

Predavanje

## Evolucijsko računarstvo

u okviru predmeta

### Neizrazito, evolucijsko i neuro-računarstvo

Marin Golub  
2013/2014

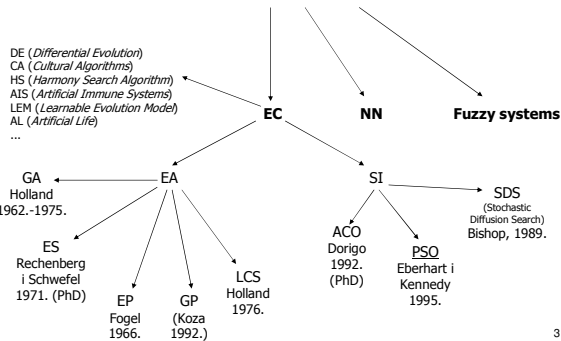
1

## Literatura

- Genetski algoritam, skripta prvi i drugi dio, M. Golub, 2010., dostupno na Internet adresi:  
<http://www.zemris.fer.hr/~golub/ga/ga.html>
- **Predznanje stečeno na drugim predmetima (UI, APR, NASP, ...)**
  - ♦ Genetski algoritam, skripta prvi dio, M. Golub, 2010., treće poglavlje, dostupno na Internet adresi:  
[http://www.zemris.fer.hr/~golub/ga/ga\\_skripta1.pdf](http://www.zemris.fer.hr/~golub/ga/ga_skripta1.pdf)

2

## Meko računarstvo (Soft computing) Računska inteligencija (Computational intelligence - CI)



3

## Zajedničke osobine evolucijskih algoritama

- **Jedinka** predstavlja točku u prostoru rješenja, odnosno predstavlja moguće rješenje problema.
- Algoritmi su zasnovani na **populaciji** rješenja.
- **Jedinke** su međusobno usporedive prema **dobroti**.
- **Populacija** jedinki se s vremenom mijenja, evoluira jer se provodi postupak **selekcije** jedinki.
- **Svojstva jedinki** se prenose s roditelja na djecu.
- **Prostor rješenja** se pretražuje slučajnim procesom mutacije.

4

## Stečeno predznanje na drugim predmetima

- Analiza i projektiranje računalom
  - ♦ tGA i sGA
  - ♦ uniformno križanje, križanje s jednom ili više točaka prekida
  - ♦ jednostavna mutacija
- Umjetna inteligencija
  - ♦ GA, ACO, imunološki algoritmi i rojevi čestica
  - ♦ sGA

5

## Ponavljjanje

```
Genetski_algoritam
{
    t = 0;
    generiraj početnu populaciju potencijalnih rješenja P(0);
    sve dok nije zadovoljen uvjet završetka evolucijskog procesa
    {
        t = t + 1;
        selektiraj P'(t) iz P(t-1);
        križaj jedinke iz P'(t) i djecu spremi u P(t);
        mutiraj jedinke iz P(t);
    }
    ispiši rješenje;
}
```

6

## Priprema za optimiranje genetskim algoritmom

- Prikaz rješenja (jedinka, kromosom)
  - ♦ binarni, brojevi s pomičnom točkom
  - ♦ permutirani niz cijelih brojeva
  - ♦ niz, matrica, program
- Vrednovanje jedinki (evaluacija)
  - ♦ svakom je kromosomu pridružena vrijednost dobrote
  - ♦ iz funkcije cilja: funkcija dobrote ili funkcija kazne (funkcija cilja != funkcija dobrote)
  - ♦ često se najviše vremena troši upravo na evaluaciju

7

## Selekcija

- postupak odabira jedinki za reprodukciju i/ili iduću generaciju
- neki postupci selekcije zahtijevaju pripremu:
  - ♦ vrijednosti dobivene evaluacijom se trebaju pomaknuti (translacija) ili
  - ♦ linearno normalizirati (sortiranje)
- obično se podrazumijeva da je selekcija odabir jedinki za iduću populaciju, a za reprodukciju se jedinke odabiru nasumično

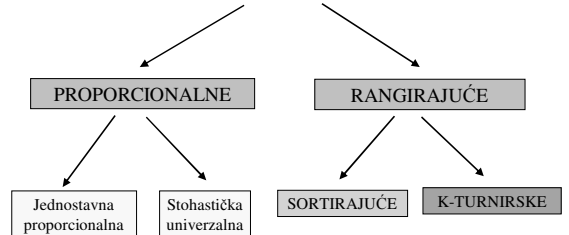
8

## Vrste selekcija

- generacijske
  - ♦ selektiraju se jedinke koje će sudjelovati u reprodukciji
  - ♦ potrebno je istovremeno voditi brigu o dvije populacije
  - ♦ problem duplikata
  - ♦ vjerojatnost odabira ovisi o *funkciji dobrote*
- eliminacijske
  - ♦ samo jedna populacija iz koje se izbacuju (eliminiraju) odabrane jedinke
  - ♦ novi parametar *M* (*mortalitet*)
  - ♦ vjerojatnost odabira ovisi o *funkciji kazne*

