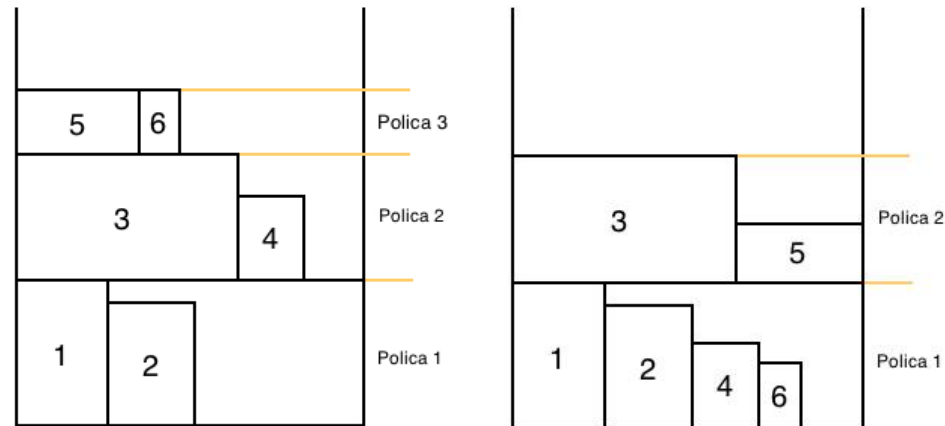
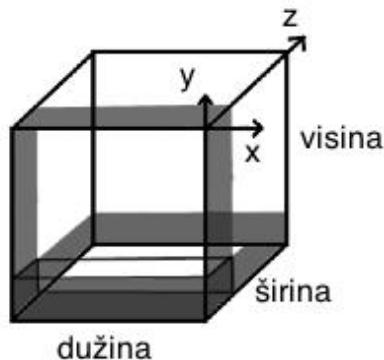
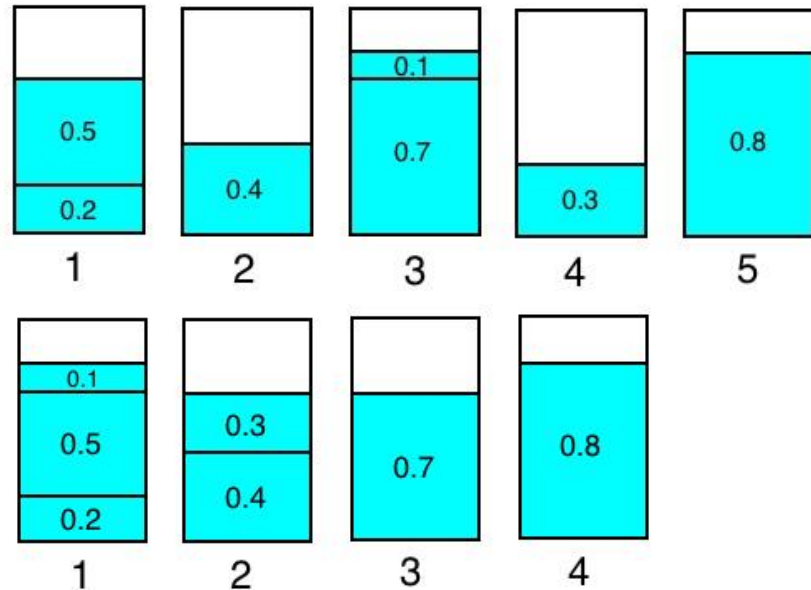
U korist GA	Protiv GA
Kriterijumska funkcija je potpuno proizvoljna, nema posebnih zahteva kao što su neprekidnost, derivabilnost i sl.	Teško je definisati <i>dobru</i> funkciju prilagođenosti, potrebno je prilagoditi GA zadatim ograničenjima.
Primenjiv na veliki broj problema	Često je potrebno prilagoditi problem algoritmu.
Mogućnosti nadogradnje i povećanja efikasnosti (puno stepeni slobode)	Teško je postaviti dobre parametre (veliki uticaj parametara na efikasnost)
Jednostavnim ponavljanjem se može povećati pouzdanost. Rezultat je skup rešenja, a ne jedno rešenje.	Konvergencija znatno sporija od ostalih numeričkih metoda.
Rešava sve optimizacione probleme (realne, binarne, diskretne, znakovne, višemodalne, višedimenzione, ...)	Potrebno je prilagoditi genetske operatore kodiranju
Jednostavnost ideje i dostupnost programske podrške	Spor zbog izvođenja velikog broja računskih operacija. Traži se velika procesorska snaga.

