Automatska paralelizacija uporabom genetskog programiranja

Ivo Majić

Mentor: Doc.dr.sc. Domagoj Jakobović

Završni rad

06. srpanj 2011.

Sadržaj predavanja

- Uvod
- Programsko ostvarenje
  - Prikaz jedinke
  - Primjer primjene transformacija
  - Evaluacija dobrote jedinke
  - Petlje i uvjetni izrazi
- Rezultati
  - Opis mjerenja
  - Analiza i grafički prikaz rezultata
- Zaključak

- Trajanje prezentacije: 13 min

Uvod

- automatsko generiranje paralelnih iz slijednih algoritama
- prikaz jedinke koji omogućava
  - dinamičko raspoređivanje instrukcija
  - primjenu standardnih genetskih operatora
- ulaz je slijedni algoritam, a izlaz njegova paralelna verzija
- na slijedni algoritam primjenjujemo skup transformacija
- temeljna relacija

$$SEQ(A,B) = PAR(A,B)$$

Programsko ostvarenje
Prikaz jedinke

- Skup transformacija
  - **PXX / SXX** - funkcije
    - Uzima XX% instrukcija iz odsječka i izvodi ih paralelno (P) ili slijedno (S) s ostatkom instrukcija u odsječku
  - **FXXX / LXXX** - funkcije
    - Uzima prvu (F - *first*) ili zadnju (L - *last*) instrukciju i izvodi je ovisno o XXX paralelno (PAR) ili slijedno (SEQ) s ostalim instrukcijama u odsječku
  - **SHIFT** - funkcija
    - Odgađa izvođenje svih instrukcija u odsječku za jednu vremensku jedinicu
  - **NULL / PARNULL** - terminali
    - Izvodi sve instrukcije u odsječku slijedno (NULL) ili paralelno (PARNULL)

Programsko ostvarenje
Prikaz jedinke

- Vjerojatnosti odabira postotka **PXX / SXX** transformacija korištenih prilikom stvaranja inicijalne populacije (te prilikom mutacije)

Vjerojatnost	Vrijednost (XX)
25%	50
15%	25
15%	75
15%	33
15%	66
15%	random(1, 99)