9

## VRSTE SELEKCIJA



10

## Selekcijski pritisak

- odnos vjerojatnosti preživljavanja dobrih i loših jedinki
- **veliki selekcijski pritisak**
  - ♦ s velikom vjerojatnošću prenosi bolje jedinke u iduću generaciju, odnosno s velikom vjerojatnošću eliminira jedinke ispodprosječne dobrote
  - ♦ brže do lošijih rješenja (lokalnog optimuma)
- **premali selekcijski pritisak**
  - ♦ prespora konvergencija
- na selekcijski pritisak ne utječe samo selekcija sa svojim parametrima, već i ostali operatori pa se on određuje posredno

11

## Određivanje selekcijskog pritiska

- **trajanja preuzimanja**
  - ♦ samo operator selekcije bez križanja i mutacije
  - ♦ prosječan broj generacija nakon kojih se populacija sastoji od *N* duplikata najbolje jedinke
  - ♦ ↓ trajanje preuzimanja ⇒ selekcijski pritisak ↑
- **selekcijska razlika**
  - ♦ razlika prosječne vrijednosti dobrote preživjelih jedinki i prosječne vrijednosti dobrote svih jedinki
$$s(t) = d_p(t) - d(t)$$
  - ♦ ↑ selekcijska razlika ⇒ selekcijski pritisak ↑

12

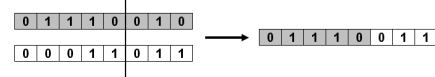
## Reprodukcija = križanje + *mutacija*

- genetski operatori križanja i mutacije ovise o prikazu rješenja jer treba izbjegavati nemoguća rješenja

13

## Binarni prikaz, križanje

Križanje s jednom točkom prekida



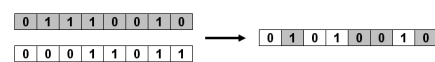
*One-point crossover*

Križanje s više točaka prekida



*N-point crossover*

Uniformno križanje

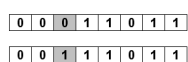


*Uniform crossover*

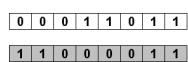
14

## Binarni prikaz, mutacija

Jednostavna mutacija



Potpuna miješajuća mutacija (broj jedinica i nula ostaje isti)

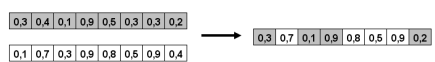


15

## Polje realnih brojeva, križanje

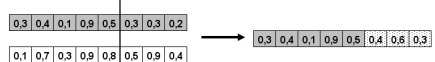
Diskretna rekombinacija

Primjer: diskretna uniformna rekombinacija



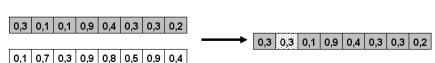
*Discrete recombination*

Jednostavna aritmetička rekombinacija



*Simple arithmetic recombination*

Jednostruka aritmetička rekombinacija

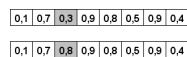


*Single arithmetic recombination*

16

## Polje realnih brojeva, mutacija

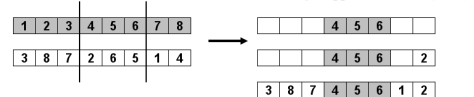
Jednostavna mutacija



17

## Permutirani niz cijelih brojeva, križanje

Djelomično mapirano križanje



*Partially mapped crossover (PMX)*

Križanje u poretku



*Order crossover (OX)*

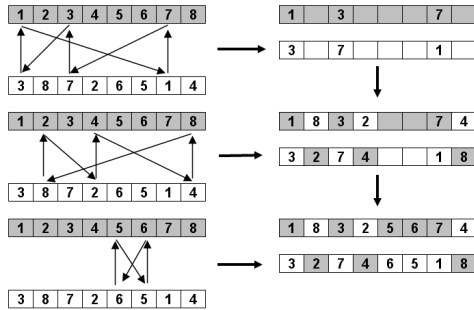
Počevši od druge točke križanja preslikaj sve nerasporedene elemente.