Primeri praktičnih primena

- ◆ Permutacioni problemi
 - ◆ Primena genetskog algoritma u rešavanju bin-packing problema
- ◆ Optimizacija parametara FSO
 - ◆ Realizacija sistema za podršku u odlučivanju pri upravljanju brodskom prevodnicom zasnovanog na rasplinjutoj (fuzzy) logici
- ◆ Optimalne putanje na terenu
 - ◆ Pronalaženje optimalne putanje za ski stazu upotrebom genetskih algoritama
- ◆ Podešavanje parametara regulatora

Primena genetskog algoritma u rešavanju bin-packing problema 1/2

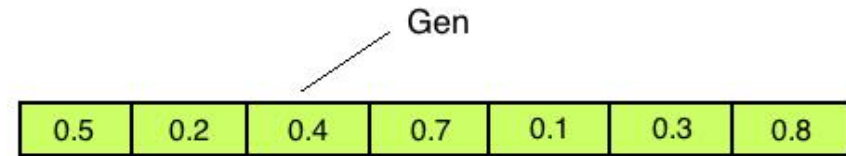
- ❑ Šta predstavljaju bin-packing problemi?
- ❑ Primeri: sečenje žice, sečenje pravougaonih profila, pakovanje kutija u skladište, itd.
- ❑ Bin-packing problemi:
 - jednodimenzionalni (1D)
 - dvodimenzionalni (2D)
 - trodimenzionalni (3D)
- ❑ Algoritmi za pakovanje:
 - trenutni algoritmi
 - naknadni algoritmi
- ❑ Neki klasični algoritmi:
 - Next Fit
 - First Fit
 - Best Fit
 - Worst Fit



Primena genetskog algoritma u rešavanju bin-packing problema 2/2

☐ Kodiranje

- ☐ Ukoliko je N broj objekata u ulaznoj sekvenci, jedinka u tom slučaju predstavlja niz od N objekata. U slučaju jednodimenzionalnog problema, to je niz od N pozitivnih realnih brojeva, u slučaju dvodimenzionalnog problema, to je niz od N pravougaonika, a u slučaju trodimenzionalnog problema je to niz od N kvadrova. Svaki pojedinačni objekat u nizu predstavlja gen jedinke.



☐ Prilagođenost

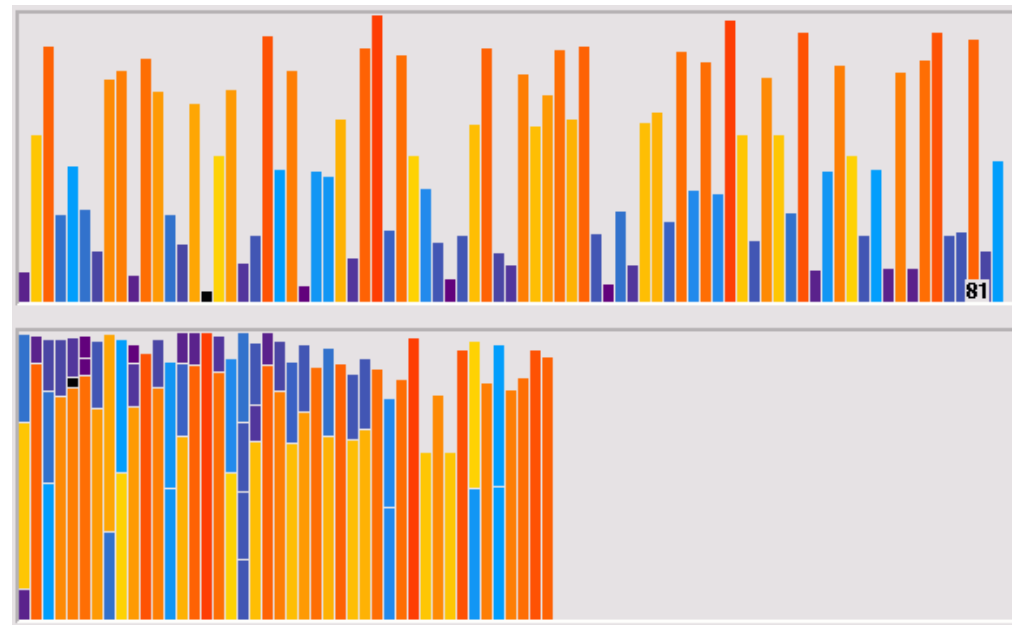
- ☐ Broj iskorišćenih „binova“ u procesu pakovanja