Programsko ostvarenje
Primjer primjene transformacija

- Odsječak od 5 slijednih instrukcija

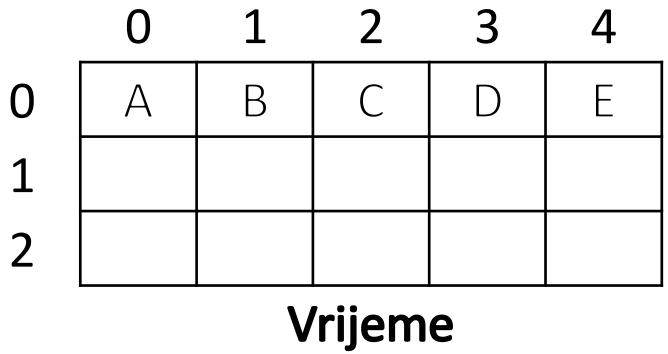
A: $a = (3 + c) * a$

B: $d = c + 7$

C: $d = d * (a * (e - 4))$

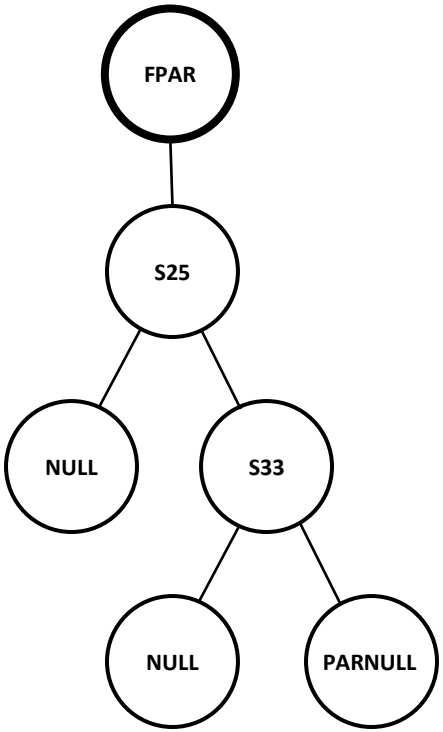
D: $e = (d - 3) * 4$

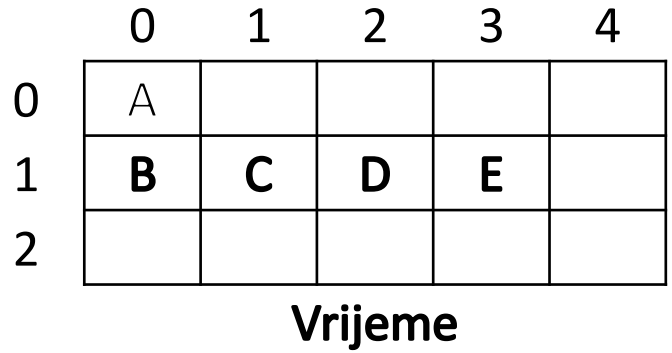
E: $f = a * d$



Programsko ostvarenje
Primjer primjene transformacija

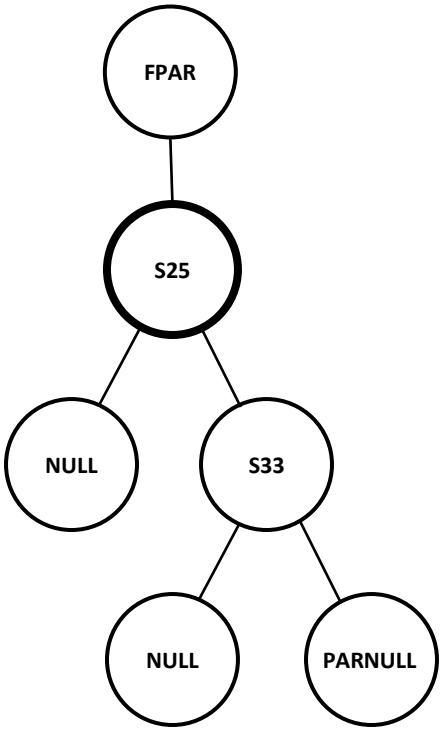
