

# Genetski Algoritmi

Predavač: Aleksandar Kovačević

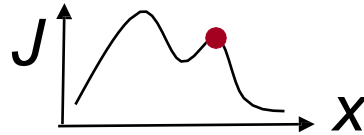
Slajdovi preuzeti sa:

<https://ocw.mit.edu/courses/institute-for-data-systems-and-society/ids-338j-multidisciplinary-system-design-optimization-spring-2010/>

# Tehnike za Heurističke Pretrage (*Heuristic Search*)

Glavne motivacije za ove tehnike:

(1) Rešavanje problema zaglavljivanja u lokalnim optimumima.



(2) Optimizacija kod problema kod kojih promenljive nisu numeričke već npr. kategorijalne ili kod problema kod kojih funkcija koja se optimizuje ne zadovoljava neke karakteristike (npr. nije diferencijabilna)

$$x_i \notin \mathbb{R}$$

$$x_i = \{1, 2, 3, 4, 5\}, \quad x_i = \{, "A", "B", "C" \} \quad x_i = \{\text{true}, \text{false}\}$$



Ove tehnike ne garantuju da će globalni optimum biti pronađen.

# Tehnike za Heurističke Pretrage (*Heuristic Search*)

- Genetski Algoritmi (Holland – 1975)
  - Inspirisani genetikom i prirodnom selekcijom
- Simulirano Kaljenje (Kirkpatrick – 1983)
  - Inspirisano statistikom
- Optimizacija Rojem (Eberhart Kennedy - 1995)
  - Inspirisano ponašanje rojeva insekata ili jata ptica u potrazi za hranom.



Sve ove tehnike koriste kombinaciju slučajnosti i heuristike da vode pretragu do globalnog optimuma.

# Današnja tema: Genetski Algoritmi

- Genetika i Prirodna Selekcija
- Jednostavan Genetski Algoritam (SGA)
- Kodiranje - Dekodiranje (Reprezentacija)
- Fitness Funkcija - Selekcija
- Ukršanje – Dodavanje u populaciju - Završetak

# Pretpostavke Genetskih Algoritama

- Prirodna selekcija je uspešan način za optimizaciju individua u populaciji
- Ako možemo nekako da imitiramo prirodnu selekciju dobićemo moćan alat za optimizaciju
- Jedan način da to postignemo – svako potencijalno rešenje  $x$  je jedinka koja se bori za opstanak u populaciji
- Opstaju samo najbolje jedinke – Fitness (kvalitet)  
jedinke kreiramo tako da bude funkcija koju pokušavamo da optimizujemo

# Prirodna Selekcija

Čarls Darwin (1809-1882)

Kontraverzna i jako uticajna knjiga (1859)

*"On the origin of species by means of natural selection, or the preservation of favored races in the struggle for life"*

Šta možemo zaključiti iz Darvinovog rada:

- Vrste se stalno razvijaju
- Varijacije između vrsta su ogromne
- Samo mali/srednji procenat potomaka uspe da prežvi dovoljno dugo da postane odrasla jedinka

⇒ Evolucija = prirodna selekcija varijacija koje se nasleđuju

# Nasleđivanje Karakteristika

Gregor Mendel (1822-1884)

Istraživao je nasleđivanje karakteristika (“osobina”)

Vršio je veliki broj eksperimenta sa biljkama graška

Ispitivao je hibride i različite vrste graška

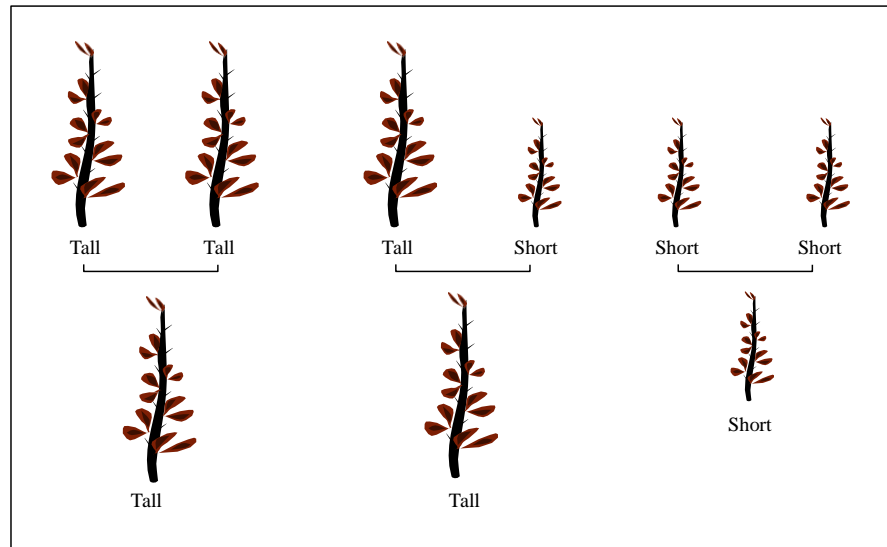
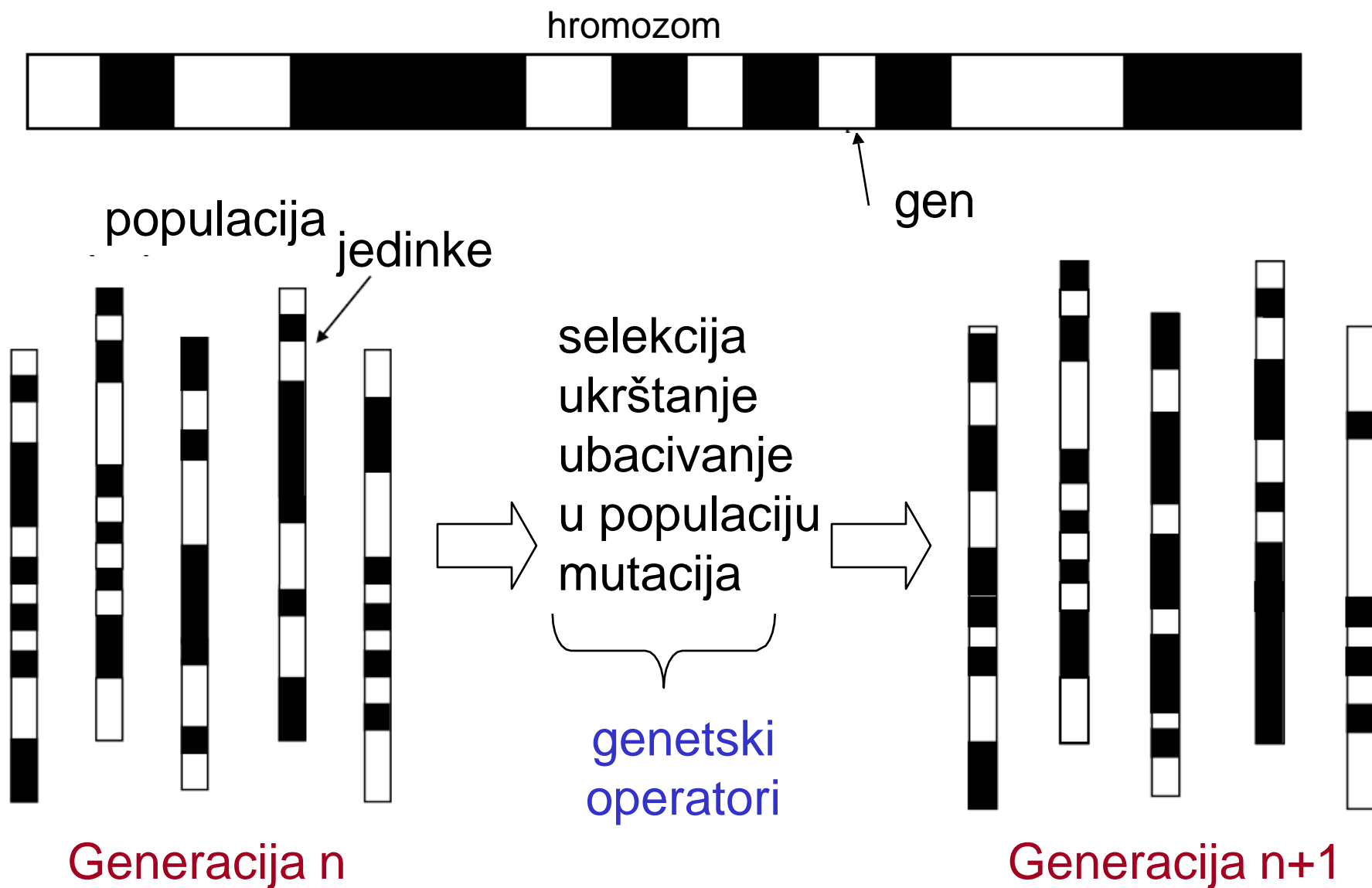


