

Umjetna inteligencija

Genetski algoritam

Doc. dr. sc. Edouard Ivanjko

- **Sadržaj**

- Uvod
- Optimizacijski problemi
- Princip evolucije
- Koncept genetskog algoritma
- Kôdiranje rješenja
- Genetski operatori
- Primjer primjene

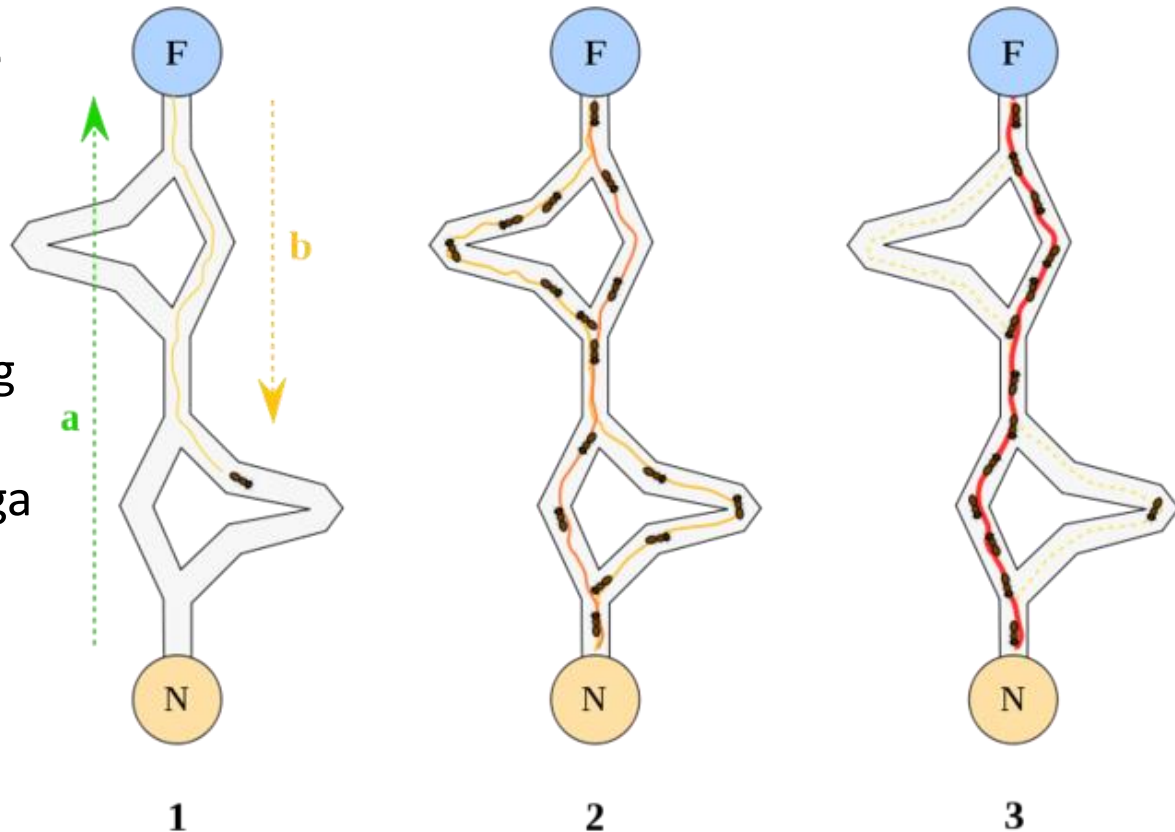
- Razvoj umjetne inteligencije se jako oslanja na primjeni načela živih organizama
 - Ljudska civilizacija je kreirala zakone koji sadrže razna pravila ponašanja
 - Najjednostavniji sustav umjetne inteligencije je zaključivanje pomoću pravila
 - Ljudska komunikacija koristi lingvističke varijable pune neodređenosti
 - Neizrazita logika modelira neodređenosti
 - Uključivanjem neizrazite logike u pravila zaključivanja nastao bolji sustav zaključivanja

- Svi organizmi na Zemlji teže poboljšanju sebe, a neki prilagođavaju i svoju okolinu
 - Koncept: brže, više, jače
 - Roditelji žele da njihova djeca budu bolja od njih i da su bolje prilagođene okolini u kojoj žive
 - Roditelji žele predati svoje najbolje značajke svojoj djeci
 - Neki roditelji u tome uspiju, a neki ne
 - Odvija se konstantna evolucija organizama koji traže optimalno rješenje za preživljavanje u svojoj okolini
 - Kako taj koncept iskoristiti u tehničkim (prometnim) sustavima?

- Problem pronalaska najkraćeg puta i optimalnog prometnog rješenja je vrlo bitan u prometu i transportu

– Najbolje rješenje našli mravi

- Algoritam kolonije mrava nastao imitacijom ponašanja mrava
 - Kreiranje vlastitog i slijeđenje feromonskog traga drugih mrava



- Optimizacija je postupak pronalaska optimalnog rješenja zadanog problema
 - Definiran problem
 - Definirana ograničenja
 - Definirana ocjena kvalitete rješenja
 - Kriterijska funkcija ili funkcija dobrote (engl. fitness function)
- Optimalno rješenje jest najbolje rješenje prema korištenoj ocjeni kvalitete rješenja i danim ograničenjima
 - Jesu li najkraći, najbrži, najjeftiniji, najsigurniji put uvijek isti?

- Mogući pristupi optimizaciji
 - Tehnika grube sile ili iscrpna pretraga (engl. brute-force)
 - Samo za male jednostavne probleme s malim prostorom/skupom mogućih rješenja
 - Heuristike odnosno algoritmi koji zadovoljavajuće brzo i zadovoljavajuće dobro mogu riješiti problem
 - Nema garancije za pronalazak optimalnog rješenja
 - Metaheuristike su skup algoritamskih koncepata za definiranje heurističkih metoda za rješavanje širokog skupa problema
 - Evolucijski algoritmi koji poboljšavaju postojeći skup rješenja u potrazi za najboljim rješenjem
 - » Nema garancije konvergencije prema optimalnom rješenju

- Tijekom optimizacije pretražuje se prostor svih mogućih rješenja i traži se najbolje rješenje
- Kod složenih problema prostor rješenja jako velik
 - Vrijeme izračuna raste jako brzo
- Primjer problema trgovačkog putnika
 - Svodi se na pronalazak Hamiltonovog ciklusa u grafu
 - Ima faktorijsku složenost (NP težak problem)
 - Pregled vremena izračuna primjenom iscrpnog pretraživanja
 - Obilazak 12 gradova -> x vremena
 - Obilazak 14 gradova -> 12,5x vremena
 - Obilazak 16 gradova -> 33840x vremena

- Algoritam optimizacije traži najmanju ili najveću vrijednost kriterijske funkcije
 - Najbrži put
 - Put s najvećom brzinom putovanja (max)
 - Put s najkraćim vremenom putovanja (min)
- Isti problem se može riješiti traženjem najveće odnosno najmanje vrijednosti
 - Koristi se inverz kriterijske funkcije ili promjena predznaka

- Kriterijska funkcija se uvijek prilagođava problemu koji se rješava
 - Potrebno obratiti pažnju na fizikalna/matematička svojstva varijabli koje se koriste
 - Npr. traženje najmanjeg odstupanja između točne i procijenjene (estimirane) vrijednosti
 - Točna procjena ima pogrešku 0