- Korak 1 - FPAR

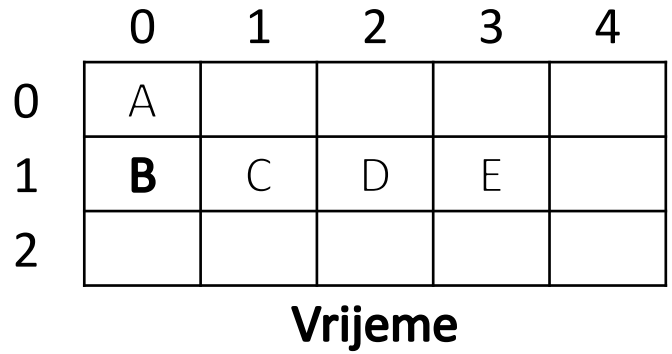




Programsko ostvarenje
Primjer primjene transformacija

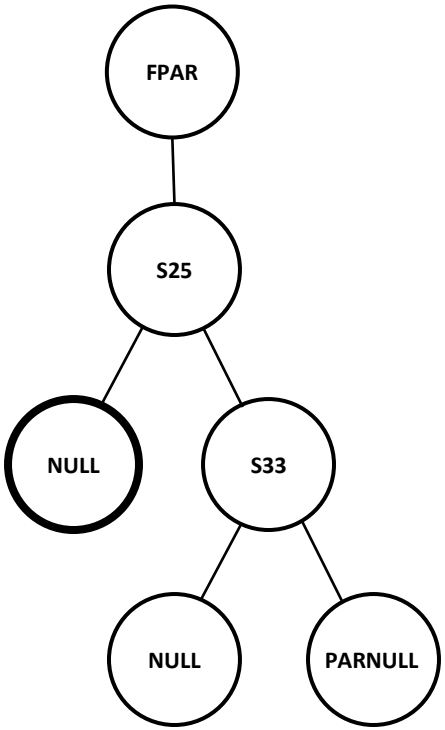
- Korak 2 – S25





Programsko ostvarenje
Primjer primjene transformacija

- Korak 3 - NULL

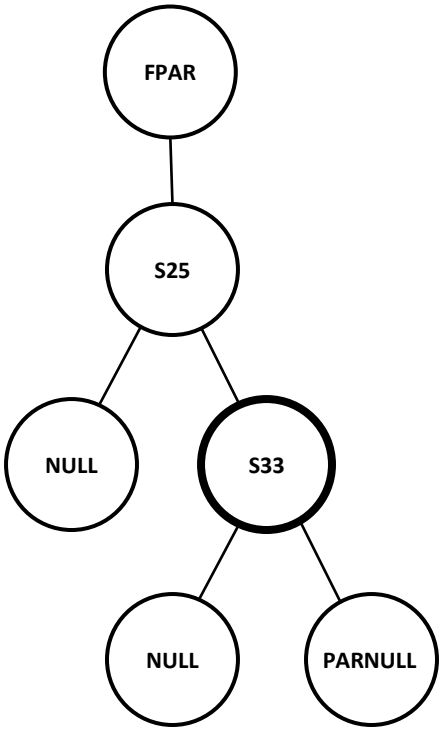


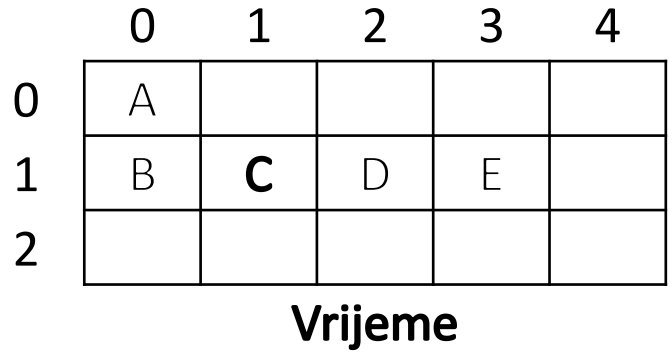
	0	1	2	3	4
0	A				
1	B	**C**	**D**	**E**	
2					

