

# Osnove genetskih algoritama

# Osnovna ideja

- genetski algoritmi inspirirani su evolucijom
- Evolucija:
  - jedinke neke vrste žive u nekom okolišu
  - svojstva jedinke ovise o njenim kromosomima
  - pojednostavljeno, kromosomi su dugački nizovi nukleotida (adenin, gvanin, timin, citozin)

# Osnovna ideja

- Evolucija:
  - kombinacije nukleotida određuju svojstva pojedine jedinke ("kromosom kodira jedinku")
  - evolucija "djeluje" na kromosome, a ne izravno na jedinke
  - evolucija sama, kao proces, nema "znanja" o uspješnosti jedinke
  - prirodna selekcija (odabir) "povezuje" kromosome (evoluciju) i uspješnost jedinke

# Osnovna ideja

- Evolucija:
  - prirodna selekcija utječe na to da će uspješnije jedinke (tj. bolji kromosomi) imati veću šansu za preživljavanje i razmnožavanje
  - “mehanizmi” evolucije jesu razmnožavanje i mutacije
  - razmnožavanje nove kromosome potomaka stvara kombiniranjem roditeljskih kromosoma
  - mutacija na slučajan način mijenja kromosom jedinke

# Osnovna ideja

- Evolucija:
  - evolucija nema "globalne" memorije: sva informacija sadržana je u kromosomima. Ako neki dobar (ili loš) kromosom ne uspije preživjeti, proces evolucije ga ne može povratiti natrag (osim slučajno, za što je mala vjerojatnost)

# Osnovna ideja

- inspiriran teorijom evolucije, genetske algoritme (kraće GA) izumio je John Henry Holland sredinom 70-ih
- Holland je primjenio ideje evolucije na rješavanje teško rješivih problema, pri čemu lošija početna rješenja evoluiraju prema sve boljim rješenjima
- rješenja, kao jedinke, predstavljena su (kodirana) nizovima 0 i 1 koje je Holland nazvao kromosomima

# Osnovna ideja

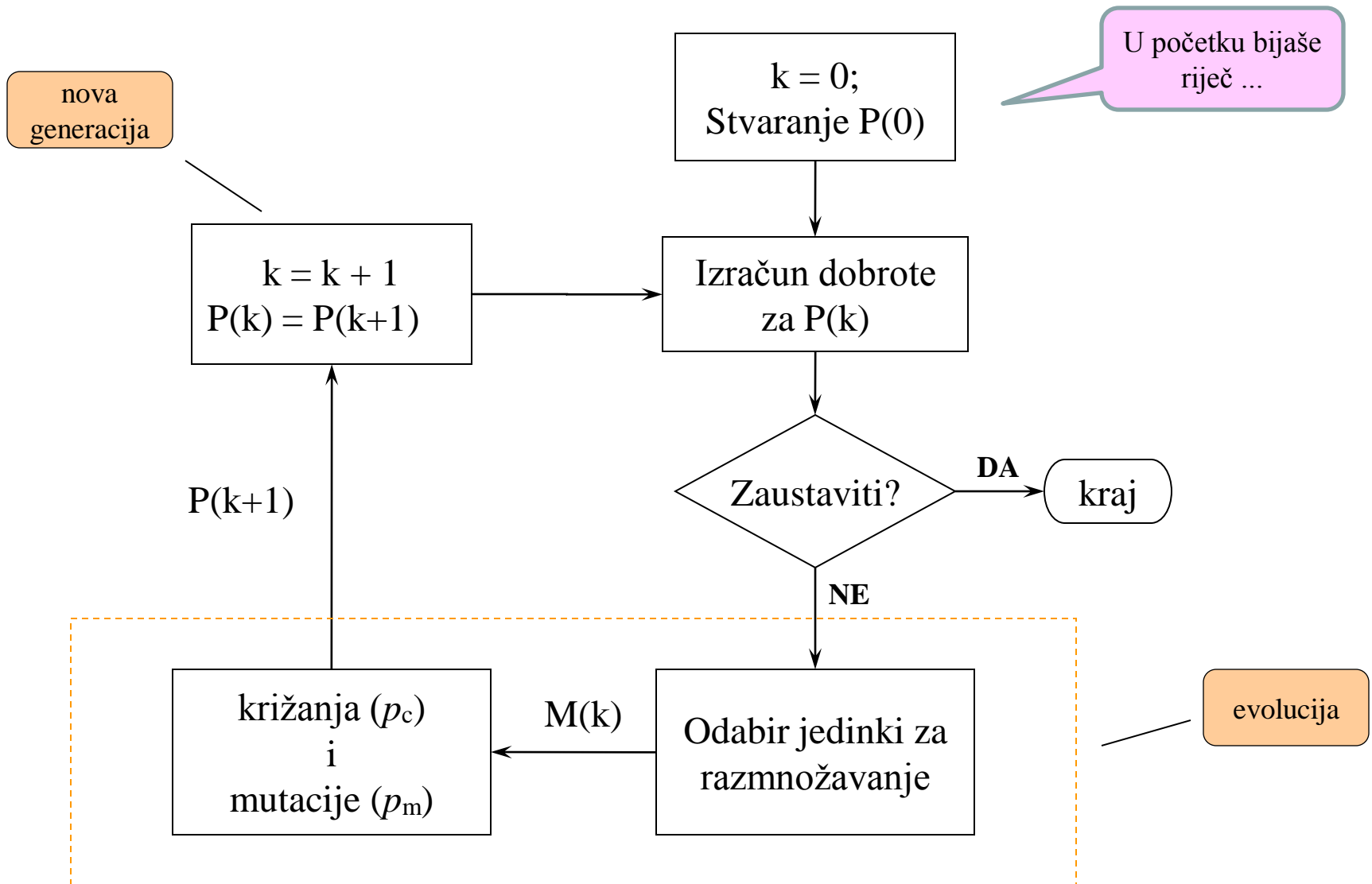
- Genetski algoritam sastoji se od:
  - populacije jedinki predstavljenih kromosomima
  - mehanizma reprodukcije
  - mehanizma vrednovanja uspješnosti jedinki