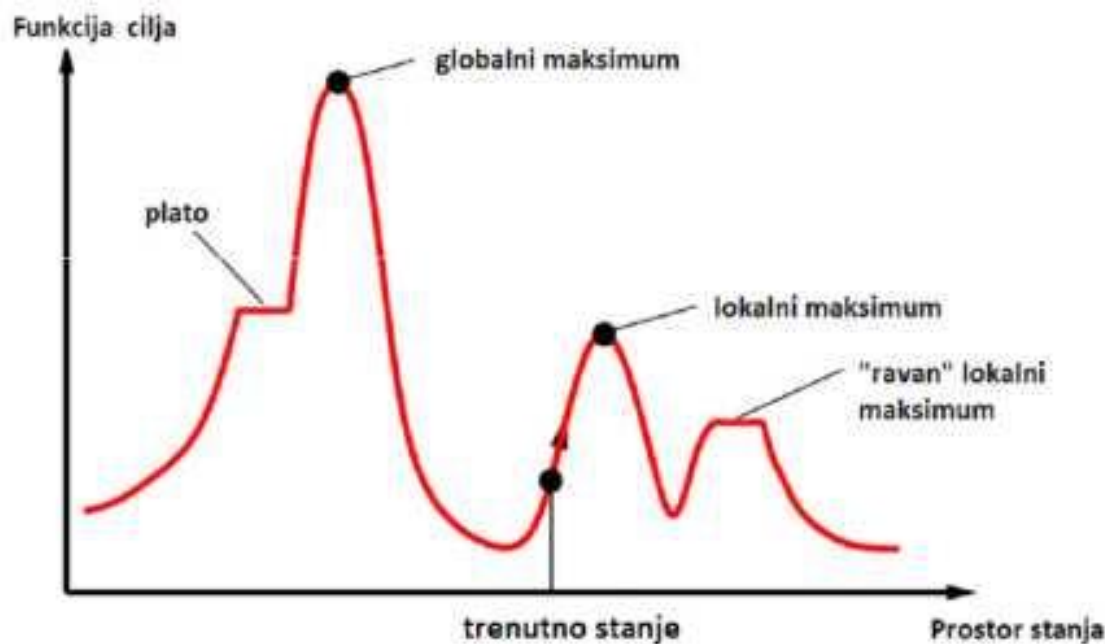
$$\mathcal{E}_1 = x - \hat{x}$$

» Odstupanje teži u $-\infty$

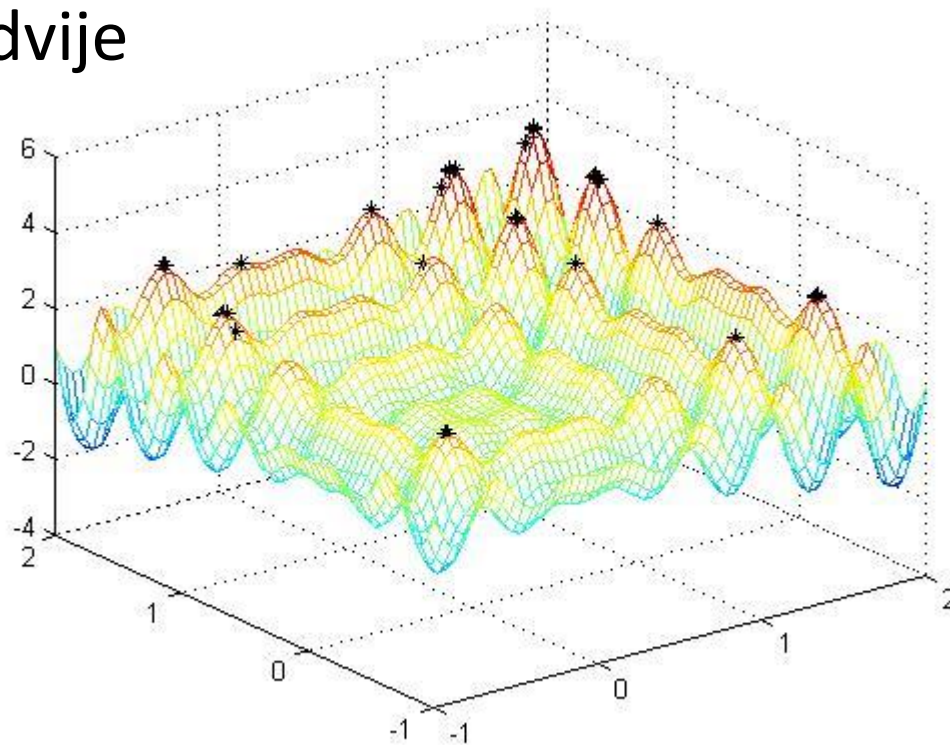
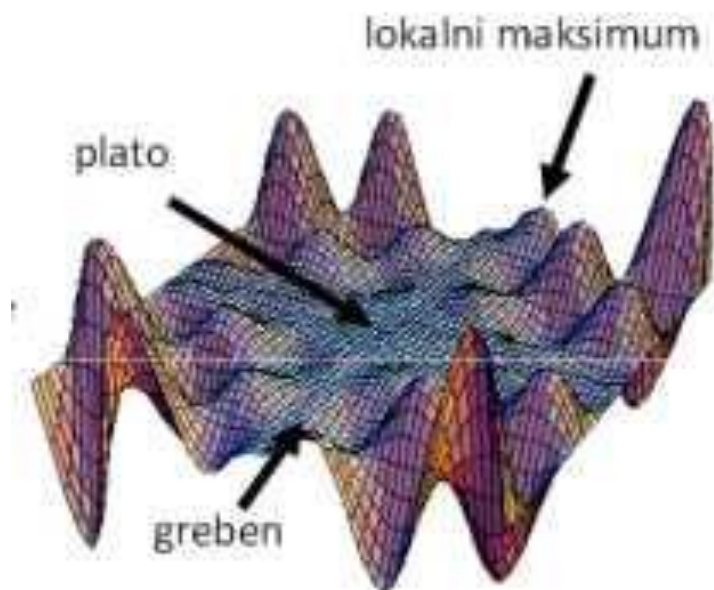
$$\mathcal{E}_2 = |x - \hat{x}|$$

» Odstupanje teži u 0

- Kod pretraživanja prostora rješenja često postoji više bliskih istaknutih većih/malih vrijednosti kriterijske funkcije
 - Ovisno tražimo li najveću ili najmanju vrijednost
- Problem lokalnog i globalnog optimuma



- Optimalno rješenje nepoznato i ovisi o više varijabli koje optimiramo
 - Svaka varijabla predstavlja jednu dimenziju prostora rješenja
 - Primjer problema s dvije varijable



- Procedura optimizacije može zapeti u lokalnom optimumu
 - Svako malo lokalno udaljavanje od trenutno optimalnog rješenja ima manju vrijednost kriterijske funkcije
 - Povratak u lokalni optimum
 - Veće udaljavanje od trenutno optimalnog rješenja može osigurati konvergenciju prema globalnom optimumu
- Ovaj problem se može izbjeći dovoljno velikim skupom razmatranih rješenja ili izbacivanja jednog rješenja iz podskupa sličnih razmatranih rješenja
 - U slučaju više lokalnih optimuma dolazi do grupiranja dijelova skupa rješenja

- Osnove teorije evolucije postavio je Charles Darwin u svom djelu „Postanak vrsta”
 - Organizmi su prilagođeni životu u prirodi
 - Prilagodbu organizama uzrokuju procesi križanja i mijenjanja gena u kromosomima
 - Radi ograničenosti prirodnih bogatstava preživljavaju samo najспособniji (selekcija)
 - Roditelji predaju svojoj djeci svoje značajke i djeca su slična svojim roditeljima
 - Postoje male razlike zbog spajanja gena
 - Mutacija može uzrokovati veće razlike

- Postavke Darwinove teorije razvoja vrsta
 1. Potomaka uvijek ima više no što je potrebno
 2. Veličina populacije je približno stalna
 3. Količina hrane je ograničena
 4. Kod vrsta koje se seksualno razmnožavaju, nema identičnih jedinki već postoje varijacije
 5. Najveći dio promjena jedinke prenosi se nasljeđivanjem
- Zbog ograničenosti hrane preživjeti će samo jedinke koje se mogu najbolje prilagoditi novonastalim promjenama

- Jedinke koje su najsposobnije imati će priliku razmnožavati se i predati svojoj djeci svoje dobre značajke
- Manje sposobne jedinke neće imati priliku razmnožavati se i njihove značajke nestaju
- Nesposobne jedinke umiru uslijed nedostatka hrane, predatora i neprilagođenosti okolini u kojoj žive
 - Može li se princip evolucije iskoristiti u pronalasku optimalnog rješenja složenog problema?