Vrijeme

Programsko ostvarenje
Primjer primjene transformacija

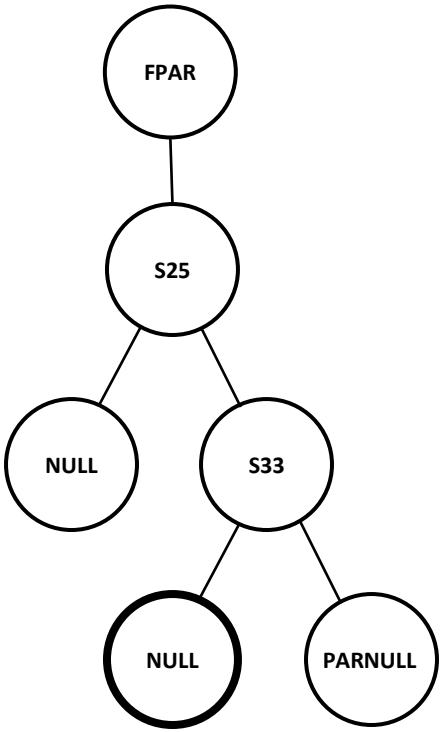
- Korak 4 - S33

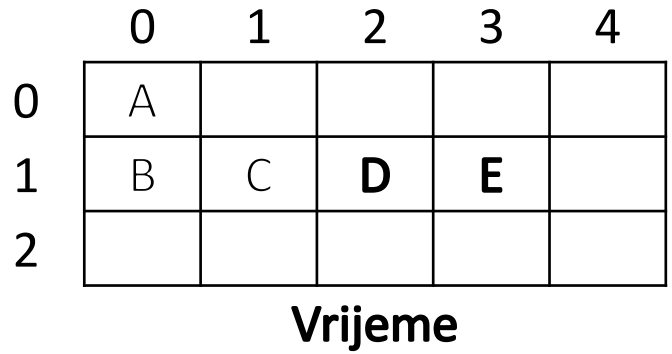




Programsko ostvarenje
Primjer primjene transformacija

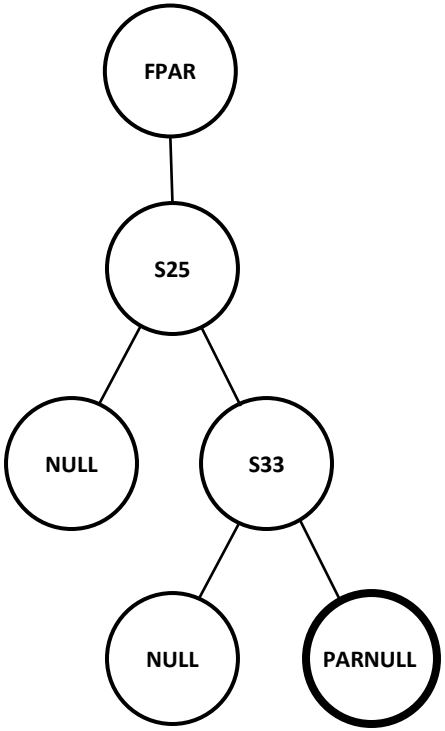
- Korak 5 - NULL

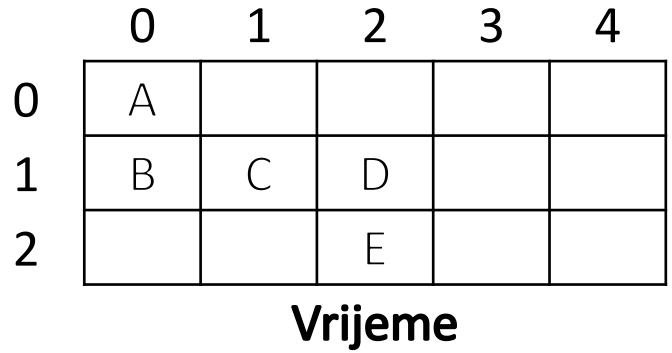




Programsko ostvarenje
Primjer primjene transformacija

- Korak 6 - PARNULL





Programsko ostvarenje
Evaluacija dobrote jedinke

- Bernsteinovi uvjeti - podatkovne ovisnosti

U_X - *skup ulaznih varijabli instrukcije X*
I_X - *skup ulaznih varijabli instrukcije X*

$$(U_A \cap I_B) \cup (I_A \cap U_B) \cup (I_A \cap I_B) = \varnothing$$

- Primjer

$$A: \mathrm{a} = (\mathrm{b} + \mathrm{c}) * 3 \quad \rightarrow \quad U_A = \{b, c\} \qquad I_A = \{a\}$$
$$B: \mathrm{d} = \mathrm{a} * (\mathrm{d} + 5) \quad \rightarrow \quad U_B = \{a, d\} \qquad I_B = \{d\}$$

$$(U_A \cap I_B) \cup (I_A \cap U_B) \cup (I_A \cap I_B) = \{a\}$$

Programsko ostvarenje
Evaluacija dobrote jedinke

Instrukcija	A	B	C	D	E
Vremenski korak	0	0	1	2	2

- Korektnost
  - podatkovne ovisnosti (Bernstein) instrukcija koje se izvode paralelno
  - očuvanost redoslijeda izvršavanja (toka programa) podatkovno ovisnih instrukcija
  - u prijašnjem primjeru – 0 (savršena ocjena)
- Paralelnost
  - Konačan broj vremenskih koraka potrebnih za izvođenje algoritma
  - u prijašnjem primjeru – 3

Programsko ostvarenje
Petlje i uvjetni izrazi

- Primjer – računanje broja π

Algoritam Računanje broja π

```
b_tocaka = 100000
b_tocaka_krug = 0
for b_tocaka do
    R1 = slucajan_broj (1, 2r)
    R2 = slucajan_broj (1, 2r)
    if tocka(R1, R2) in krug then
        b_tocaka_krug ++
    end if
end for
pi = 4 * (b_tocaka_krug / b_tocaka)
```

Programsko ostvarenje
Petlje i uvjetni izrazi

- Instrukcijske razine

A: *b_tocka = 10000*

B: *b_tocaka_krug = 0*

C: *metaFOR*

D: *pi = 4 * (b_tocaka_krug / b_tocaka)*

(Razina 1)

A: *R1 = slucajan_broj(1, 2r)*

B: *R2 = slucajan_broj(1, 2r)*

C: *metaIF*

(Razina 2 - metaFOR)

A: *b_tocaka_krug ++*

(Razina 3 - metaIF)