# Osnovna ideja

- Pseudo-kod osnovne ideje:
  1. inicijaliziraj početnu populaciju kromosoma
  2. vrednuj svaku jedinku
  3. koristeći postojeće jedinke (roditelje) proizvedi nove jedinke (djecu) pomoću križanja i mutacije i vrednuj svaku novu jedinku
  4. zamjeni dio prethodne populacije s novim jedinkama
  5. Ako je dosegnuto vremensko ograničenje: vrati najbolju jedinku kao rješenje; inače: ponavlja algoritam od koraka 3



# Općenita struktura genetskog algoritma



# Osnovna ideja

- Kromosom je niz nula i jedinica koji kodira moguće rješenje problema (način kodiranja je jedinstven za svaki konkretni problem)
- Funkcija vrednovanja ili dobrote (fitness) povezuje konkretan problem s genetskim algoritmom. Ona predstavlja uspješnost jedinke u konkretnom problemu.

# Osnovna ideja

- Postoje mnogobrojni načini za:
  - kodiranje rješenja kromosomom
  - provođenje križanja
  - mutiranje kromosoma
  - odabir jedinki za križanje
  - brisanje starih jedinki
  - vrednovanje jedinke
- svi ovi faktori utječu na ponašanje genetskog algoritma, tj. na kvalitetu algoritma i na njegove performanse
- dodatno: faktori utječu jedan na drugoga (nisu neovisni)
- za početak: jedan konkretan primjer kao ilustracija

# Primjer genetskog algoritma

(prema Lawrence Davis: "Handbook of Genetic algorithms")