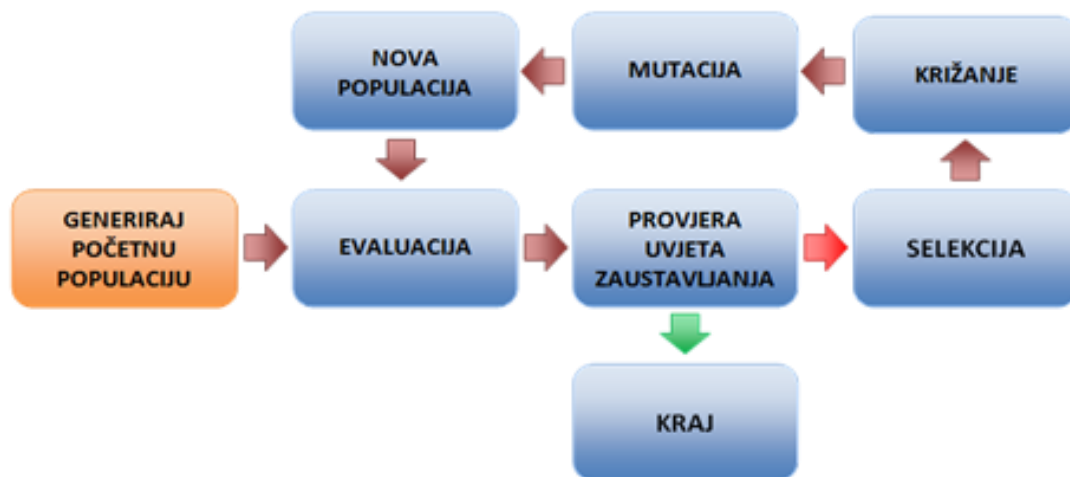
- Populacija jedinki predstavlja skup postojećih rješenja problema za optimizaciju
 - Skup rješenja najčešće ima stalan broj članova
- Nova rješenja problema kreiraju se iz postojećih rješenja
 - Dva rješenja kako dva roditelja stvaraju dva nova rješenja kao svoju djecu
 - Malim promjenama jednog postojećeg rješenja nastaje jedno novo rješenje
- Kriterijska funkcija određuje kvalitetu pojedinog rješenja
 - Sva rješenja se rangiraju
 - Samo najbolja rješenja ostaju u skupu rješenja ili kreiraju potpuno novi skup rješenja

- Koncept genetskog algoritma (GA) zasnovan na konceptu evolucije
- Stohastički (slučajno) pretražuje prostor rješenja na temelju “prirodnog” izbora najsposobnijih kandidatnih rješenja
- Tijekom pretraživanja skup rješenja (populacija) se mijenja



- Skup rješenja se može u svakoj iteraciji potpuno izmijeniti

– Generacijski GA



- Skup rješenja može u svakoj iteraciji zadržati najbolja rješenja

– Eliminacijski GA





- Pseudokôd općenitog genetskog algoritma

Generiraj slučajni skup početnih rješenja veličine N

Dok je (rješenje loše) ILL

(broj iteracija < dopuštenih iteracija)

Selektiraj najbolja rješenja

Primjeni genetske operatore na odabrana rješenja uz provjeru ispravnosti novog rješenja

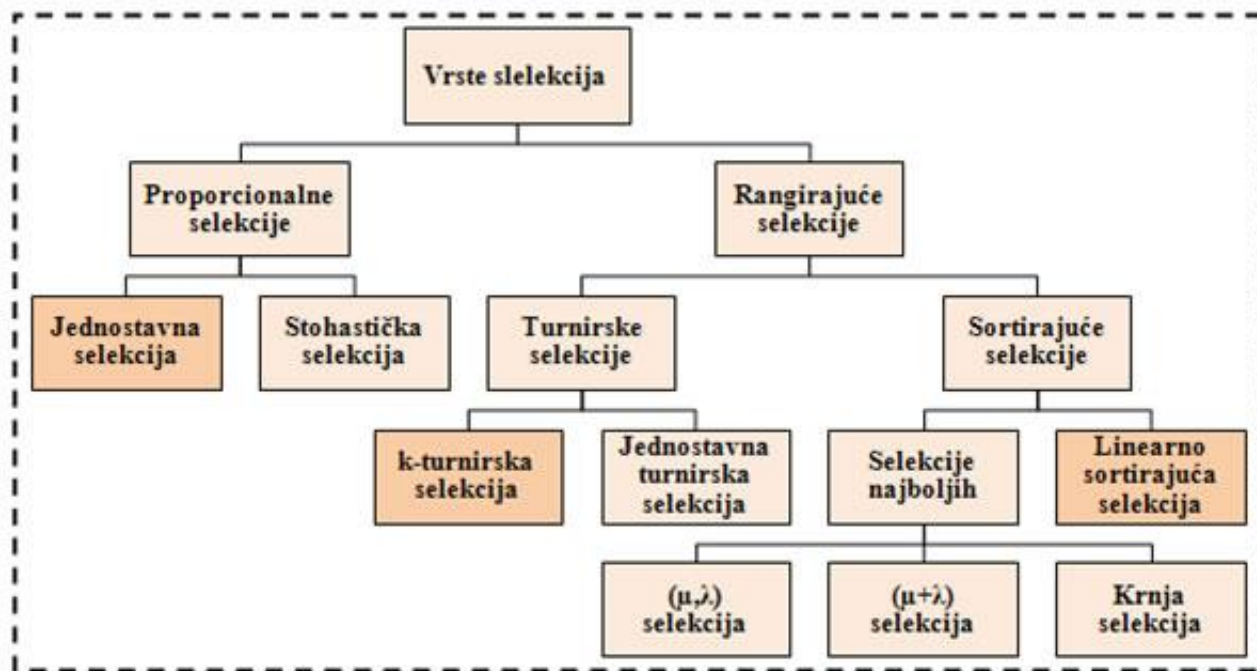
Križanje

Mutacija

Inverzija

Ispiši najbolje rješenje

- Selekcija je proces odabira rješenja u GA radi
 - Očuvanja dobrih rješenja
 - Odbacivanja loših rješenja
- U procesu selekcije se često koristi elitizam
 - Očuvanje najboljeg rješenja