- Primjer ulaznih i izlaznih varijabli – metaFOR

$$I_{metaFOR} = \{R1, R2, b\_tocaka\_krug\}$$

$$U_{metaFOR} = \{b\_tocaka\_krug, b\_tocaka\}$$

Programsko ostvarenje
Petlje i uvjetni izrazi

- Primjer – računanje broja π

Algoritam Računanje broja π

```
[PARBEGIN]
b_tocaka = 100000
b_tocaka_krug = 0
[PAREND]
parfor b_tocaka do
   R1 = slucajan_broj (1, 2r)
   R2 = slucajan_broj (1, 2r)
   if tocka(R1, R2) in krug then
      b_tocaka_krug ++
   end if
end parfor
pi = 4 * (b_tocaka_krug / b_tocaka)
```

Rezultati
Opis mjerenja

- Odsječak od 9 neovisnih instrukcija
- Isključena PARNULL transformacija

a = a + 1
b = b + 2
c = c + 3
d = d + 4
e = e + 5
f = f + 6
g = g + 7
h = h + 8
i = i + 9

(Slijedno)

[PARBEGIN]
a = a + 1
b = b + 2
c = c + 3
d = d + 4
e = e + 5
f = f + 6
g = g + 7
h = h + 8
i = i + 9
[PAREND]

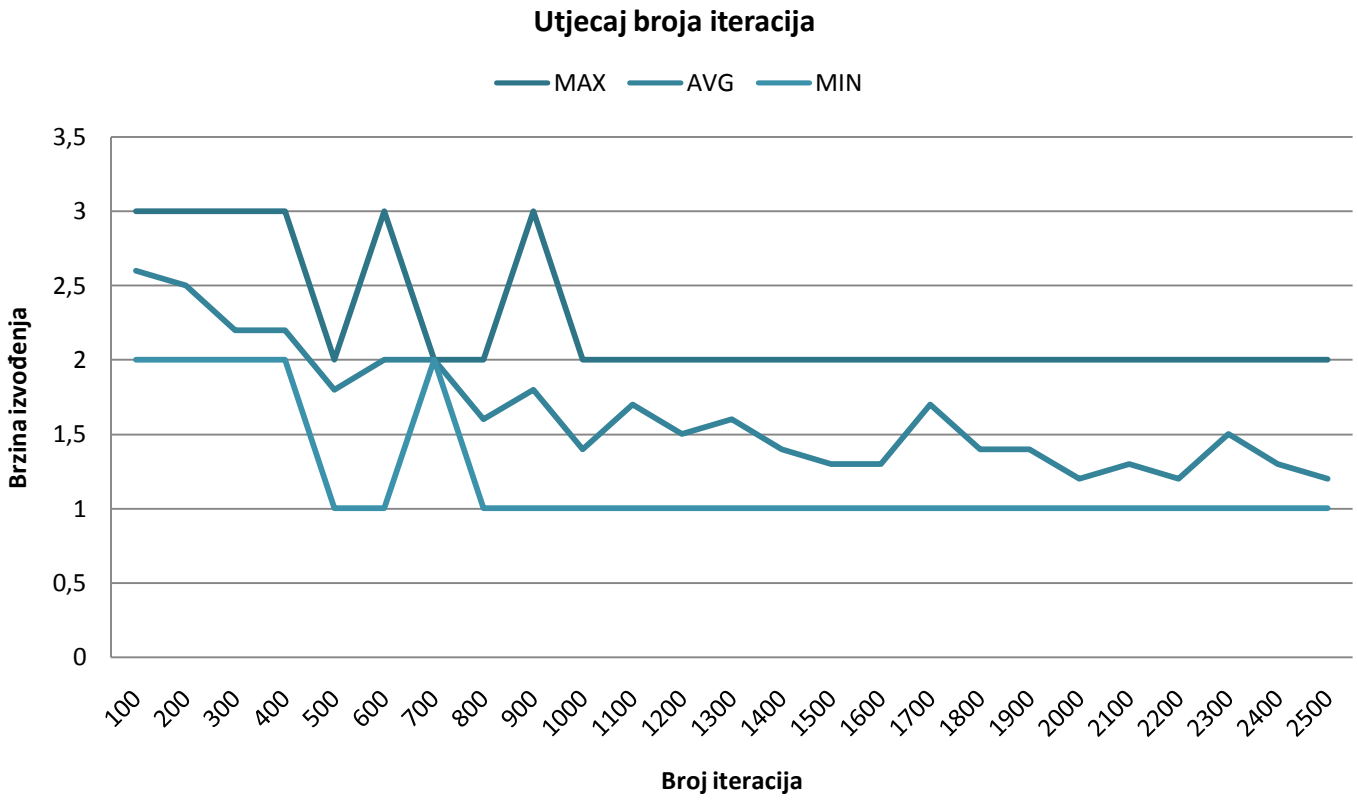
(Paralelno)