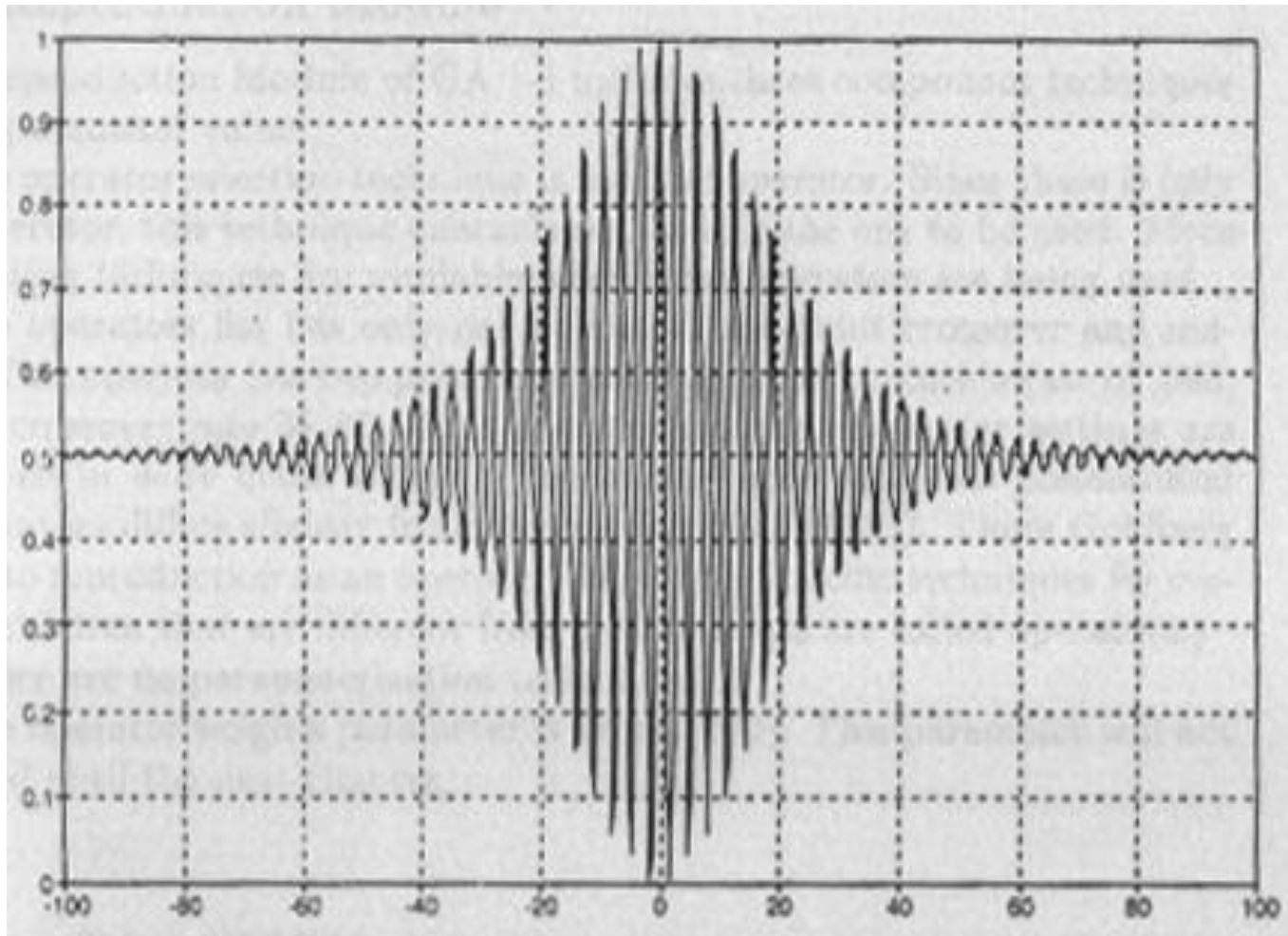
- želi se maksimizirati funkcija  $f_6$ :

$$f_6(x, y) = 0,5 - \frac{(\sin \sqrt{x^2 + y^2})^2 - 0,5}{(1 + 0,001(x^2 + y^2))^2}$$

- domena funkcije je ravnina, tj. parovi  $(x, y)$ , gdje su  $x$  i  $y$  realni brojevi, a traži se najbolji par (maksimum funkcije) u intervalu  $x, y \in [-100, 100]$
- s gledišta genetskog algoritma, svaki par  $(x, y)$  je jedna jedinka podložna evoluciji
- traženo najbolje rješenje je:  $x=0, y=0$

# Primjer genetskog algoritma

- Funkcija  $f_6$  je nezgodna za druge tehnike optimizacije jer ima puno lokalnih ekstrema; izgleda otprilike ovako (za optimalni  $y=0$ ):



# Primjer genetskog algoritma

- U ovom primjeru vrijedit će:
  - kromosom će koristiti binarnu reprezentaciju (*binary representation*): svaki kromosom će biti lista (niz) od 44 bita
  - inicijalizacija populacije bit će slučajna binarna inicijalizacija (*random binary initialization*)
  - odabir jedinki za križanje provodit će se proporcionalnom selekcijom (*roulette wheel parent selection*)
  - križanje će imati jednu prekidnu točku (*one-point crossover*) i mutaciju bita (*bit mutation*)
  - vjerojatnost križanja je 0,65, a vjerojatnost mutacije bita je 0,008
  - prilikom zamjene starih jedinki novima, primjenjuje se tzv. *generational replacement* - sve stare jedinke se brišu i zamjenjuju novima
  - uspješnost svake jedinke “mjeri se”, tj. jednaka je, vrijednosti funkcije  $f_6$  (*fitness is evaluation*)
  - veličina populacije bit će 100 jedinki
  - ukupan broj jedinki koji će se obraditi bit će 4000 (uvjet zaustavljanja), tj. evolucija će trajati 40 generacija

# Grada kromosoma

- broj gena (bitova) u kromosomu je proizvoljan
- odabiremo 44
- prvih 22 bita predstavljaju  $\mathbf{x}$ , a drugih 22 bita  $\mathbf{y}$
- $x$  i  $y$  su kodirani; interval  $x, y \in [-100, 100]$  se kodiranjem preslikava u  $x', y' \in [0, 2^{22}-1]$
- dekodiranjem opet treba interval  $[0, 2^{22}-1]$  preslikati u  $[-100, 100]$
- dekodiranje:  
 $x', y'$  pomnožiti s  $4,768372718899898 \cdot 10^{-5}$  i rezultatu oduzeti 100