18

## Permutirani niz cijelih brojeva, križanje

Kružno križanje

Cycle crossover (CX)

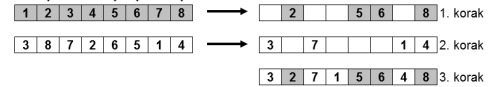


19

## Permutirani niz cijelih brojeva, križanje

Križanje zasnovano na lokaciji

Position-Based crossover (PBX)

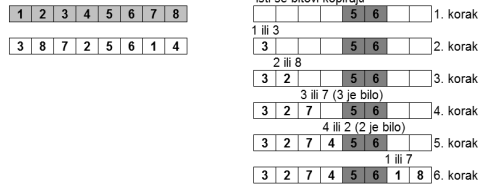


20

## Permutirani niz cijelih brojeva, križanje

Uniformno križanje nad permutacijskim nizom

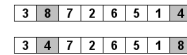
Uniform like crossover (ULX)



21

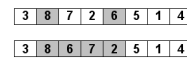
## Permutirani niz cijelih brojeva, mutacija

Jednostavna mutacija (mutacija zamjenom) Swap mutation



Mutacija ubacivanjem

Insert mutation



Nasumično se odabiru dva elementa i tada se jedan premješta da oni budu susjedni.

Mutacija premetanjem

Scramble mutation



22

## Teorem sheme i hipoteza blokova

- teorijske osnove genetskog algoritma
- odnose se na genetski algoritam s binarnim prikazom
- pretpostavke:
  - populacija je neograničena,
  - funkcija dobrote vjerno odražava problem i
  - geni u kromosomu su međusobno nezavisni.

23

## Teorem sheme

- Schema
  - uzorak ili skup rješenja koja su međusobno slična
  - \*101\*1 predstavlja skup od četiri rješenja:
    - 010101
    - 010111
    - 110101
    - 110111
  - zvjezdica "\*" označava "bilo što" ili "bilo koji znak", a u ovom slučaju binarnog prikaza to je 0 ili 1
- za kromosom duljine  $n$  i za shemu sa  $r$  znakova "\*"
  - takva shema predstavlja  $2^r$  rješenja
  - neko rješenje može biti predstavljeno s  $2^n$  shema
  - ukupan broj shema kod binarnog prikaza rješenja iznosi  $3^n$  (jer jedan znak sheme može biti "\*", "0" ili "1")

24

## Teorem sheme

- Red sheme,  $o(S)$ 
  - broj nepromjenjivih (fiksni) gena, odnosno broj svih onih znakova sheme  $S$  koji nisu  $u_{fix}$
  - $o(S) = n - r$
  - primjer:  $o(**1*001) = 4$  i  $o(****11*****) = 2$
- Definirana dužina sheme,  $\delta(S)$ 
  - udaljenost između prve i zadnje nepromjenjive znamenke
  - primjer:  $\delta(*01110011*) = 9 - 2 = 7$
  - $\delta(10**0) = 5 - 1 = 4$
  - $\delta(**1**) = 3 - 3 = 0$ .

25

## Teorem sheme

**Broj jedinki koje sadrže shemu niskog reda, kratke definirane dužine i iznadprosječne dobrote raste eksponencijalno:**

$$N(S, t+1) \geq N(S, t) \frac{\overline{D_S}}{D} \left[ 1 - \frac{\delta(S)}{n-1} p_c - o(S) p_m \right]$$

$N(S, t)$  - očekivani broj jedinki koje su podskup sheme  $S$  u generaciji  $t$   
 $\overline{D_S}$  - prosječna dobrot sheme  
 $D$  - prosječna dobrot populacije (3.12)  
 $\delta(S)$  - definirana dužina sheme  $S$   
 $o(S)$  - red sheme  $S$   
 $p_c$  - vjerojatnost križanja  
 $p_m$  - vjerojatnost mutacije  
 $n$  - duljina kromosoma

26

## Hipoteza građevnih blokova

**Genetski algoritam pretražuje prostor rješenja nizajući sheme niskog reda, kratke definirane dužine i iznadprosječne dobrote, zvane građevni blokovi.**

- Decepcija
  - građevni blokovi mogu navesti genetski algoritam i na krivo rješenje, tj. da algoritam konvergira lokalnom optimumu
  - ta se pojava naziva decepcija, a funkcija koja izaziva tu pojavu decepcijska ili varajuća funkcija
  - primjer *potpunog decepcijskog problema*:
    - funkcija koja vraća broj jedinica u kromosomu veličine  $n$ , a za kromosom ispunjen nulama ( $n$  nula) vraća  $2n$

27

## Postavljanje parametara algoritma

- Djelotvoran evolucijski algoritam ima **uravnoteženo slučajno i usmjereno pretraživanje** prostora rješenja i uravnoteženu selekciju roditelja i selekciju jedinki za novu populaciju.
- Kako postići tu ravnotežu?
  - Odrediti skup parametara EA
  - Koje su početne (preporučene) vrijednosti parametara
  - Podesiti parametre EA
    - Najbolje bi bilo optimirati parametre (Kako? Koristeći EA? Osim toga, predugo traje!)
    - Bolje, držati se preporuka i znati tendencije promjena kada se mijenja jedan od parametara.