Rezultati
Opis mjerenja

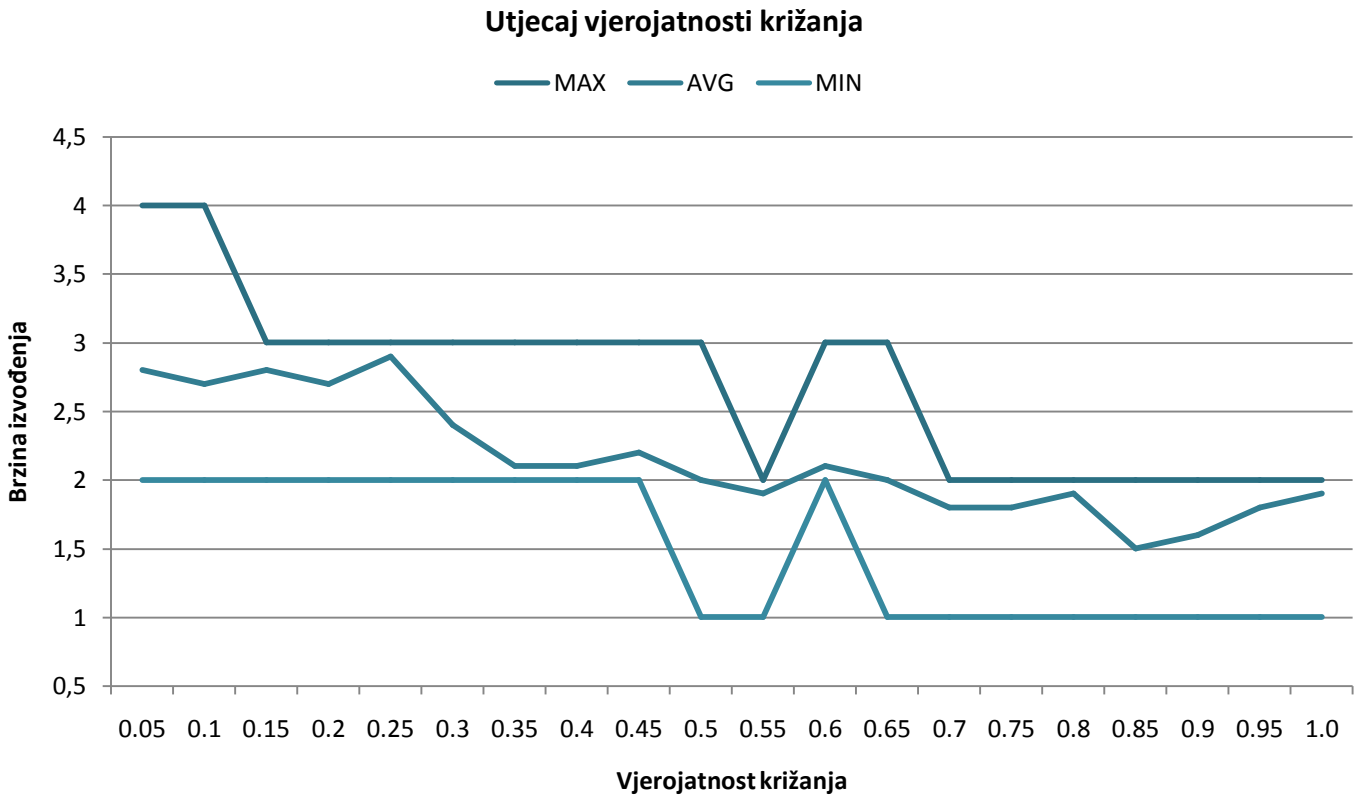
- Analiziran je
  - utjecaj **broja iteracija – u intervalu [100, 2500], korak 100**
  - utjecaj **vjerojatnosti križanja – u intervalu [0, 1]**
  - utjecaj **vjerojatnosti mutacije – u intervalu [0, 1]**
- Promatramo utjecaj parametara na razinu paralelizacije
- Inicijalne vrijednosti parametara