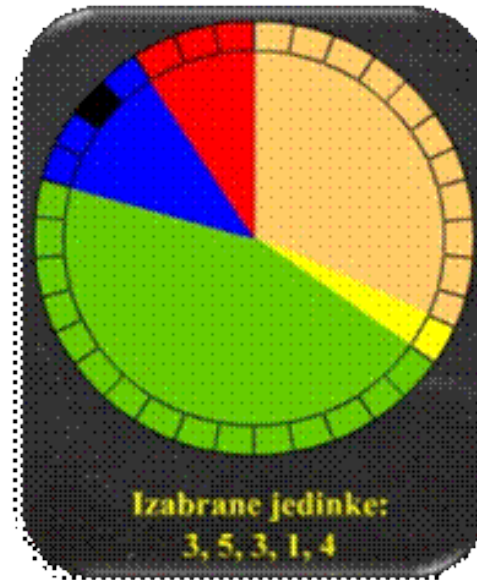
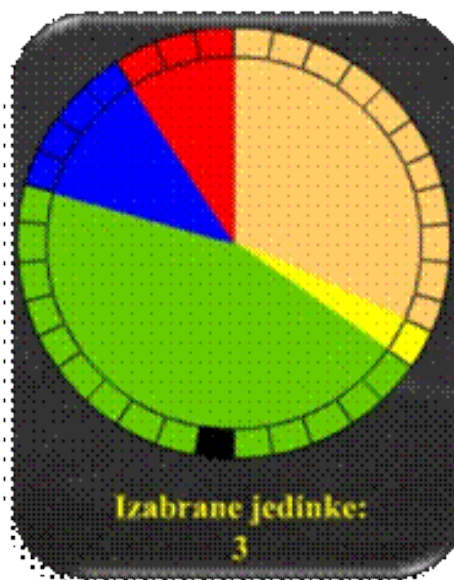
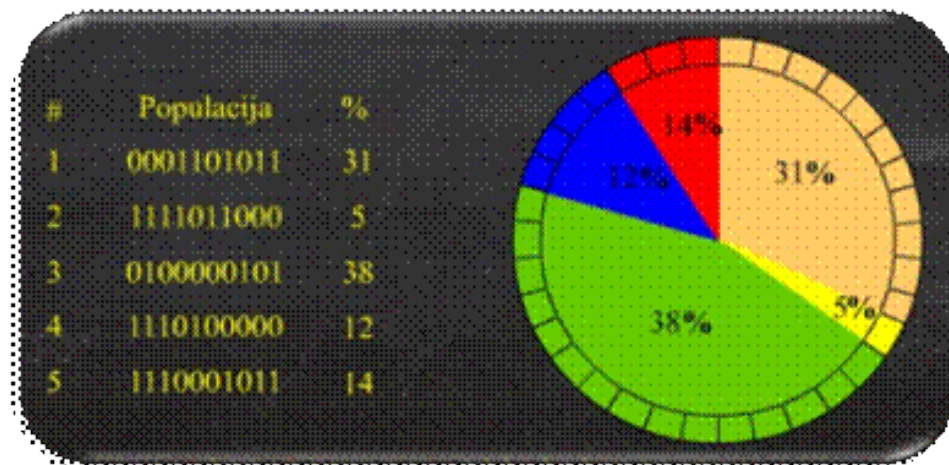


- Jednostavna selekcija
 - Naziva se i metoda ruleta (engl. roulette-wheel selection)
 - Vjerojatnost odabira pojedine jedinice (rješenja) ovisi o iznosu dobrote te jedinice
 - Određuje se kumulativna dobrota te omjer dobrote jedinice i ukupne dobrote populacije
$$q_k = \sum_{i=0}^k \text{dobrota}(v_i), \quad k = 1, 2, \dots, VEL_POP;$$
$$p_k = \frac{\text{dobrota}(v_k)}{\sum_{i=0}^N \text{dobrota}(v_i)}$$
 - Generira se N slučajnih brojeva i odabire se rješenje u čiji interval kumulativne dobrote je upao generirani broj
 - Neučinkovita kod malih razlika u kvaliteti rješenja

• Jednostavna selekcija

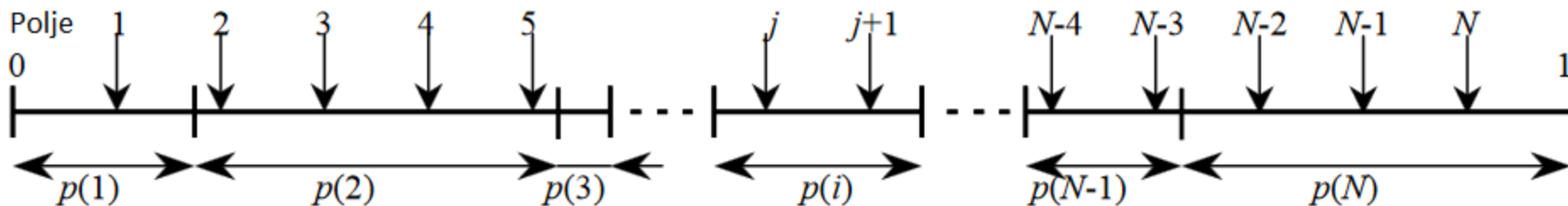
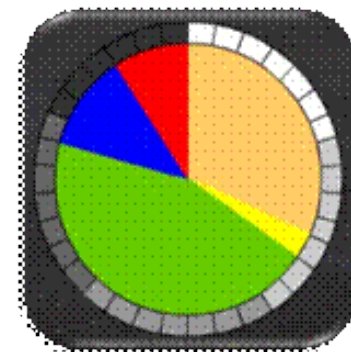
– Primjer

- Iznos dobrote generira raspodjelu rješenja u obliku ruleta
- Generator slučajnih brojeva označava polje ruleta
- Odabire se rješenje u čijem području je označeno polje ruleta

