**DIPLOMSKI RAD br. 1482**

ZEMRIS FER

# HIPERHEURISTIČKE METODE RJEŠAVANJA PROBLEMA RASPOREĐIVANJA U OKRUŽENJU NESRODNIH STROJEVA

*Hrvoje Backović*

Voditelj: izv. prof. dr. sc. Domagoj Jakobović

Zagreb, 11.7.2017.

# Sadržaj



- Raspoređivanje u okruženju nesrodnih strojeva
- Raspoređivanje uz pomoć pravila
- Kartezijsko genetsko programiranje
- Gramatička evolucija
- Neuronske mreže
- Rezultati i analiza
- Zaključak

# Raspoređivanje u okruženju nesrodnih strojeva



- Skup poslova potrebno rasporediti na dostupni skup resursa
- Primjeri:
  - Raspoređivanje letova po pistama
  - Raspoređivanje dretvi po procesorima
- Optimizacijski problem, minimizacija kriterija
- NP-težak problem
- Heuristička rješenja
  - Pretraživanje prostora stanja
  - Iterativna izgradnja rješenja

# Raspoređivanje u okruženju nesrodnih strojeva



| Oznaka | Opis |
|---|---|
| $p_{ij}$ | vrijeme izvođenja posla $j$ na stroju $i$ |
| $r_j$ | vremenski trenutak kada posao $j$ dolazi u sustav |
| $d_j$ | rok, vremenski trenutak kada posao $j$ treba biti gotov, u suprotnom dolazi do određenog gubitka zbog kašnjenja |
| $w_j$ | težina, važnost posla |

**Tablica 2.1:** Svojstva problema raspoređivanja

# Raspoređivanje u okruženju nesrodnih strojeva



| Oznaka | Opis |
|---|---|
| $C_j$ | vremenski trenutak kada posao $j$ završi s izvođenjem |
| $F_j$ | tok, vrijeme koje je posao $j$ proveo u sustavu. Definira se kao $F_j = C_j - r_j$ |
| $T_j$ | zakašnjelost, vrijeme koliko se posao $j$ izvršavao nakon roka $T_j = max(C_j - d_j, 0)$ |
| $U_j$ | je li posao $j$ zakasnio, $U_j = \begin{cases} 1 & \text{if } T_j > 0 \\ 0 & \text{if } T_j = 0 \end{cases}$ |

**Tablica 2.2:** Kriteriji za pojedinačne poslove

# Raspoređivanje u okruženju nesrodnih strojeva



| Oznaka | Opis |
|---|---|
| $Cmax$ | maksimalno vrijeme završetka izvođenja. $Cmax = max(Cj)$ |
| $Ft$ | ukupni tok. $Ft = \sum_{j=1}^{n} F_j$ |
| $Twt$ | ukupna težinska zakašnjelost. $Twt = \sum_{j=1}^{n} w_j T_j$ |
| $Uwt$ | težinski broj zakašnjelih poslova. $Uwt = \sum_{j=1}^{n} w_j U_j$ |

**Tablica 2.3:** Kriteriji po kojima se provodi optimizacija

# Raspoređivanje uz pomoć pravila



- Podjela raspoređivanja na:
  - Statičko – raspored konstruiran prije izvođenja sustava
  - Dinamičko – raspored konstruiran za vrijeme izvođenja sustava
- Raspored se dinamički gradi za vrijeme izvođenja sustava
- Često se koristi u uvjetima raspoređivanja u stvarnom vremenu
- Koristi se prioritetna funkcija
  - Ulaz je svaki par posao/stroj koji su dostupni
  - Izlaz je informacija uz pomoć koje sustav prelazi u sljedeće stanje

# Raspoređivanje uz pomoć pravila



- Meta-algoritam korištenjem prioritetne funkcije raspoređuje poslove
- Primjer meta-algoritma:

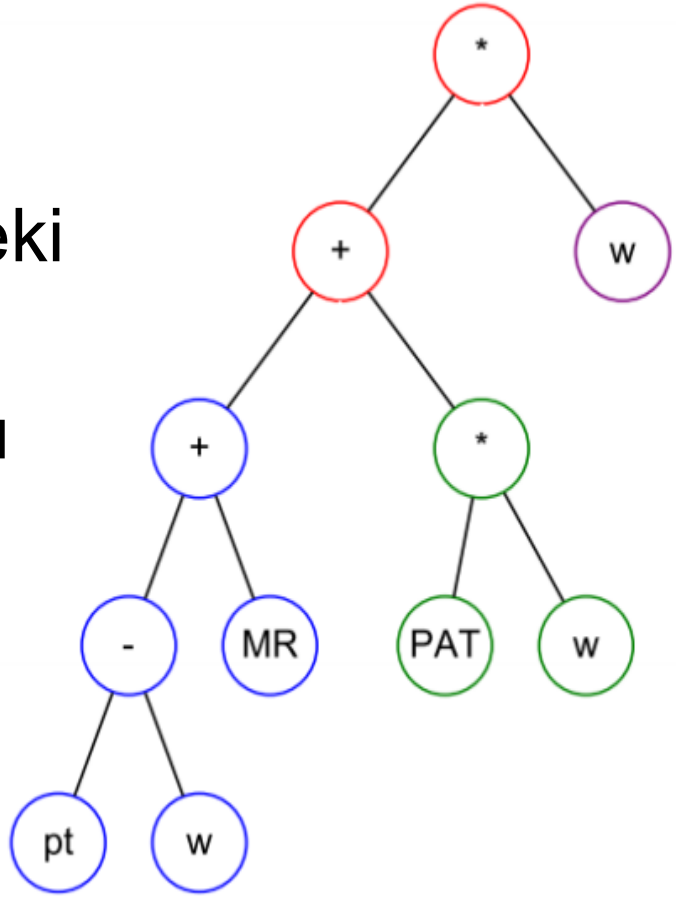
**dok** (postoje neraspoređeni poslovi)
{
    čekaj dok stroj ne postane raspoloživ;
    odredi prioritete $\pi_i$ svih neraspoređenih poslova;
    rasporedi posao s najboljim prioritetom na stroj;
}

- Prioritetna funkcija se evoluira s obzirom na optimirani kriterij
- Primjer prioritetne funkcije: $\pi_j = \frac{w_j}{p_j}$

# Genetsko programiranje



- Evolucijski algoritam za otkrivanje funkcija ili programa koji rješavaju neki problem
- Funkcije zapisane u obliku stabla
- Pretraživanje prostora funkcija uz pomoć evolucijskog algoritma (najčešće GA)