Image by MIT OpenCourseWare.



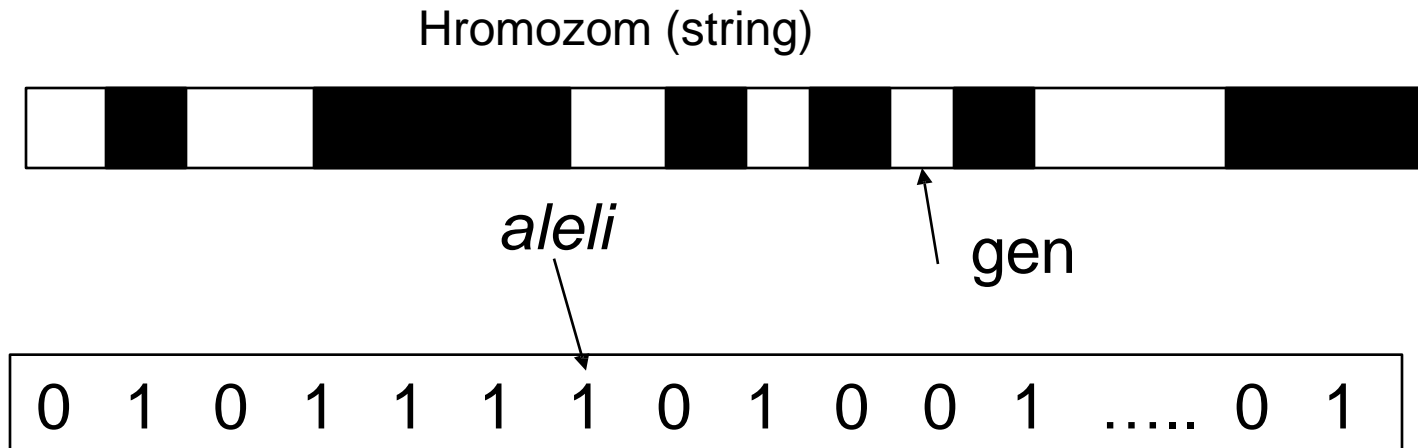
Gen koji imaju visoke biljke je dominantan  
Gen koji imaju niske biljke je recesivan

# GA Terminologija





# Hromozomi



Svaki hromozom je jedno potencijalno rešenje, često reprezentovan kao niz 0 i 1. Svaki bit je jedan gen. Ovo je binarna reprezentacija.