☐ Selekcija

- ☐ Stohastička uniformna

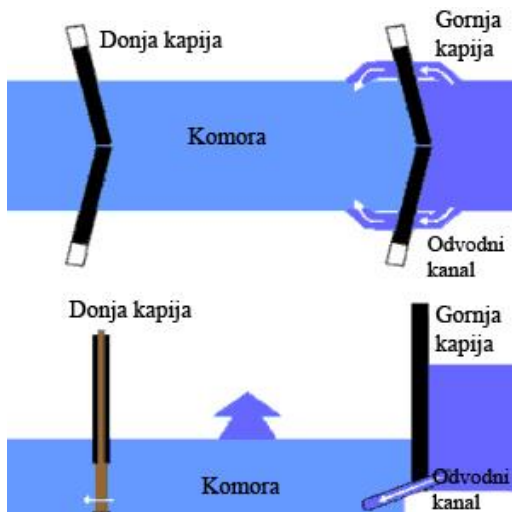
☐ Ukrštanje i mutacija

- ☐ Permutacioni problem (PMX)

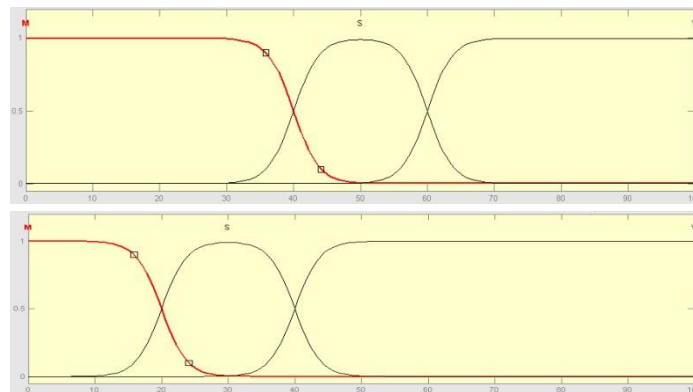


Realizacija sistema za podršku u odlučivanju pri upravljanju brodskom prevodnicom zasnovanog na rasplinutoj (fuzzy) logici 1/2

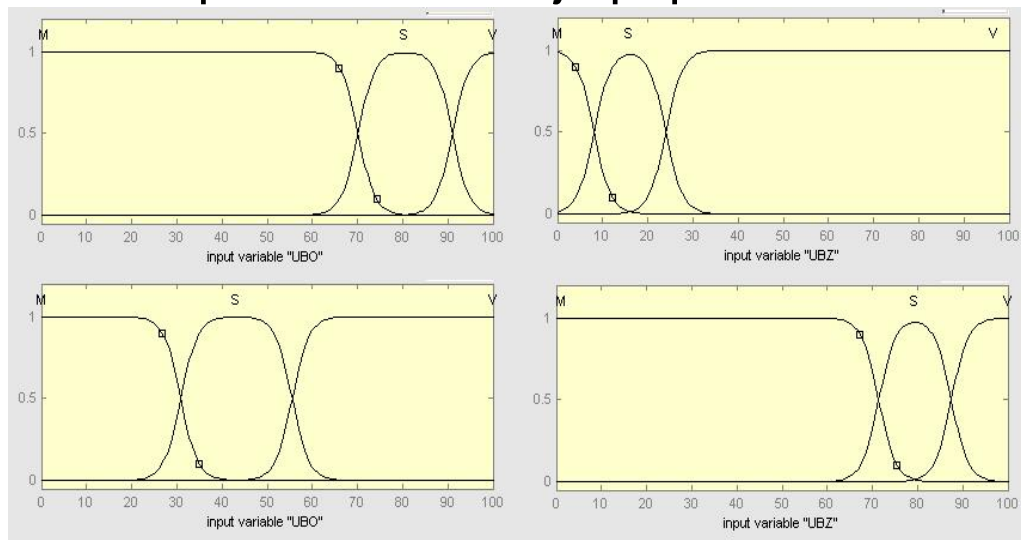
- ❖ Upotreba brodske prevodnice – kompromis između smanjenja vremena čekanja plovila i racionalnog korišćenja prevodnice (štednja opreme, struje i vode, a samim tim i para)



Inicijalne funkcije pripadnosti



Optimalne funkcije pripadnosti



Realizacija sistema za podršku u odlučivanju pri upravljanju brodskom prevodnicom zasnovanog na rasplinutoj (fuzzy) logici 2/2