# Inicijalizacija populacije

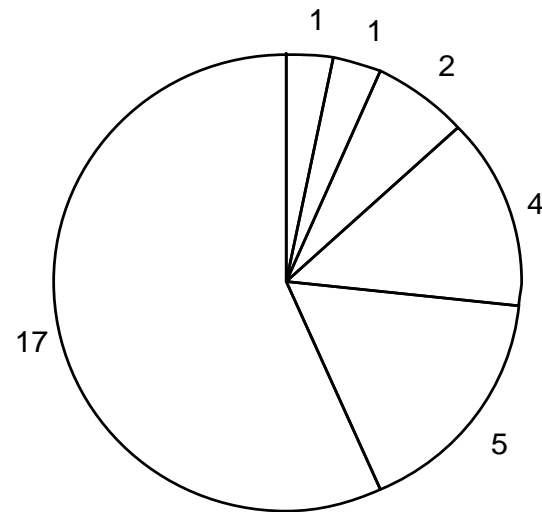
- Svaka jedinka u početnoj populaciji generira se kao niz od 44 slučajna bita
- Slučajna inicijalizacija je vrlo česta u GA



# Odabir jedinki za križanje

- Razmjerni ili proporcionalni odabir (ili selekcija): jedinku za križanje odabire se slučajnim odabirom, pri čemu je vjerojatnost odabira razmjerna dobroti jedinke
- Izvorni naziv "*roulette wheel parent selection*": jedinke su polja u kolu ruleta, a širine polja odgovaraju vjerojatnostima odabira pojedine jedinke.

Na slici je primjer za 6 jedinki: svako polje (odsječak) predstavlja jedinku, a površina odsječka je dobrota (vjerojatnost odabira) te jedinke.



# Odabir jedinki za križanje

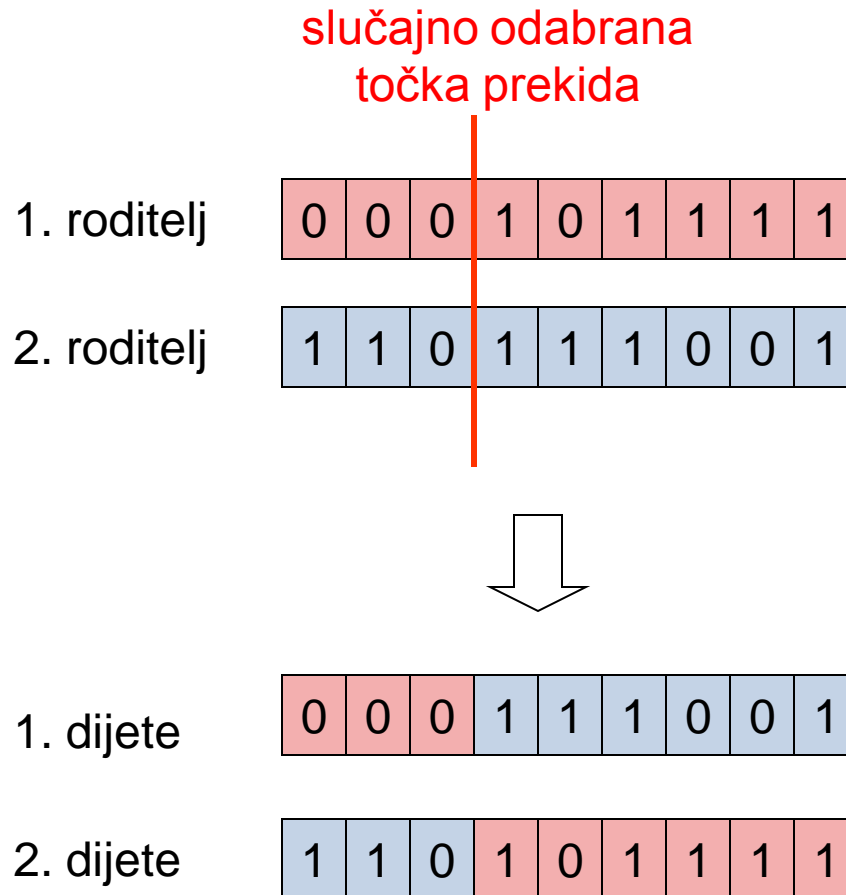
- Razmjerni odabir:  
 $p_i^{(k)}$  = vjerojatnost odabira  $i$ -te jedinke  
 $g_i^{(k)}$  = dobrota  $i$ -te jedinke u  $k$ -toj generaciji

$$p_i^{(k)} = \frac{g_i^{(k)}}{\sum_i g_i^{(k)}} = \frac{g_i^{(k)}}{G(k)}$$

- Mehanizam odabira oponaša prirodnu selekciju u smislu da se uspješnijim jedinkama daje veća šansa za reprodukciju