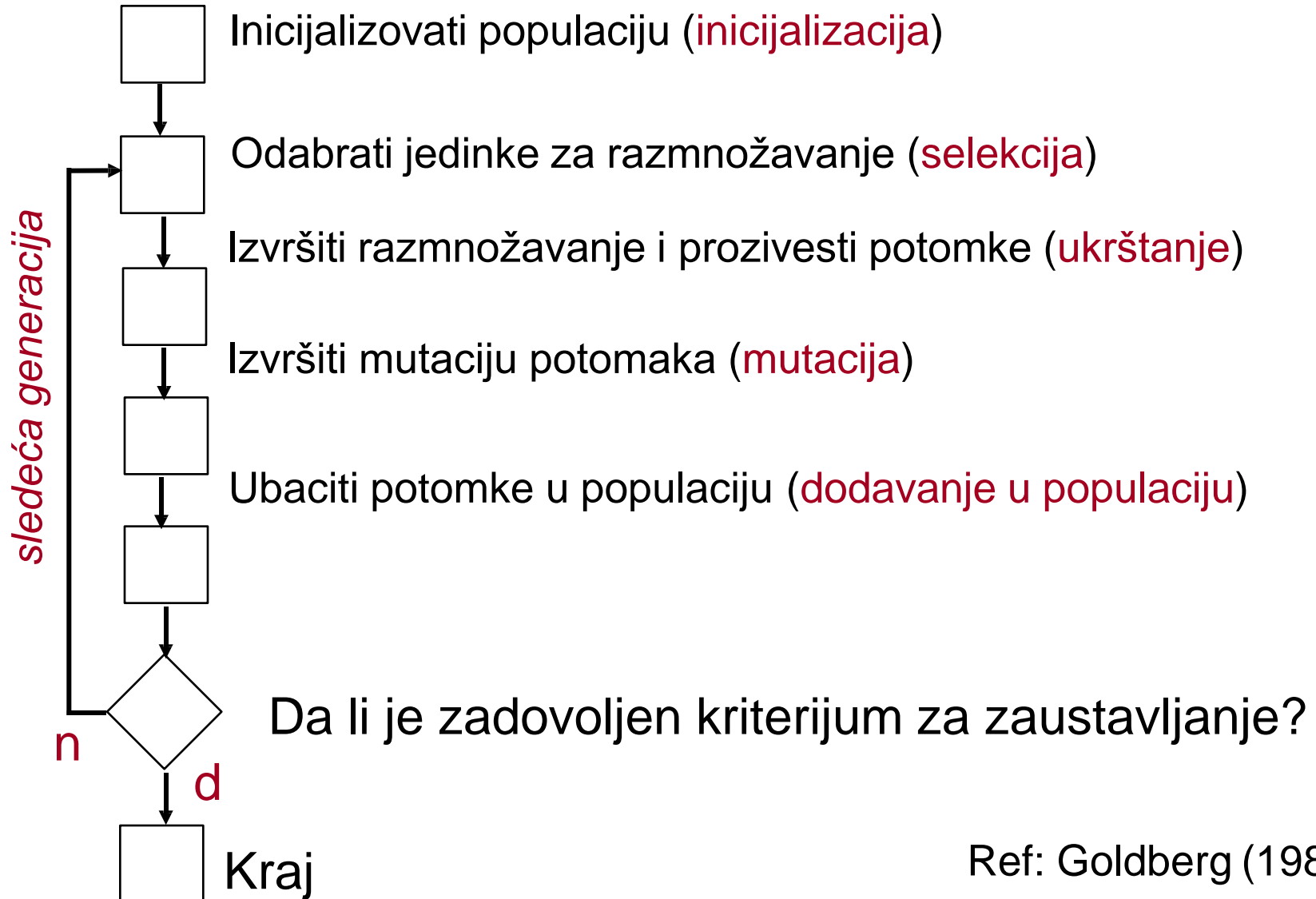
Vrednosti konkretnog gena su aleli.

Hromozom sam po sebi nema mnogo smisla –

Potreban je način da ga dekodiramo u konkretnu vrednost iz skupa rešenja (npr. realan broj).

Ovo je koverzija genotip -> fenotip

# Jednostavni Genetski Algoritam - SGA



Ref: Goldberg (1989)

# Realizacija GA na računaru

- (1) odabrati reprezentaciju (kodiranje-dekodiranje)
- (2) definisati “fitness” funkciju  $F$ 
  - uključiti sva ograničenja i ciljeve vezane za problem koji se rešava
- (3) definisati genetske operatore
  - inicijalizacija, selekcija, ukrštanje, mutacija, dodavanje u populaciju
- (4) pokrenuti algoritam
  - pratiti prosečan fitness
  - identifikovati najbolju jedinku
- (5) štelovati algoritam
  - podesiti/promeniti selekciju, strategiju dodavanja u populaciju, stopu mutacije,...

# Kodiranje - Dekodiranje

**genotip**

**fenotip**

*domen kodiranih vrednosti    domen dekodiranih vrednosti*

**Biologija**

UGCAACCGU  
(“DNK” gradivni  
blokovi)

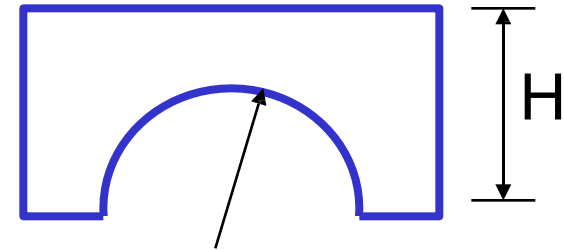
*ekspresija*  
→  
*sekvenciranje*  
←

“plave oči”

**GA**

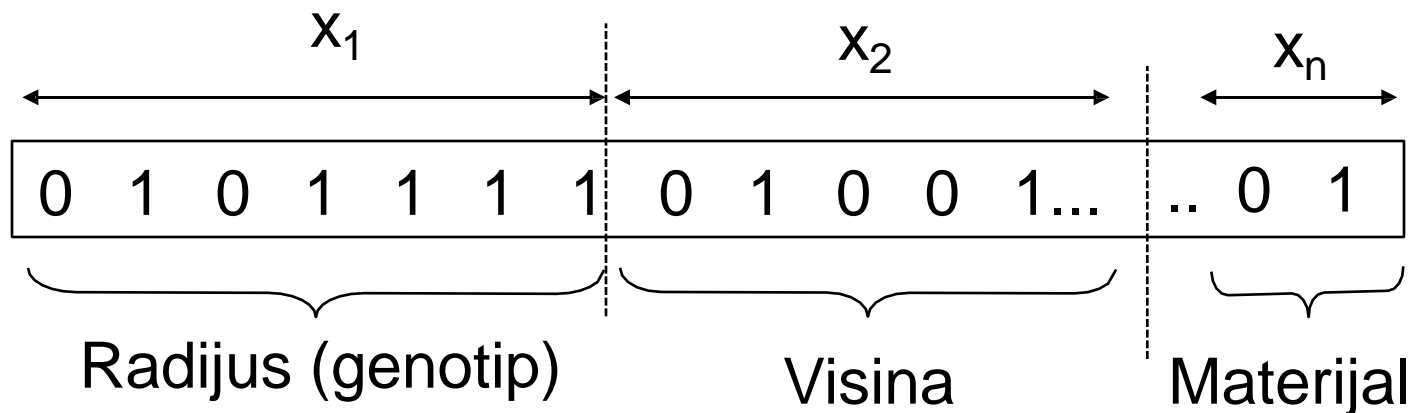
10010011110  
(hromozom)

*kodiranje*  
→  
*dekodiranje*  
←



Radius  $R=2.57$  [m]

# Dekodiranje



Npr. binarno kodiranje celih brojeva:

10100011

$$(1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0)$$

$$128 + 0 + 32 + 0 + 0 + 0 + 2 + 1 = 163$$

# Binarno kodiranje detaljnije

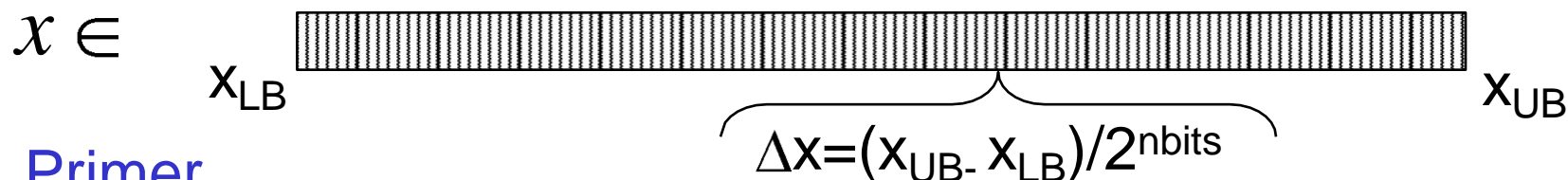
Broj bitiova koji se koristi je veoma važan.

Broj potrebnih bitova za dato  $\Delta x$ :

Preciznost (gubitak informacija) zavisi od:

- gornje i donje granice  $x$ :  $x_{LB}$ ,  $x_{UB}$
- broja bitova koji se koriste

$$nbits = \left\lceil \frac{\ln \left( \frac{x_{UB} - x_{LB}}{\Delta x} \right)}{\ln 2} \right\rceil$$



## Primer

```
[G]=encode(137.56, 50, 150, 8)
```

```
G = 1      1      0      1      1      1      1      1
```

```
[X]=decode(G, 50, 150, 8);
```

```
X = 137.4510
```

$$\Delta x = (150 - 50) / 2^8 = 0.39$$

Gubitak preciznosti !!!

```
encode(vrednost,  $x_{UB}$ ,  $x_{LB}$ , broj_bitova)
```

# Druge vrste kodiranja

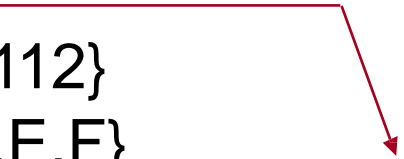
Binarno Kodiranje nije jedina opcija

Možemo koristiti bilo koji ALFABET za kodiranje

Najčešće korišćen je binarni alfabet  $\{0,1\}$

Možemo koristiti:

- ternarni:  $\{0,1,2\}$   $\{A,B,C\}$
- kvaternarni:  $\{0,1,2,3\}$   $\{T,G,C,A\} \Rightarrow$  biologija
- cele brojeve:  $\{1,2,\dots,13,\dots\}$
- realne brojeve:  $\{3.456\ 7.889\ 9.112\}$
- Hexadecimalne  $\{1,2,\dots,A,B,C,D,E,F\}$



Korisiti se kod  
Problema  
Trgovačkog Putnika

# Primer: Reprezentacija Problema Rasporeda Vatrogasnih Stanica

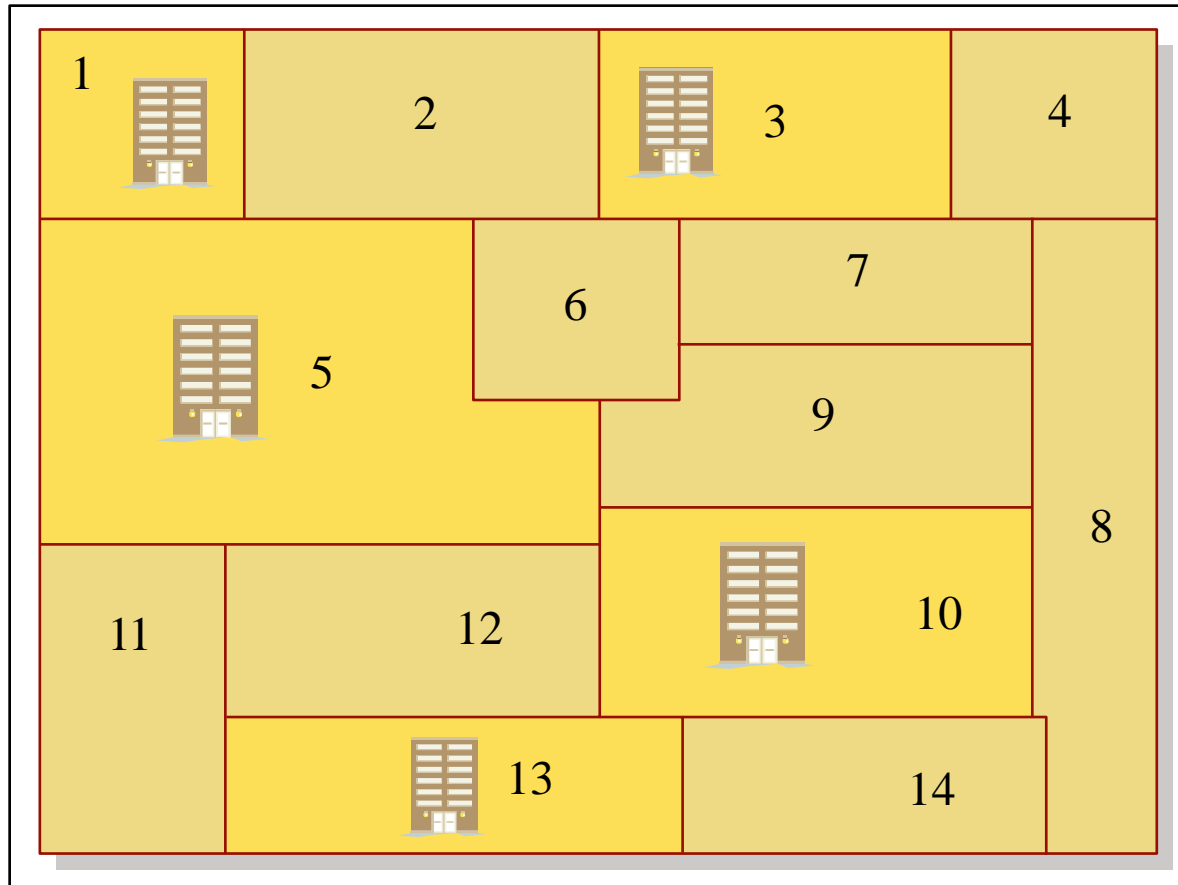


Image by MIT OpenCourseWare.