Parametar	Broj iteracija	Populacija	Mutacija	Križanje	Dubina
Vrijednost	600	200	0.2	0.9	9

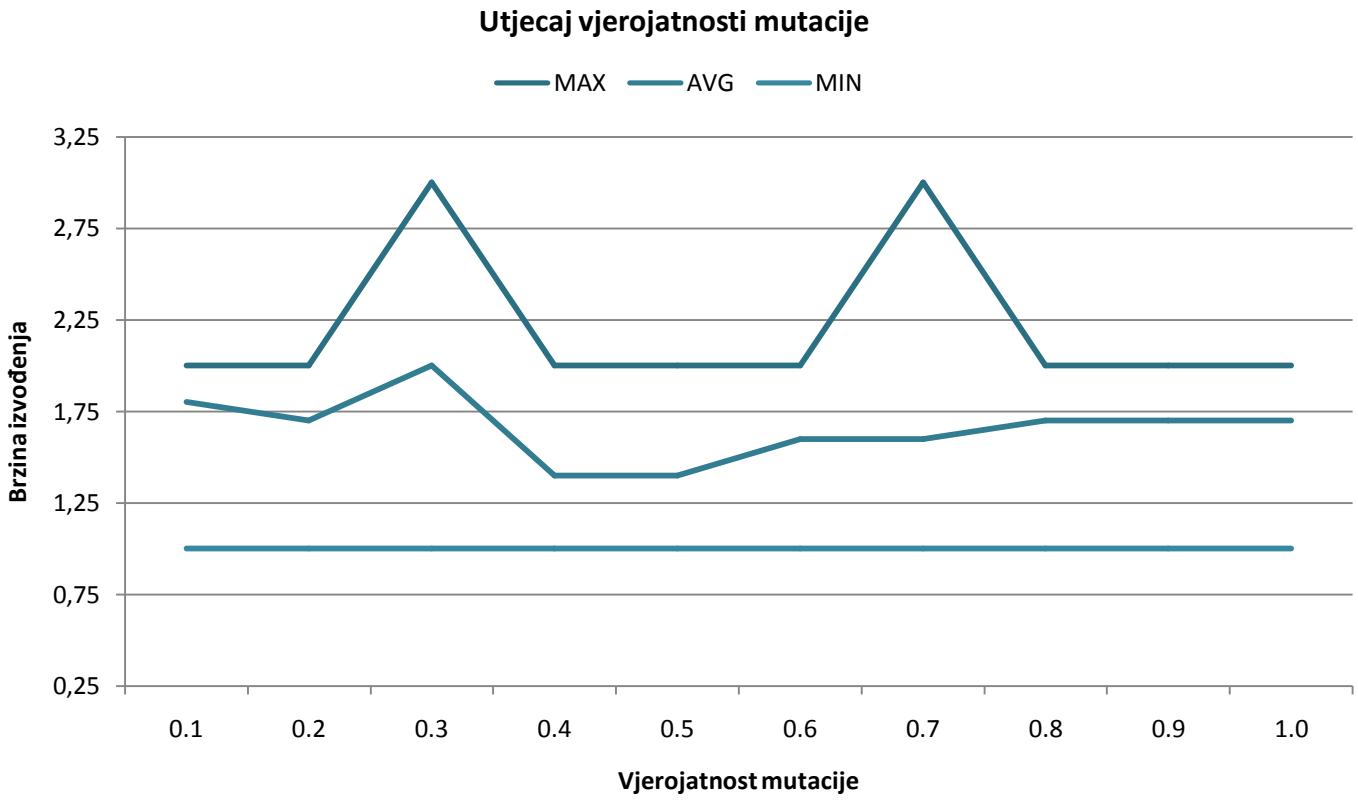
Rezultati
Analiza i grafički prikaz rezultata



Rezultati
Analiza i grafički prikaz rezultata



Rezultati
Analiza i grafički prikaz rezultata



Zaključak

- Demonstracija rada implementacije
- Pitanja?