# Križanje

- Križanje s jednom prekidnom točkom (engl. *one-point crossover*):



# Križanje

- Križanje je ključan korak u GA, koji ga razlikuje od ostalih algoritama baziranim na slučajnom pretraživanju
- Križanje s jednom prekidnom točkom od dva roditelja proizvodi dva djeteta
- Treba uočiti da nova rješenja mogu biti znatno drugačija od roditeljskih
- Također, na mjestima gdje su bitovi roditelja jednaki, ovo križanje ne unosi nikakvu promjenu

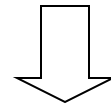
# Mutacija

- u ovom primjeru, mutacija se može provoditi na svakoj novoj jedinki
- mutacija djeluje na pojedine bitove i to nezavisno
- za svaki bit se s određenom (relativno malom) vjerojatnošću odluči da li će mutirati ili ne. Ako mutira, onda se slučajno odabire njegova nova vrijednost (0 ili 1). To znači da postoji 50% šanse da se mutirani bit zapravo neće promijeniti
- alternativna varijanta mutacije je da se mutirani bit uvijek mijenja

bitovi za koje je slučajno odabrano da mutiraju

prije mutacije

0	0	0	0	1	1	0	0	1	0	0	0	0	1	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



poslije mutacije

0	0	0	1	1	1	0	0	1	0	0	0	0	1	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

bit za koji je odabrana  
različita vrijednost od početne

bit za koji je odabrana  
vrijednost ista početnoj

# Mutacija

- Mutacija u ovom primjeru djeluje nakon što se proizvede novo dijete, dakle uvijek nakon križanja
- Mutacija može biti i zaseban "operator", a ne nužno drugi korak operacije križanja
- Mutacija unosi manje promjene u jedinke nego križanje

# Odnos križanja i mutacije

- Križanje je važno u **početnim fazama** rada GA, kad treba početnu slučajno generiranu populaciju (koja je vrlo heterogena) "izkonvergirati" čim brže do bolje populacije (koja je više homogena)
- Mutacija je važna u **završnim fazama** rada kad homogena populacija kvalitetnih jedinki križanjem daje iste ili vrlo slične kromosome ne unoseći više znatne promjene. Tada mutacija može unijeti promjene koje će pomaknuti rješenja bliže optimumu

# Zamjena starih jedinki novima

- U svakom koraku proizvodi se nova generacija jednake veličine kao stara generacija, a stara generacija se cijela briše
- Nova generacija dobiva se selekcijom iz stare generacije, nakon koje se na odabranim jedinkama primjenjuju križanje i mutacija
- Može se dogoditi, a u kasnijim fazama rada GA to je čest slučaj, da neki potomci u novoj generaciji budu jednaki ili vrlo slični roditeljima.