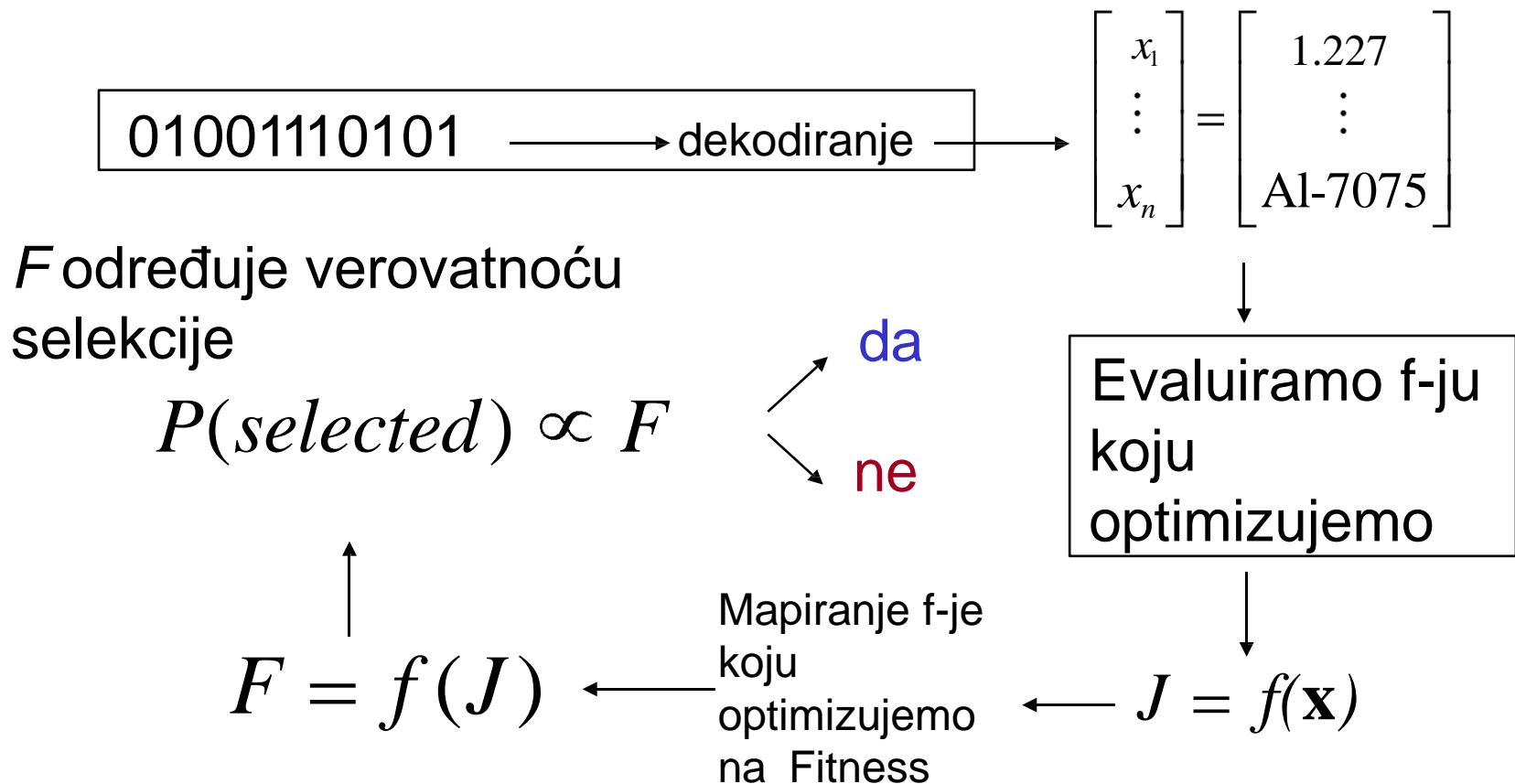
1 0 1 0 1 0 0 0 0 1 0 0 1 0

*“1” predstavlja vatrogasnu stanicu*



# Fitness i Verovatnoća Selekcije

Tipično, selekcija je najvažniji i vremenski najzahtevniji deo GA.



$\propto$  - znači „proporcionalno sa“

# Fitness Funkcija

- Definisanje fitness funkcije je vrlo značajno i nje uvek lako
- GA nemaju eksplicitna “ograđivanja” kao neke druge optimizacione metode
- Ograničenja se mogu zadati na razne načine:
  - pomoću fitness funkcije – kazna za jedinke koje krše ograničenja
  - pomoću operatora selekcije (“ne dozvoliti jedinkama koje krše ograničenja da budu odabrane”)
  - pomoću reprezentacije/kodiranja npr. kod TSP problema dozvoliti reprezentacije koje predstavljaju validan obilazak (nema ponavljanja i sva mesta su tu)

# Maksimizacija vs. Minimizacija

Postoji mnogo načina da se maksimizacija ciljne funkcija (obj) konvertuje u minimizaciju:

- $N \cdot \text{obj}$
- $1/\text{obj}$
- $-\text{obj}$

# Selekcija Rangiranjem

- Cilj: **odabrati roditelje za ukrštanje**
- Trebalo bi dati prednost jedinkama koje imaju više fitness-a
- Takođe, treba očuvati raznovrsnost u populaciji

## (1) Selekcija Rangiranjem:

Rangirati sve jedinke po fitness-u.

Primer: Neka je  $D = \sum_{j \in P} 1/j$

k-to rangirana jedinka ima sledeću verovatnoću da bude odabrana:  $P_k = \frac{1}{k} D^{-1}$



**Bolji rang znači veću šansu za odabir**

**npr.  $1 \propto 1$ ,  $2 \propto 1/2$ ,  $3 \propto 1/3$  ...**

D je faktor normalizacije, da bi se verovatnoće sabirale na 1. To je zbir  $1/\text{rang}$  za sve jedinke.