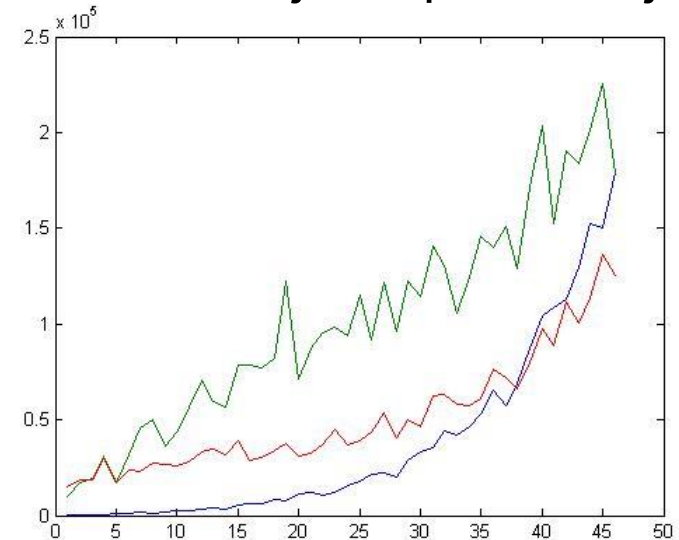
Ukupno vreme čekanja na prevođenje

Min. vreme čekanja na prevođenje (želje broдача) FIFS – First In First Served

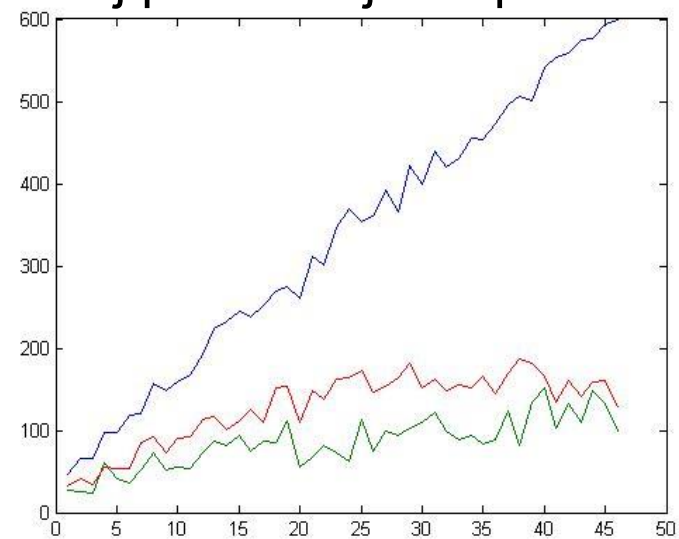
Min. broj prevođenja (želje prevodničara)

FSO

Simulacije za 5 do 50 brodova na dan



Broj prevođenja na prazno



❖ Kodiranje

❖ Realni brojevi

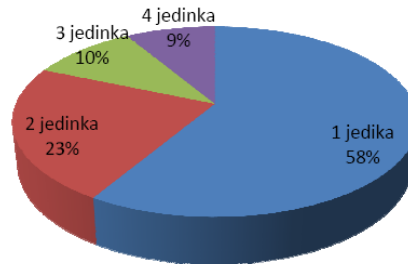
❖ Prilagođenost (kriterijum optimalnosti)

$$E = A \cdot \text{broj_prevodjenja_na_prazno} + B \cdot \text{prosecno_vreme_cekanja}$$