# Izračunavanje dobrote jedinki

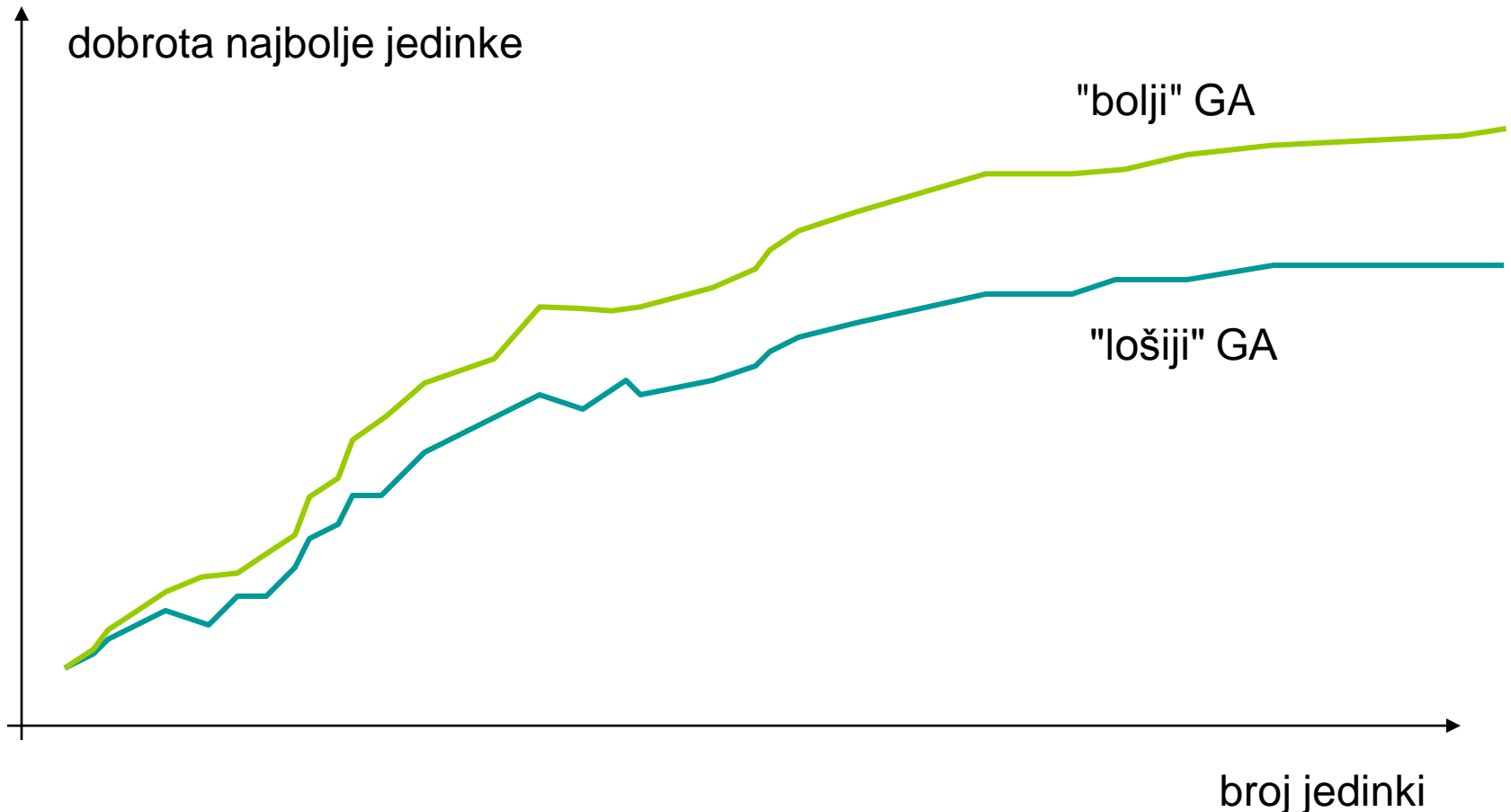
- U ovom primjeru izračunavanje dobrote (vrednovanje ili evaluacija) je najjednostavnije moguće: dobrota odgovara vrijednosti funkcije  $f_6$  koju maksimiziramo.

# Uvjet zaustavljanja

- Ovdje se koristi ukupan broj jedinki, a postoje i drugačiji uvjeti (npr. istijek vremena, pad poboljšanja dobrote rješenja ispod prihvatljive granice itd.)

# Tipičan rad algoritma

- Tipičan rad GA može se simbolički prikazati slikom:



# Tipičan rad algoritma

- Obično nema učinka puštati algoritam da radi dugo, jer nakon što dođe u "stabilno" stanje, daljnje promjene su malo vjerojatne
  - razlog je u tome što populacija postane homogena i za bitnije poboljšanje dobrote jedinki bilo bi potrebno da mutacija izmjeni relativno puno bitova u kromosomima (a to je malo vjerojatno)
- Korisnije je pokrenuti algoritam iznova, jer će početna generacija biti drugačija i možda će dati neko drugačije stabilno stanje

# Druge varijante GA i poboljšanja

- Svaki od djelova u prethodnom primjeru GA može biti i drugačiji
- Ovisno o konkretnom problemu, jedan način izvedbe može biti bolji ili lošiji

# Funkcija dobrote


- Nedostatci jednostavnog razmjernog odabira
  - negativne vrijednosti funkcije dobrote daju negativnu vjerojatnost, što nema smisla  
Rješava se translacijom.
  - slaba učinkovitost selekcije za male razlike u dobrotama velikih vrijednosti  
Ako je raspon premalen (npr. od 98 do 100), onda će sve jedinice imati približno jednaku šansu za razmnožavanje, a želimo da šanse izrazitije ovise o kvaliteti.  
Rješava se linearnom normalizacijom ili tehnikom “windowing”.

# Funkcija dobrote

- Povoljno je da funkcija dobrote daje vrijednosti u određenom rasponu (npr. od 0 do 100) tako da se prilikom razmjerne selekcije bolje jedinke biraju s većom vjerojatnošću, ali da se i one lošije ipak ne eliminiraju u potpunosti (jer dijelovi njihovih kromosoma mogu biti dobri)
- Razmjerni odabir gotovo uvijek se može poboljšati kombiniranjem raznih tehnika, a tri osnovne su:
  - translacija
  - linearna normalizacija
  - *windowing*

# Funkcija dobrote

- Linearna normalizacija se provodi ovako:
  - Jedinke (kromosomi) se sortiraju po početnoj funkciji dobrote u padajućem redoslijedu
  - Najboljoj jedinki pridijeli se dobrota određene vrijednosti (1. parametar postupka)
  - Svakoj sljedećoj jedinki se dobrota umanjuje za neku vrijednost (2. parametar postupka), ali ne ispod nekog minimuma
  - može se krenuti i od minimuma pa uvećavati dobrote za neki korak