# Selekcija po Fitness-u

(2) Verovatnoća da će jedinka biti izabrana je proporcionalna fitness-u

Primer: Neka je  $\bar{F} = \sum_{j \in P} Fitness_j$

k-ta jedinka najbolja po fitness-u ima sledeću verovatnoću da bude odabrana:

$$P_k = Fitness(k) \cdot \bar{F}^{-1}$$

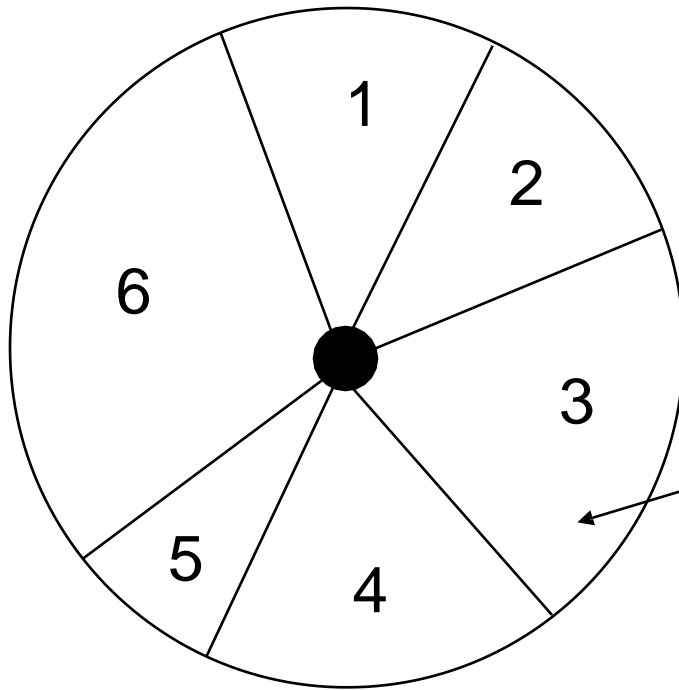


Verovatnoća da će jedinka biti izabrana je direktno proporcionalna fitness-u.

F je faktor normalizacije, da bi se verovatnoće sabirale na 1.

# Selekcija pomoću ruleta

## Selekcija pomoću ruleta



Još jedan metod zasnovan na verovatnoći.

Jedinke sa većim fitnessom (ili nekom proporcionalnom merom) zauzimaju veću površinu na krugu za rulet.

Biramo jedinku na kojoj se zaustavi kuglica (slučajno generisan broj).

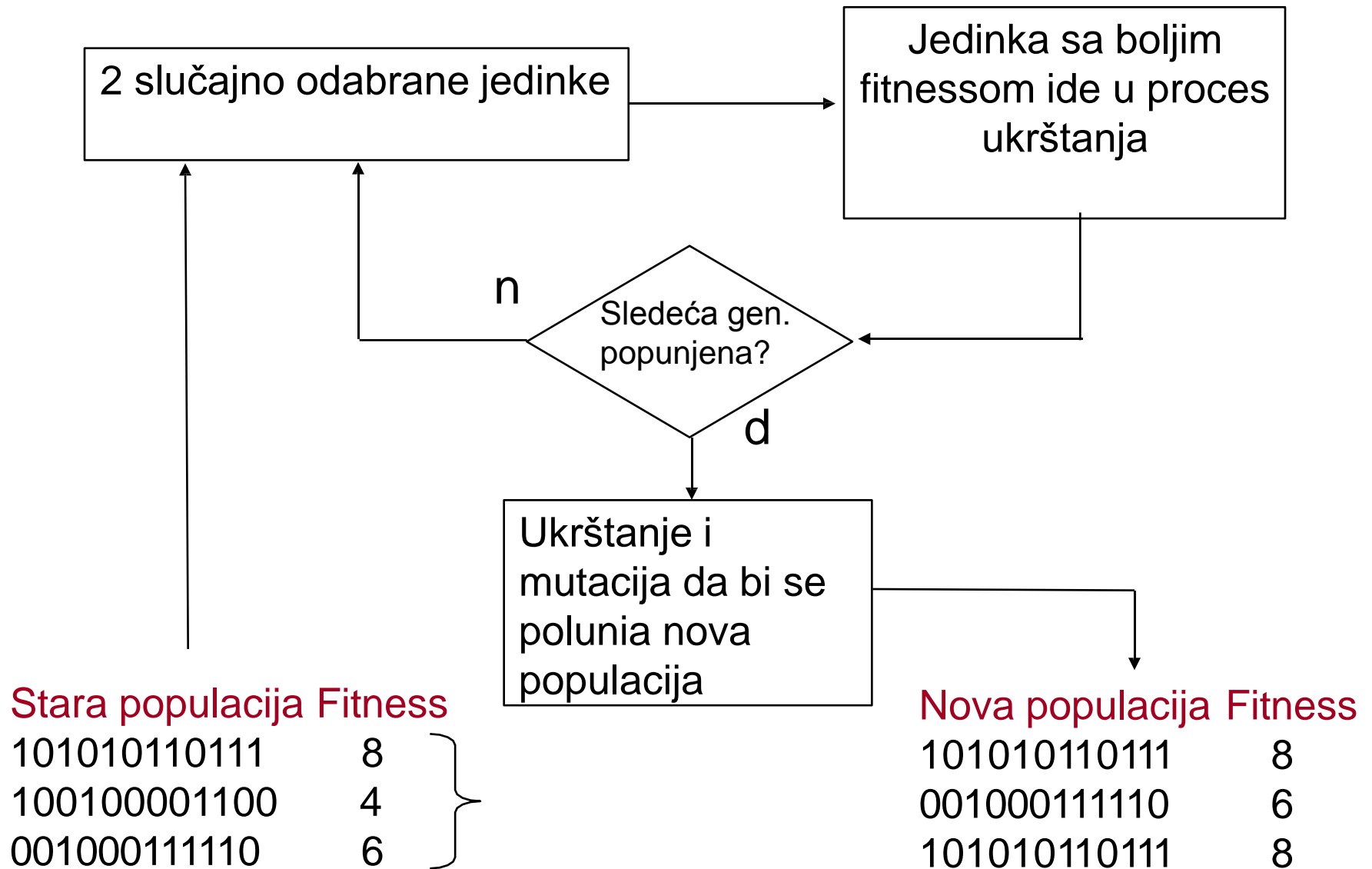
treća jedinka u populaciji mapirana na interval  $[0, Zbir]$

$Zbir$  = zbir vrednosti odabrane mere za sve jedinke

Selekcija: generišemo slučajan broj u  $[0, Zbir]$

Ponavljamo proces dok ne odaberemo zadati broj jedinki

# Selekcija pomoću turnira



# Ukrštanje



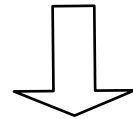
0	1	0	1	1	1	1	0	1	1	1	1	.....	1	1
---	---	---	---	---	---	---	---	---	---	---	---	-------	---	---

R1



1	1	1	0	0	1	0	0	0	1	0	1	.....	0	0
---	---	---	---	---	---	---	---	---	---	---	---	-------	---	---

R2



ukrštanje

P1

?
---

P2

?
---

Na koji način upotrebiti reprezentacije roditelja R1 i R2 da bi dobili potomke P1 i P2 (iste dužine, takođe samo 0 i 1)?