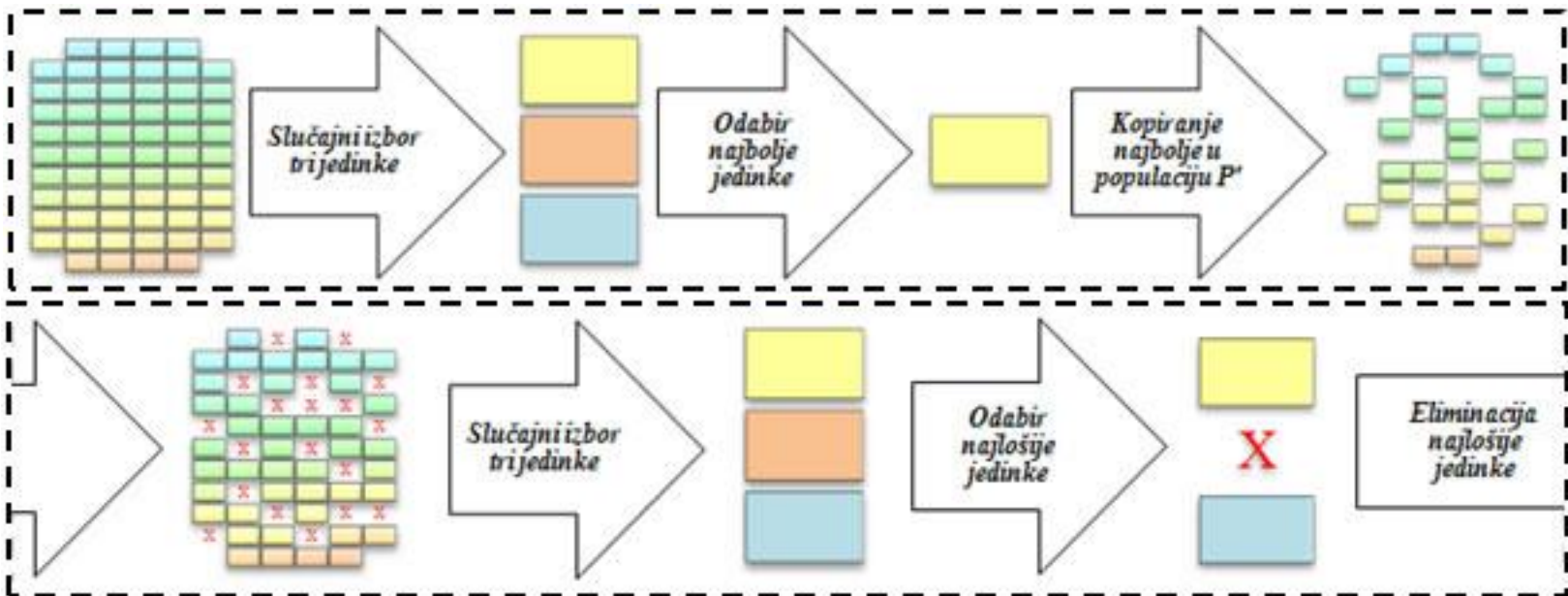


- Stohastička selekcija

- Slična jednostavnoj selekciji
- Generira se ista raspodjela rješenja i kreira se rulet s N polja
- Odabiru se rješenja u čijem području kumulativne dobrote se nalazi polje ruleta
- Rješenje se također odabire onoliko puta koliko puta se u njegovom području kumulativne dobrote nalazi polje ruleta



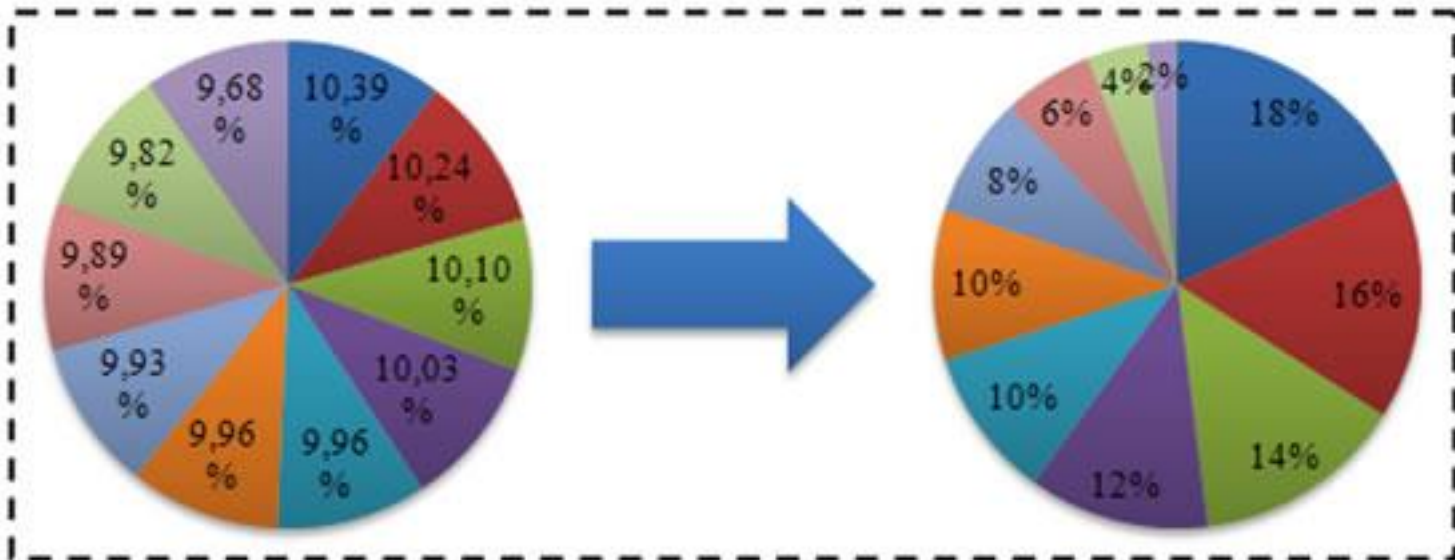
- Turnirska selekcija
 - Slučajnim odabirom određenog broja rješenja se radi turnir između njih
 - Odabire se najbolje ili se odbacuje najlošije rješenja
 - Jednostavan turnir je između dva rješenja
 - K -turnirska selekcija radi turnir između k rješenja
 - Primjer 3-turnirske selekcije



- Selekcija najboljih
 - Odabire se unaprijed zadani broj najboljih jedinki
 - Tri vrste
 - $\mu+\lambda$ selekcija
 - Slučajno se odabire μ roditelja kako bi se stvorilo λ djece
 - Zatim se iz dobivenog skupa rješenja odabire μ rješenja za novi skup rješenja
 - (μ, λ) selekcija
 - Slučajno se odabire μ roditelja kako bi se stvorila λ djece
 - Vrijedi $\lambda \geq \mu$ pa se odabire μ najbolje djece za novi skup rješenja
 - Krnja selekcija
 - Odabire n najboljih jedinki i kopira ih N/n puta u novi skup rješenja

- Linearno sortirajuća selekcija
 - Vjerojatnost selekcije je proporcionalna rangu (poziciji) jedinke u poretku jedinki sortiranih po dobroti
 - Najbolje rješenje ima indeks N , a najlošije 1
 - Vjerojatnost odabira pojedinog rješenja

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



- Informacije o živim bićima sadržane su u niti deoksiribonukleinske kiseline (DNK)
 - Sastoji se većinom (98 %) od kromosoma
 - Kromosomi se sastoje gena
- Na isti način je potrebno kôdirati skup rješenja
 - Jedan kromosom biti će jedno rješenje
- Računala mogu obrađivati samo brojeve i potrebno je rješenje kôdirati brojevima
 - Binarnim brojevima
 - Prirodnim brojevima
 - Realnim brojevima
 - Permutacijama



- Byte ili više njih predstavlja kromosom, a bit gen
 - Gen ima samo dva moguća stanja 0 i 1
- Pojedini kromosom može predstavljati cijeli broj, realni broj, naredbu ili simboličku vrijednost