Pronalaženje optimalne putanje za ski stazu upotrebom genetskih algoritama 1/2

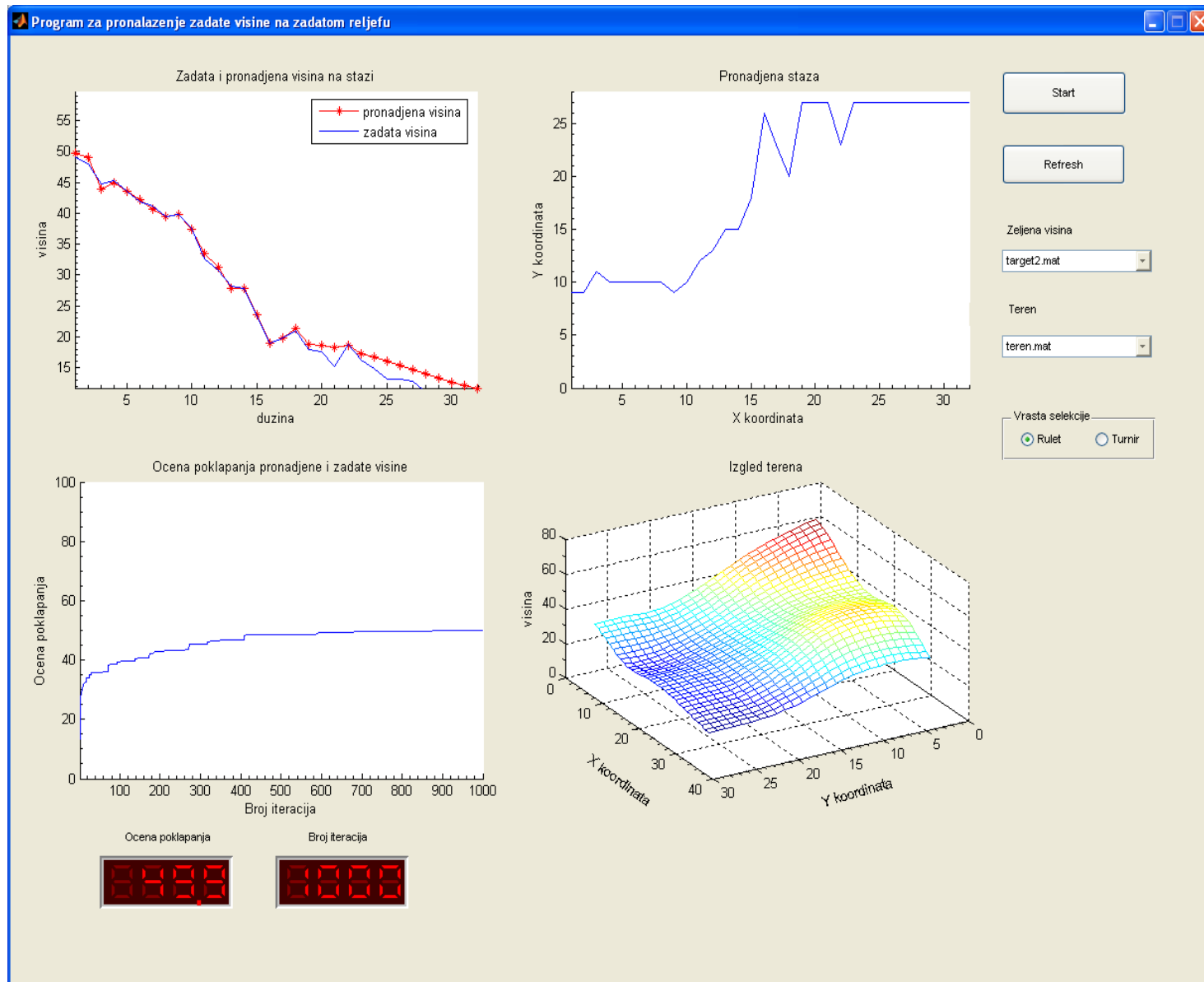
- Tereni su u formi matrica koje sadrže nadmorske visine na merenim koordinatama, a zadate staze su u formi vektora koji sadrže željene nadmorske visine na optimalnoj stazi.
- Populacija se sastoji od 100 jedinki koje predstavljaju moguće staze
- Kodiranje – X i Y koordinate putanje
- Prilagođenost – srednja kvadratna greška skalirana na [0,100]
- Selekcija – rulet ili turnir

$$\frac{\sum_{i=1}^{dužina} \sqrt{(jedinka \cdot visina_i - target_i)^2}}{dužina_{hromozoma}}$$



- Ukrštanje – ograničeno u jednoj tački
- Mutacija – ograničena izmena jedne koordinate putanje na slučajno izabranom mestu

Pronalaženje optimalne putanje za ski stazu upotrebom genetskih algoritama 2/2



Veličina populacije	100
Broj iteracija	1000
Prag ukrštanja	0.9
Prag mutacije	0.01
Broj igrača turnirske selekcije	5

Podešavanje parametara regulatora

- Podešavanje P, I i D dejstva kod PID regulatora
- Podešavanje parametara Fuzzy regulatora
 - pojačanja regulatora na ulazima i izlazima regulatora
 - broj funkcija pripadnosti
 - tip funkcija pripadnosti
 - parametri funkcija pripadnosti
 - tabela pravila
 - tip regulatora (Mamdani, Takagi-Sugeno, ...), tip implikacije, agregacije, defazifikacije, ...
- Kriterijumska funkcija se obično računa iz odziva sistema
 - vreme uspona
 - vreme smirenja
 - preskok
 - greška u ustaljenom stanju
 - indeks performanse (IAE, ISE, ITAE, ITSE)
 - Itd.
- Često je prilagođenost suma više kriterijuma ponderisana težinama.

KRAJ