# Operator Ukrštanja (I)

Koristimo 2 roditelja da kreiramo 1 ili više potomaka

Uobičajeno: ukrštanje oko jedne tačke

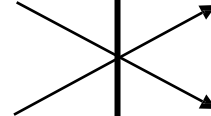
R1	0	1	1		0	1
R2	1	0	0		1	1

Roditelji

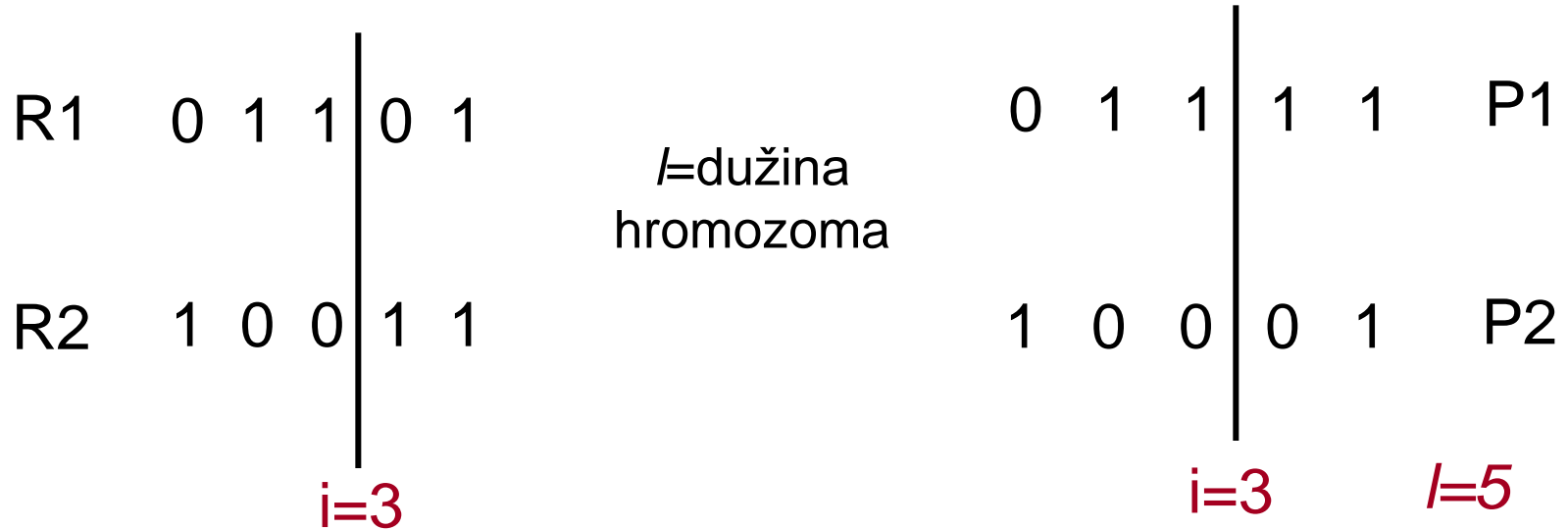
0	1	1		1	1	P1
1	0	0		0	1	P2

Potomci

tačka  
ukrštanja



# Operator Ukrštanja (II)

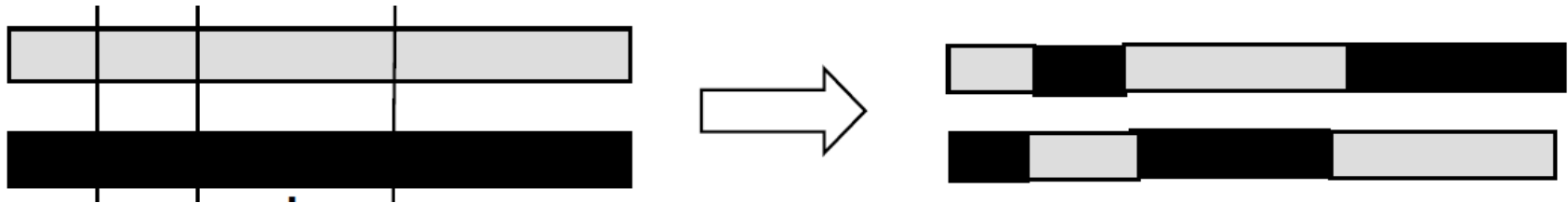


Bit na poziciji “ $i$ ” bira se namerno (ili slučajno) i tako delimo roditelje na 2 dela.

Potomak P1 dobija prvu polovinu R1, a drugu polovinu R2  
Potomak P2 dobija prvu polovinu R2, a drugu polovinu R1

# Operator Ukrštanja (III)

- Možemo koristiti jednu, dve ili više tačka oko kojih radimo razmenu gena.



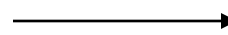
- Još jedna od mogućnosti je uniformno ukrštanje.
- Prolazimo kroz pozicije bitova, bacamo kockicu i sa nekom verovatnoćom vršimo razmenu bitova između roditelja.