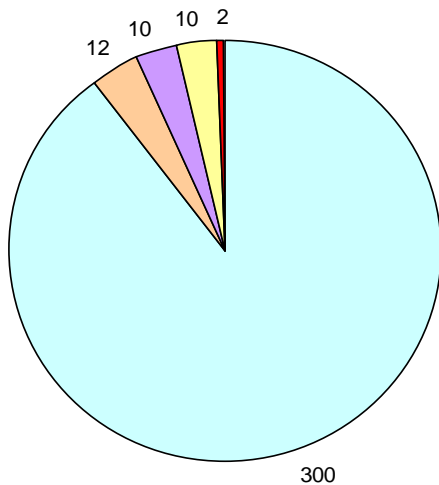


Početna dobrota:	300	12	10	10	2
Linearna dobrota (100,10):	100	90	80	70	60
Linearna dobrota (100,25):	100	75	50	25	1

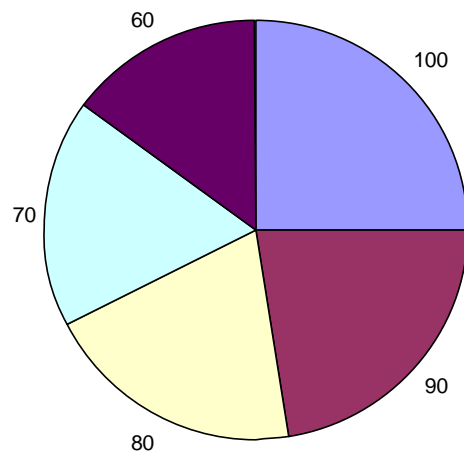


# Funkcija dobrote

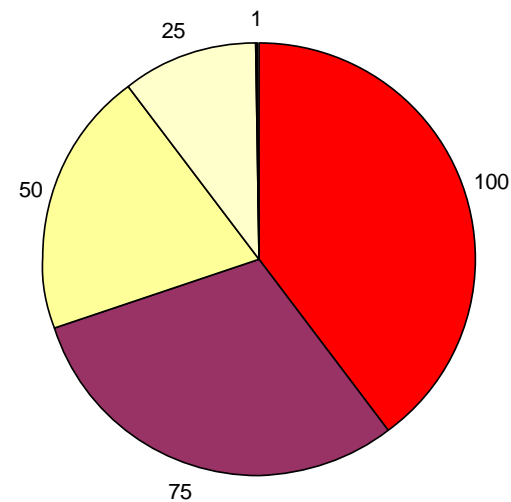
Početna dobrota:	300	12	10	10	2
Linearna dobrota (100,10):	100	90	80	70	60
Linearna dobrota (100,25):	100	75	50	25	1



Početna dobrota



Linearna dobrota (100,10)



Linearna dobrota (100,25)

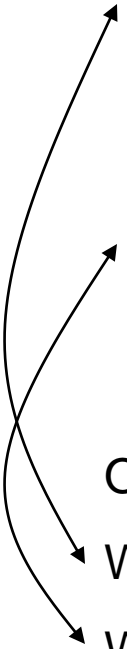
# Funkcija dobrote

- *windowing* se provodi ovako:
  - pronaći najmanju dobrotu u populaciji
  - ostalim jedinkama kao nova dobrotu se dodijeljuje razlika od njihove dobrote do najmanje dobrote:

$$\text{dobrota}_{\text{nova}}(i) = \text{dobrota}_{\text{start}}(i) - \min\{\text{dobrota}(i)\}$$

- alternativno se umjesto najmanje dobrote može postaviti neka granična najmanja vrijednost

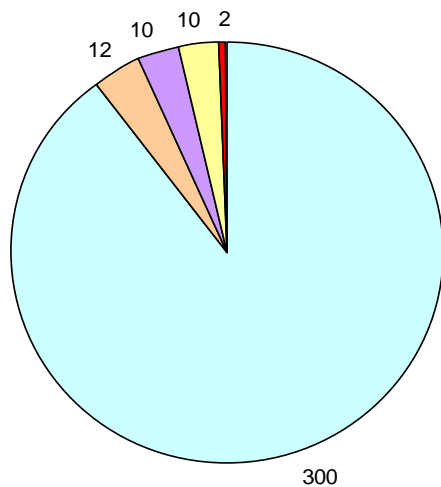
$$\text{dobrota}_{\text{nova}}(i) = \text{dobrota}_{\text{start}}(i) - \text{granica}$$