- Pretvaranje binarne vrijednosti u cijeli dekadski broj

$$a = \sum_{i=0}^n a_i \cdot 2^i$$

- Preciznost prikaza ovisi o broju bitova k

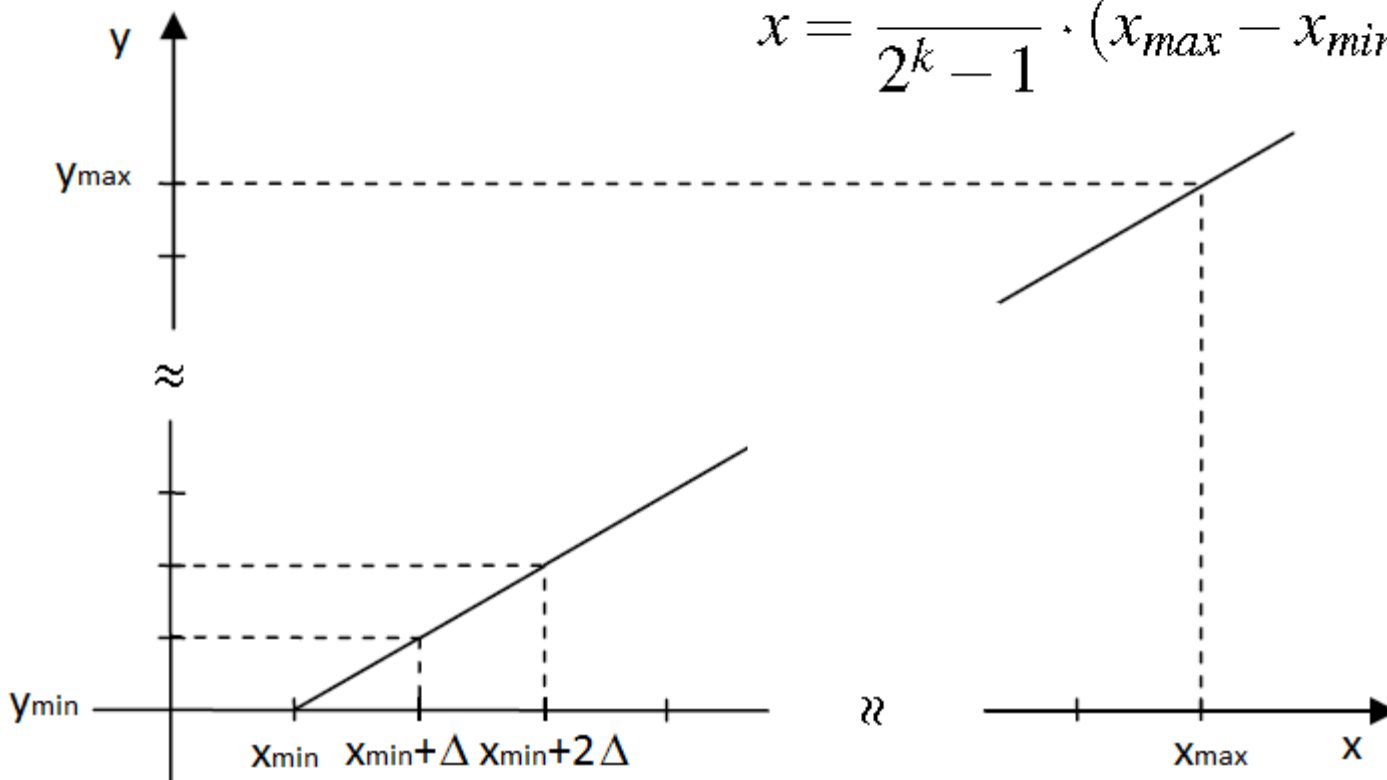
$$\Delta = \frac{1}{2^k - 1} \cdot (x_{max} - x_{min})$$

- Broj bitova potreban za precizan prikaz p decimala

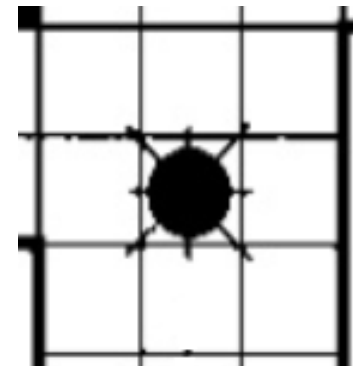
$$k \geq \frac{\log \left[(x_{\max} - x_{\min}) \cdot 10^p + 1 \right]}{\log 2}$$

- Prikaz područja

$$x = \frac{a}{2^k - 1} \cdot (x_{\max} - x_{\min}) + x_{\min}$$



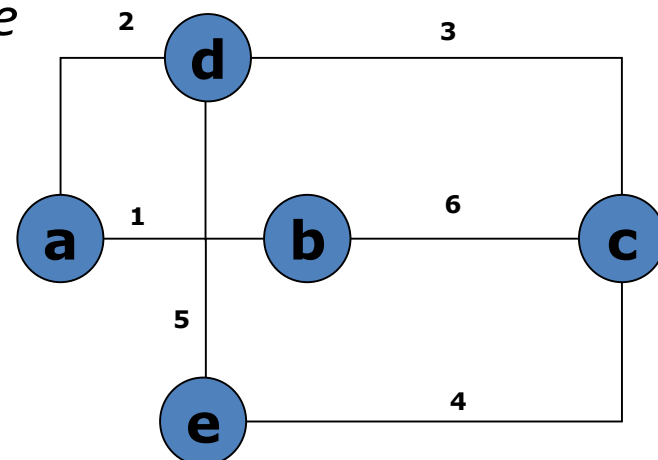
- Koriste se za kôdiranje rješenja kada se rješenja mogu predstaviti cijelim brojevima
- Dodatna primjena je za kôdiranje naredbi
 - Primjer pomicanja mobilnog robot po mreži polja
 - Kôd naredbe pomaka
 - 0 -> naprijed
 - 1 -> dijagonalno naprijed desno
 - 2 -> dijagonalno naprijed lijevo
 - 3 -> natrag
 - 4 -> dijagonalno natrag desno
 - 5 -> dijagonalno natrag lijevo
 - 6 -> ostani na mjestu
 - 7 -> desno
 - 8 -> lijevo





- Kôdiranje analogno kôdiranju pomoću binarnih i prirodnih brojeva za slučajeve kada se želi prirodnije prikazati skup rješenja iz neprekidne distribucije
 - Cijeli i binarni brojevi prikazuju rješenja iz diskretne distribucije
 - Realni brojevi preciznije prikazuju skup rješenja iz neprekidne distribucije

- Kôdiranje permutacijama se koristi kada je potrebno u rješenju prikazati slijed događaja
 - Pojedini događaj se kôdira prirodnim brojem
 - Dva načina kôdiranja
 - i -ti element niza reprezentira događaj koji se dogodio i -ti po redu
 - Vrijednost i -tog elementa reprezentira poziciju na kojoj se dogodio i -ti događaj
 - Bitan prikaz u prometu i transportu za prikaz rute
 - Primjer prikaza ruta između čvorova a i e
 - Prvi način kôdiranja
 - [1,6,4]
 - [2,3,4]
 - [2,5]
 - [1,6,3,5]



- Genetski operatori služe za unos promjena u postojeći skup rješenja
- Na osnovu postojećih rješenja (roditelja) kreiraju nova rješenja (djecu)

- Vrste

- Križanje

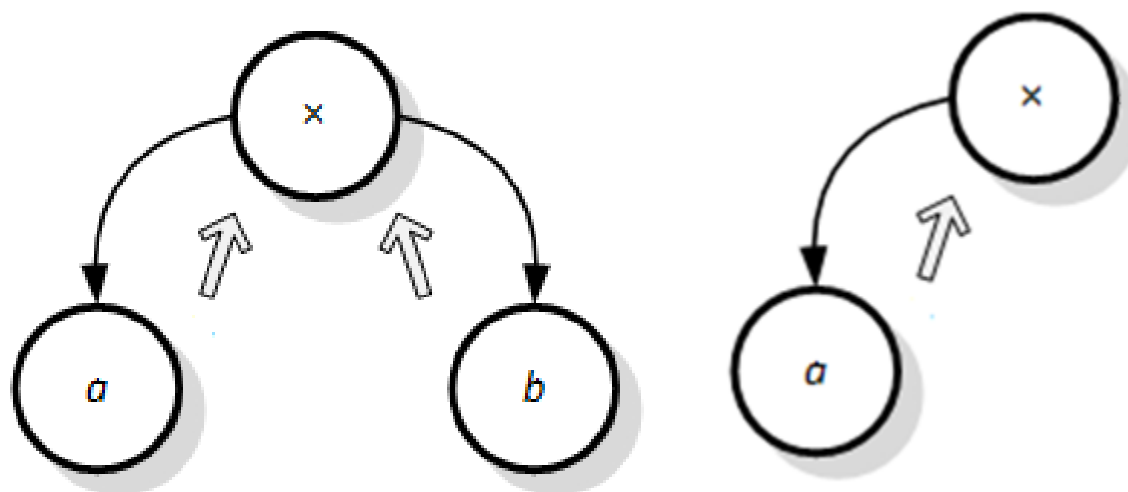
- Binarni

- Mutacija

- Unarni

- Inverz

- Unarni

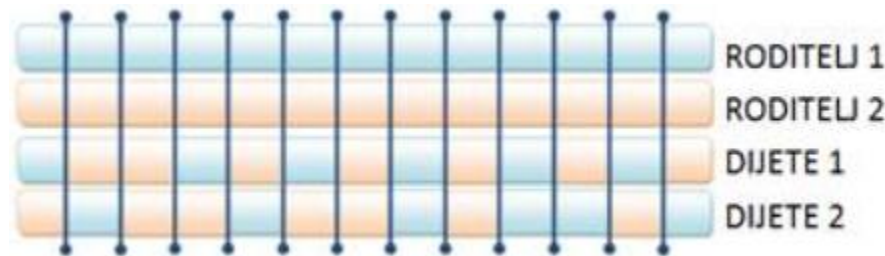


- Binarni operator -> dvoje roditelja stapanjem daje dvoje djece
- Imitira prirodnu spolnu reprodukciju
- Križanjem se postiže brza konvergencija optimumu
 - Opasnost da optimizacija završi u lokalnom optimumu
 - Djeca zadržavaju dobra svojstva roditelja