# Roditelji

# Potomci

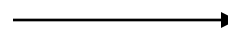
Jedna Tačka

11000101 01011000 01101010



11000101 01011001 01111000

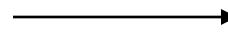
00100100 10111001 01111000



00100100 10111000 01101010

Dve Tačke

11000101 01011000 01101010



11000101 01111001 01101010

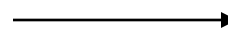
00100100 10111001 01111000



00100100 10011000 01111000

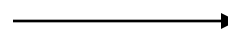
Uniformno

11000101 01011000 01101010



01000101 01111000 01111010

00100100 10111001 01111000



10100100 10011001 01101000

# Mutacija

- Svrha mutacije je da na unese malu promenu u populaciju.
- Takve promene sprečavaju zaglavljivanje u lokalnim optimumima.
- Kod binarne reprezentacije tipično se neki od bitova invertuju. (kod relanih br. vršimo +/- neka mala vrednost)

Pre: (1 0 1 1 0 1 1 0)

Nakon: (0 1 1 0 0 1 1 0)

Pre: (1.38 -69.4 326.44 0.1)

Nakon: (1.38 -67.5 326.44 0.1)

# Strategije dodavanja u novu populaciju

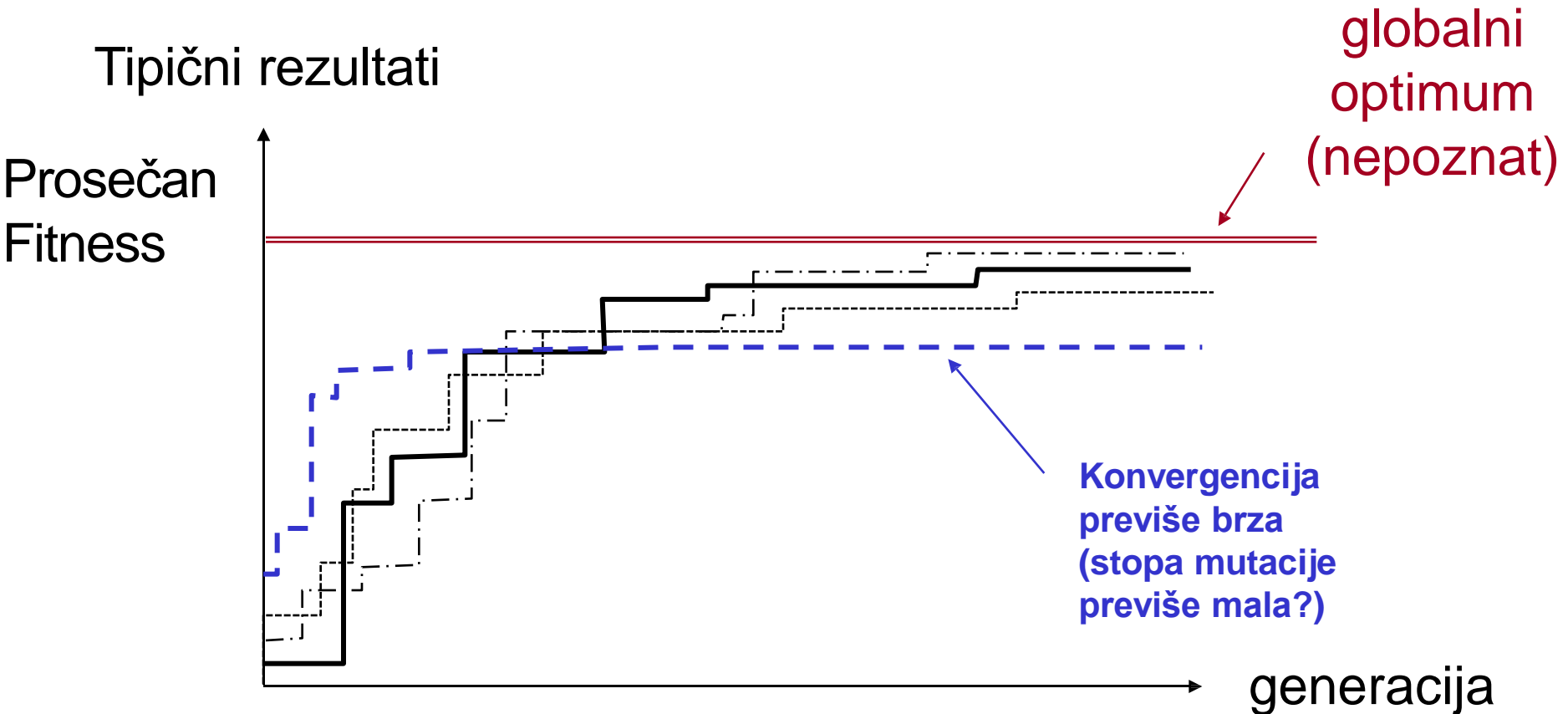
- Jedna opcija je da novu populaciju celu popunimo novim jedinkama.
  - Od 2 roditelja kreiramo 2 potomka i dodajemo ih u novu populaciju
  - Od 2 roditelja kreiramo 1 potomka i u staroj populaciji jednu jedinku (npr. najgoru) zamenimo tom
- “Elitizam”
  - Najboljih  $N$  jedinki trenutne populacije direktno prebacujemo u novu
  - Ovo je ekvivalent čuvanju trenutno najboljih rešenja
  - Zamislite da je ovo moguće kod ljudi....

# Inicijalizacija

## Kako kreirati prvu populaciju?

- Prva opcija, na slučajan način, ovo je tipično
- Naravno, ovaj proces ne bi trebalo da bude potpuno slučajan
- Moramo da pazimo da nema previše ponavljanja
- Moramo da pazimo da eventualna ograničenja budu zadovoljena.
- (Npr. kod TSP da svaka jedinka bude validan oblizak itd.)

# Konvergencija GA



Očekujemo da prosečan fitness populacije raste kroz generacije tako što selekcijom birmo dobre jedinke, a ukrštanjem prenosimo njihov kvalitet u sledeću generaciju. Mutacija je tu da unese raznovrsnost i izvuče nas iz lokalnih optimuma.



# Kriterijum zaustavljanja GA

Neke od mogućnosti:

- Zaustaviti se posle X generacija.
- Ako je srednje odstupanje u fitnessu između jedinki u celoj populaciji malo. Ovo faktički znači da su sve jedinke jako slične. Daljim generacijama ne dobijamo ništa.
- Stagnacija – najbolje rešenje se ne menja (ili vrlo malo menja) određeni broj generacija.

# GA u odnosu na druge metode

Razlike u odnosu na tradicionalne metode optimizacije i pretrage:

- Pretražuju celu populaciju rešenja paralelno, a ne samo jedno
- Koriste probabilističke metode za kreiranje novih potencijalnih rešenja, a ne determinističke
- Rade sa reprezentacijom potencijalnih rešenja, a ne sa samim rešenjem
- Ne zahtevaju da ciljna funkcija bude diferencijabilna, niti da zadovoljava bilo koje druge uslove. Bitno je samo da možemo da izračunamo fitness.