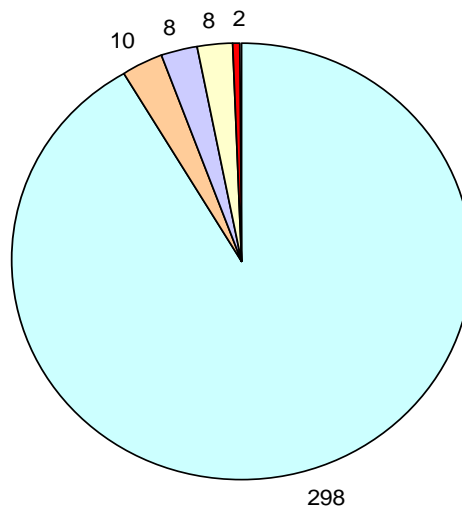
Originalna dobrotu:	300	12	10	10	2
Windowing:	298	10	8	8	2
Windowing s granicom 5:	295	7	5	5	5

# Funkcija dobrote

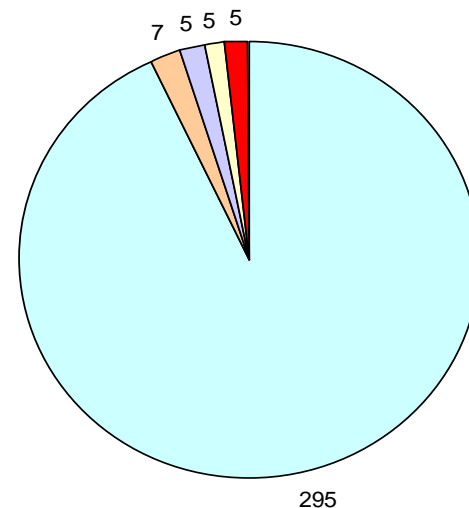
Početna dobrota:	300	12	10	10	2
Windowing:	298	10	8	8	2
Windowing s granicom 5:	295	7	5	5	5



Početna dobrota



Windowing



Windowing s granicom 5

# Zamjena starih jedinki novima

- Moguće je da neka vrlo dobra jedinka bude izgubljena u sljedećoj generaciji
  - npr. uslijed križanja i mutacija dođe do njene promjene u lošiju
- Da bi se to izbjeglo može se koristiti elitizam (*elitism*) koji čuva najbolju jedinku
- Može doći doći do dominacije superjedinke, ali u prosjeku poboljšava rad algoritma

# Zamjena starih jedinki novima

- Kod generacijske zamjene se cijela generacija briše i nadomješta novom
- Čak i sa elitizmom, može se dogoditi da mnoge dobre jedinke zbog slučajnog odabira ne dobiju šansu za križanje ili da križanje i mutacija unište dobre jedinke
- Moguće rješenje problema je korištenje ravnotežne reprodukcije (*steady-state reproduction*) kod koje se samo manji broj populacije zamijeni novim jedinkama
- Parametar ravnotežne reprodukcije određuje broj novih jedinki, a često se stvara samo jedna ili dvije jedinke

# Zamjena starih jedinki novima

- Prilikom stvaranja novih jedinki često se događa da se pojavljuju duplikati kromosoma, što nepovoljno utječe na rad GA
- Zato se često kod ravnotežne reprodukcije zabranjuje stvaranje duplikata (kompliciraniji algoritam, ali u konačnici bolje radi)

# Vrijednosti parametara

- Vjerojatnosti križanja i mutacije mogu znatno utjecati na rad GA
- Prilikom generacijske zamjene, najbolje jedinice se čuvaju nešto manjom vjerojatnošću križanja i mutacije
  - kad se primjenjuju elitizam i ravnotežna reprodukcija te vjerojatnosti mogu biti veće
- Tijekom izvođenja algoritma vjerojatnosti se mogu mijenjati

# Vrijednosti parametara

- Npr. na početku rada se može veća vjerojatnost dati križanju, a kasnije se može povećati vjerojatnost mutacije
- Ove vjerojatnosti se mijenjaju tako da npr. linearno rastu ili padaju tijekom rada GA



# Genetski operatori

- Osim kombiniranog operatora križanja i mutacije, oni se mogu razdvojiti u dva nezavisna operatora
- Također može postojati više vrsta križanja i mutacija
- Tada se operator može odabirati na slučajan način (npr. svaki ima svoju težinu i onda se odabire s vjerojatnošću razmjernoj težini - slično razmjernom odabiru)

# Genetski operatori

- Osim križanja s jednom prekidnom točkom može se napraviti i križanje s dvije ili više prekidnih točaka
- Bez obzira na broj točaka, ovakva križanja mogu proizvesti samo ograničene kombinacije jedinki, tj. križaju genetski materijal na ograničeni način

# Genetski operatori

- Uniformno križanje - odabrani bitovi jednog roditelja zamjenjuju se bitovima drugog roditelja (obrnuto za drugo dijete)
  - efektivno kao da ima  $n$  točaka prekida, pri čemu kromosom ima  $n+1$  bit