- Uniformno križanje

- Vjerojatnost križanja svakog gena je 0,5
- Svako dijete sadrže pola gena pojedinog roditelja



- Križanje s n točaka prekida

- Slučajno se odabire jedno ili više mjesta križanja
- Primjer križanja s dvije točke prekida

Roditelj1 -> 11 00010 101	→	11 00100 101 -> Dijete1
Roditelj2 -> 00 00100 010		00 00010 010 -> Dijete2

- Aritmetičko križanje

- Koristi se kod prikaza rješenja u obliku realnih brojeva
- Križanje obavlja prema izrazima

$$Dijete1 = a * roditelj1 + (1-a) * roditelj2$$

$$Dijete2 = (1-a) * roditelj1 + a * roditelj2$$

- Parametar a je slučajni težinski koeficijent iz $[0, 1]$
 - Određuje koliko će utjecaja na pojedino dijete imati pojedini roditelj

- Heurističko križanje

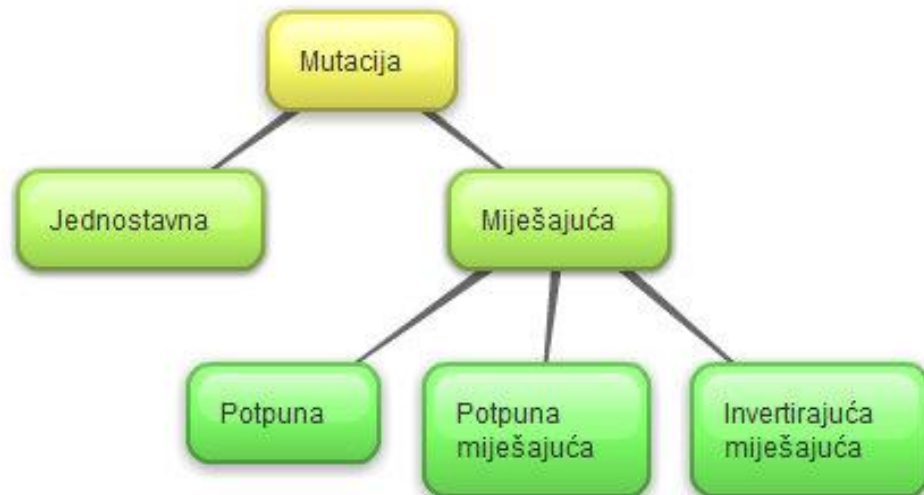
- Koristi se kod prikaza rješenja u obliku realnih brojeva
- Križanje obavlja prema izrazima

$$\text{Dijete 1} = \text{bolji roditelj} + r * (\text{bolji roditelj} - \text{lošiji roditelj})$$

$$\text{Dijete 2} = \text{bolji roditelj}$$

- Parametar r je slučajni težinski koeficijent iz $[0, 1]$
- Moguća pojava problema kod kreiranja prvog djeteta jer se može naći izvan intervala pretraživanja rješenja
 - Nakon n pokušaja generiranja koeficijenta r , lošiji roditelj postaje Dijete 1

- U prirodi je mutacija trajna promjena genetskog materijala zbog vanjskog utjecaja
- Unutar genetskog algoritam se radi ubacivanje novih nizova u pojedino postojeće rješenje
- Unarni operator koji djeluje na jedno rješenje
- U binarnoj prezentaciji radi se komplement pojedinog bita (gena)



- Jednostavna mutacija
 - Slučajno se odabire bit koji će se komplementirati
 - Moguće postaviti dodatne uvjete kada će se mutacija desiti
 - Mala promjena rješenja pa postoji opasnost da pretraživanje zapne u lokalnom optimumu
 - Jednostavna mutacija mijenja samo jedan bit
 - Primjer

Roditelj -> 1010010101



Dijete -> 1010011101

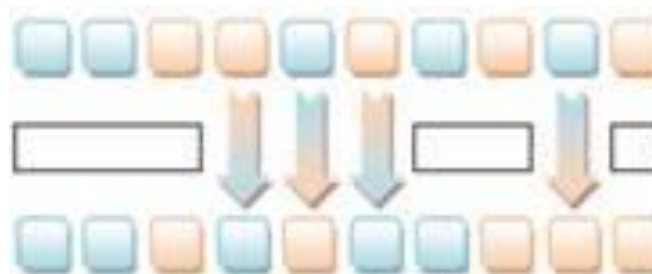
- Miješajuća mutacija
 - Svi geni iz odabranog intervala se međusobno pomiješaju tako da im se indeks trenutne pozicije u kromosomu zamijeni s nekim drugim indeksom u granicama $[0, \text{VELIČINA_KROMOSOMA}]$
 - Izmiješa se dva ili više gena
 - Broj jedinica i nula kromosoma ostaje nepromijenjen
 - Za miješajuću mutaciju moguće generirati slučajnu masku

001110001101010101010101111	JEDINKA
000000111000001100001110000	SLUČAJNA MASKA
001110100101010001011011111	MUTIRANA JEDINKA

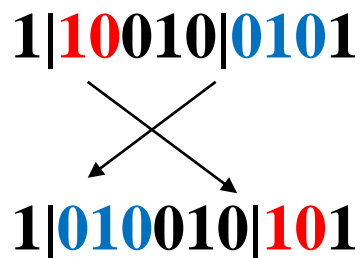
- Miješajuća mutacija
 - Potpuno miješajuća mutaciji slučajno generira vrijednosti odabranih gena

0011100011010101010101011111	JEDINKA
000000111111111100000010000	SLUČAJNA MASKA
0011101010001101010101111111	MUTIRANA JEDINKA

- Invertirajuća miješajuća mutacija invertira vrijednost gena u slučajno odabranom intervalu



- Premještanje dijela rješenja (gena) unutar niza
 - Mjesto premještanja i duljina premještanja se slučajno odabire
 - Operator pogodan kod rješavanja problema optimizacije ruta dostave
 - Mijenja redoslijed obilaska mjesta unutar rješenja
 - Primjer