28

## Postavljanje parametara algoritma

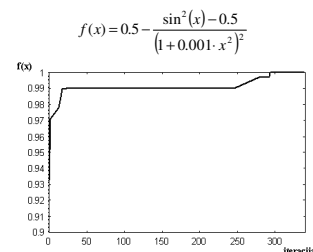
- Odrediti skup parametara EA
  - odabrati algoritam s manje parametara :)
  - eliminacijska selekcija s parametrom  $M$ , ali eliminira parametar  $p_c$
  - međutim, poželjno je da selekcija ima parametar s kojim se podešava selekcijski pritisak ( $k$ -turnirska selekcija)
  - bolje premali, nego preveliki selekcijski pritisak
  - eksperimentalno se pokazalo da je EA najosjetljiviji na parametar  $p_m$
- Preporuke za početne vrijednosti parametara:
  - $M \approx 50\%$
  - $p_m \approx 1\%$
  - $VEL \in [20, 100]$
  - $k = 3$
  - broj evaluacija  $\in [10^3, 10^7]$

(broj evaluacija za GA s  $k$ -turnirskom eliminacijskom selekcijom jednak je broju iteracija!)

29

## Primjer evolucijskog procesa

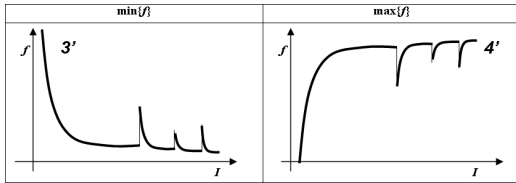
- Genetski algoritam koji iz generaciju u generaciju čuva najbolju jedinku asimptotski teži ka globalnom optimumu
- primjer: evolucijski proces optimiranja jednostavne funkcije uz pomoć GA s ugrađenim **elitizmom**



30

## Karakteristične krivulje

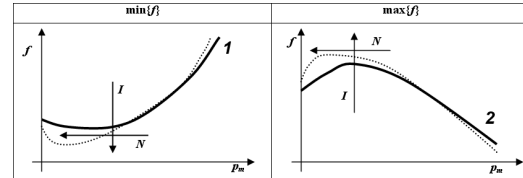
- ♦ karakteristična krivulja EA bez elitizma



31

## Međuzavisnosti parametara

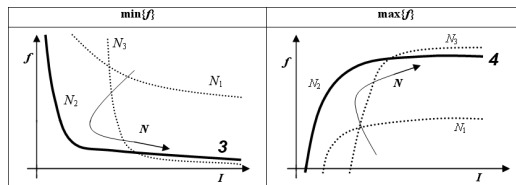
- ♦ pomaže pri određivanju vrijednosti parametara
- ♦ ovisnost kvalitete rješenja o vjerojatnosti ( $p_m$ ) mutacije, broju iteracija ( $I$ ) i veličini populacije ( $N$ ):



32

## Međuzavisnosti parametara

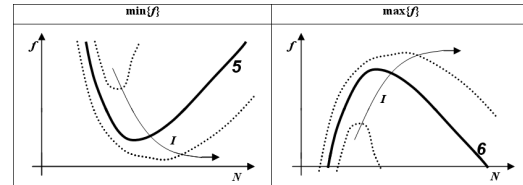
- ♦ utjecaj veličina populacije ( $N_1 < N_2 < N_3$ ) i broja iteracija ( $I$ ):



33

## Međuzavisnosti parametara

- ♦ utjecaj parametara veličine populacije ( $N$ ) i broja iteracija ( $I$ ):



34

## Umjesto zaključka

- EA s ugrađenim **elitizmom** asimptotski teži ka globalnom optimumu.
- Selekcija koja eliminira samo najgore jedinke, odnosno **veliki selekcijski pritisak** ima za posljedicu ranu konvergenciju ka **lokalnom optimumu**.
- EA s **premalim selekcijskim pritiskom** svodi se na **slučajno pretraživanje** prostora rješenja.
- EA nije pogodan za rješavanje **decepcijskih** optimizacijskih problema.
- Efikasan evolucijski algoritam ima **uravnoteženo slučajno i usmjereno pretraživanje** prostora rješenja i uravnoteženu selekciju roditelja i selekciju jedinki za novu populaciju. **Kako postići tu ravnotežu?**
  - ♦ Podesiti parametre EA.

35