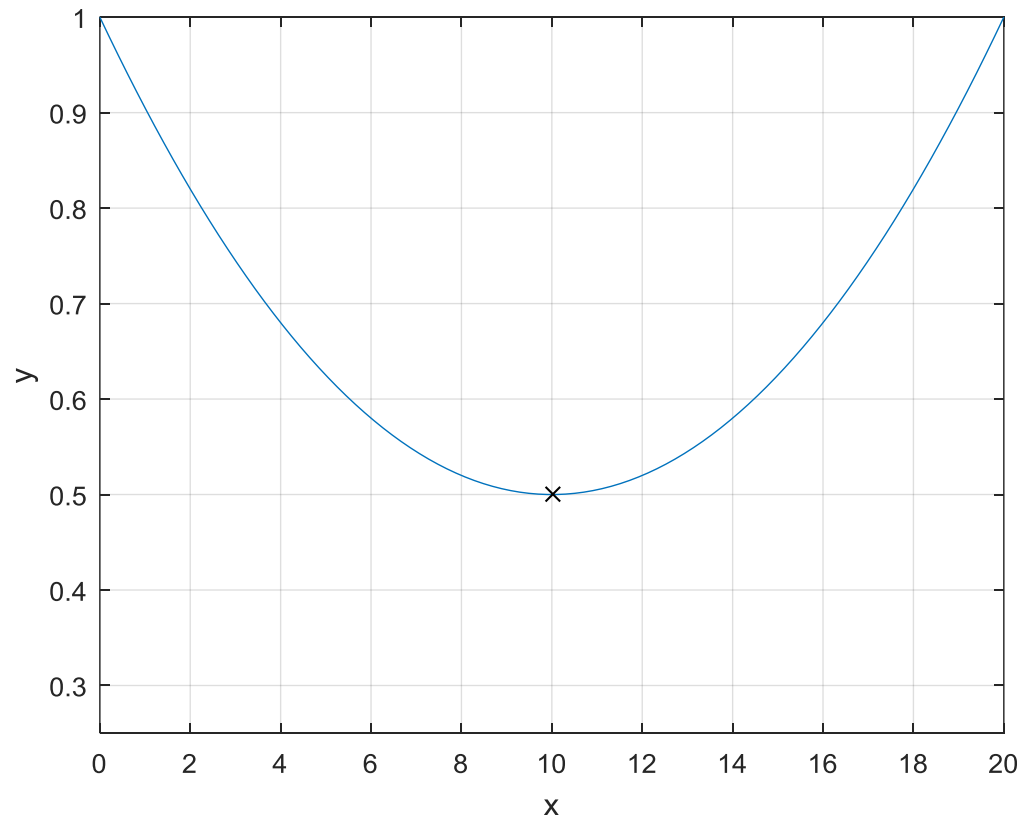
- Pronalazak optimalne vrijednosti funkcije

- Minimum funkcije

$$y(x) = 1 - \frac{1}{10}x + \frac{1}{200}x^2$$

- Rješenje

- $x = 10, y(x) = 0,5$

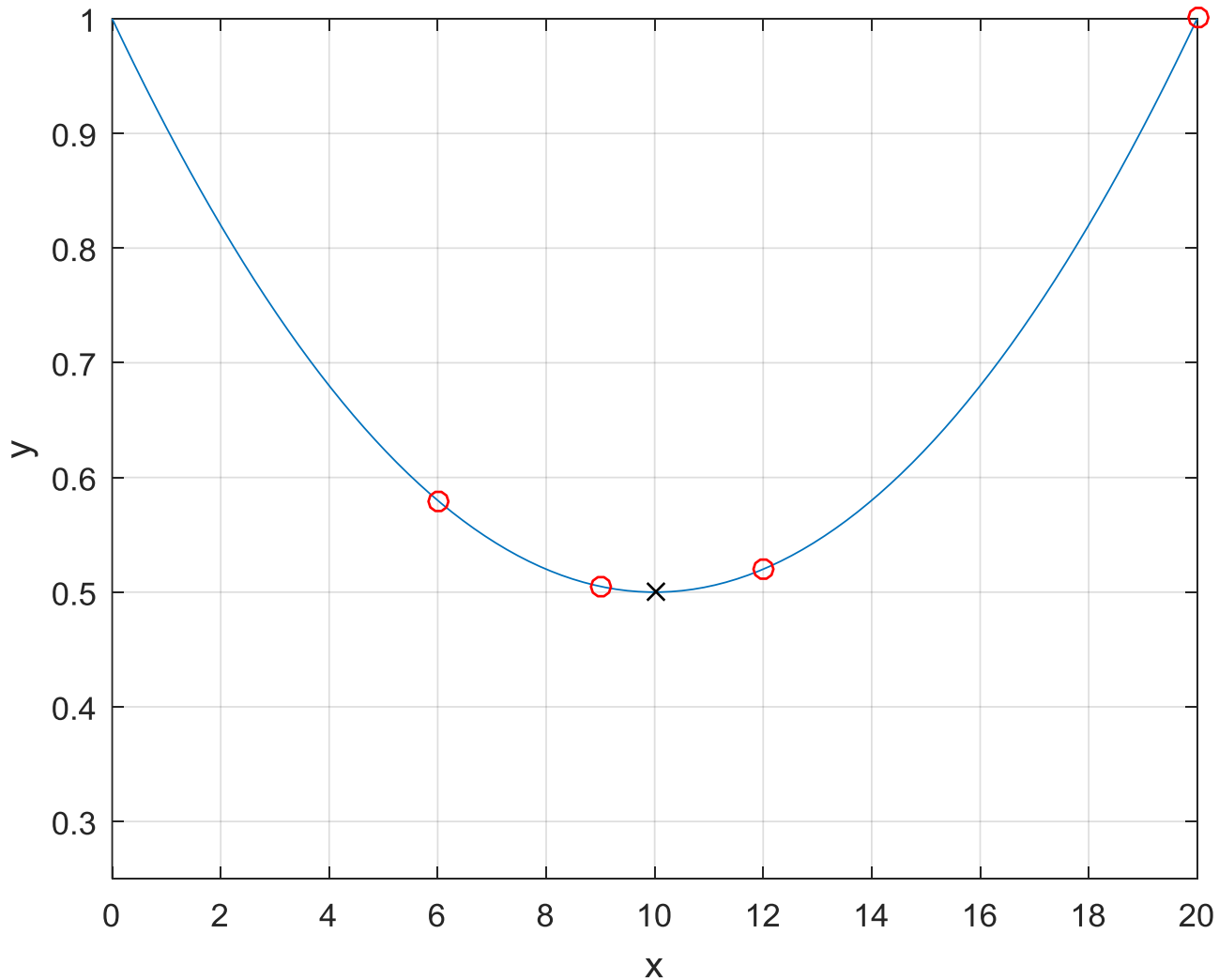


- Pronalazak optimalne vrijednosti funkcije
 - Populacija od 5 rješenja konstantne veličine
 - Koristi se binarno kôdiranje rješenja s 5 bitova
 - Funkcija dobrote je inverz funkcije
 - Minimum inverzom postaje maksimum
 - Selekcija se radi ruletom
 - Genetski operatori mutiranja i križanja

- Pronalazak optimalne vrijednosti funkcije
 - Početna populacija

	Niz	Vrijednost	$y_i(x)$	$f_i(x)$
V_1	10111	23	1,345	0,743
V_2	01100	12	0,520	1,923
V_3	10100	20	1,000	1,000
V_4	00110	6	0,580	1,732
V_5	01001	9	0,505	1,980
Zbroj				7,378
Prosječna				1,476
Najveća				1,980

- Pronalazak optimalne vrijednosti funkcije



- Pronalazak optimalne vrijednosti funkcije
 - Nova populacija

	Niz	Vrijednost	$y_i(x)$	$f_i(x)$
V_1	10000	16	1,471	0,680
V_2	01111	15	1,600	0,625
V_3	00101	5	0,625	1,600
V_4	00100	4	0,680	1,471
V_5	01010	10	0,500	2,000
Zbroj				6,375
Prosječna				1,275
Najveća				2,000

- Pronalazak optimalne vrijednosti funkcije