**Slika 3.1:** Funkcija zapisana u obliku stabla. $f = (((pt - w) + MR) + (PAT \cdot w)) \cdot w$

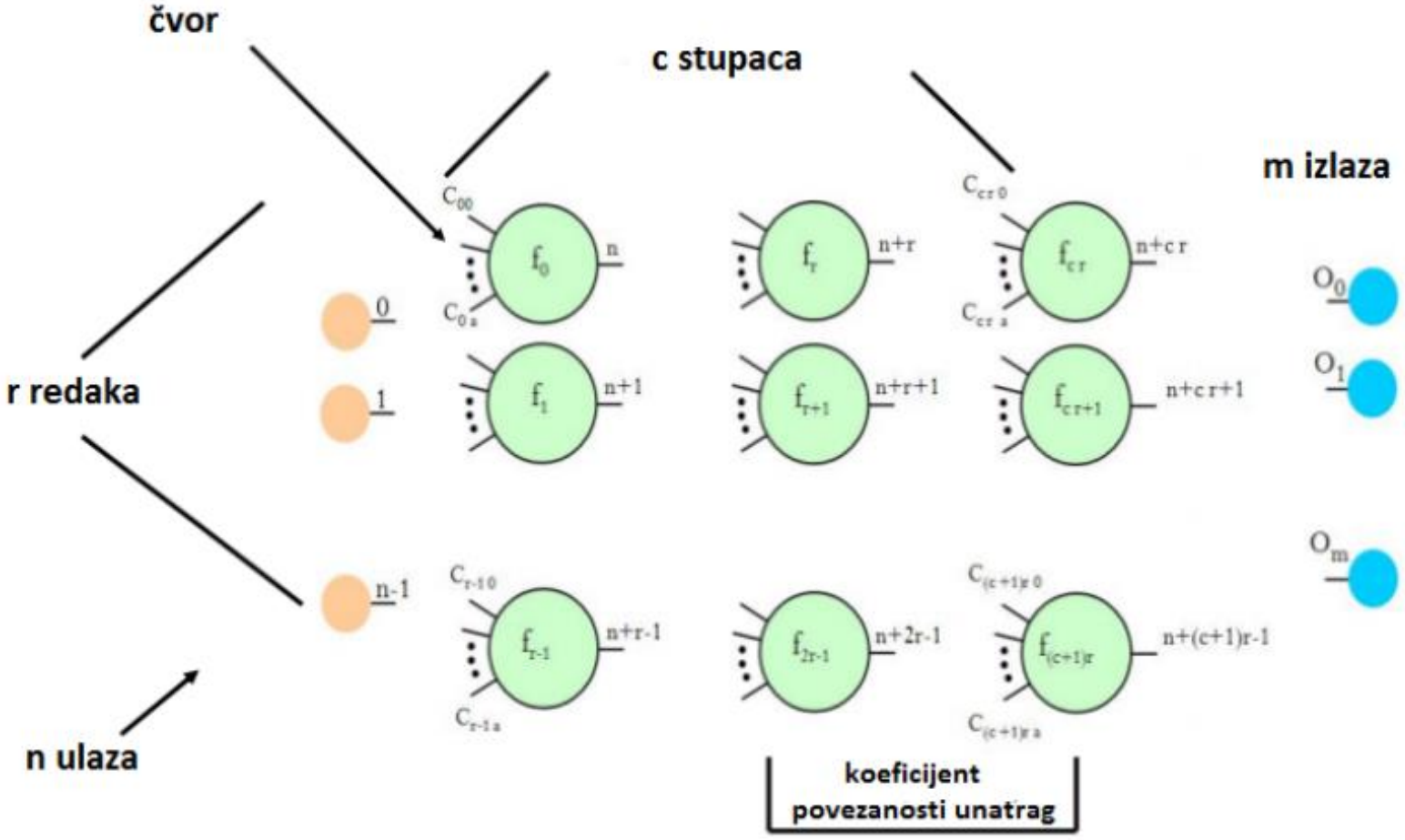
# Kartezijsko genetsko programiranje



- fleksibilna i efikasna varijanta GP-a
- Općenitija struktura - graf
- Kodira reprezentaciju strukture u obliku niza brojeva
- Moguće definirati veći broj izlaza funkcije
- Fiksna veličina genotipa

# Kartezijsko genetsko programiranje






**Slika 3.2:** Struktura genotipa

# Gramatička evolucija



- Evolucijski algoritam pretraživanja, sličan GP-u
- Generiranje programa čija je sintaksa zadana gramatikom
- Koristi se kontekstno-neovisna gramatika
- Genotip u obliku niza prirodnih brojeva
- Evolucijski operatori:
  - Jednoliko križanje
  - Mutacija jedne točke

# Gramatička evolucija



```
<expr>
<expr> : <expr> <op> <expr> | <subexpr> | ( <expr> )
<subexpr> : <func> ( <expr> ) | <var>
<func> : [pos]
<op> : + | - | * | /
<var> : [T_pt] | [T_dd] | [T_w] | [T_SL] | [T_pmin] | [T_pavg] | [T_PAT] | [T_MR] | [T_age]
```

**Slika 4.1:** Gramatika za generiranje prioritetnih funkcija

# Neuronske mreže



- Matematički model baziran na radu neuroloških sustava živih bića
- Brojne primjene u umjetnoj inteligenciji i strojnom učenju (klasifikacija, regresija, itd.)
- Učenje uz pomoć genetskog algoritma
- Prikaz rješenja u obliku niza decimalnih brojeva

# Rezultati i analiza



- Implementacija u jeziku C++ korištenjem ECF-a (*Evolutionary Computation Framework*)
- 120 primjera, 60 za učenje i 60 za testiranje
- 30 pokretanja za svaki eksperiment
- Kriterij zaustavljanja: 80000 evaluacija
- Bazni algoritmi:
  - Eliminacijski genetski algoritam s k-turnirskom selekcijom
  - Evolucijska strategija (samo za CGP)

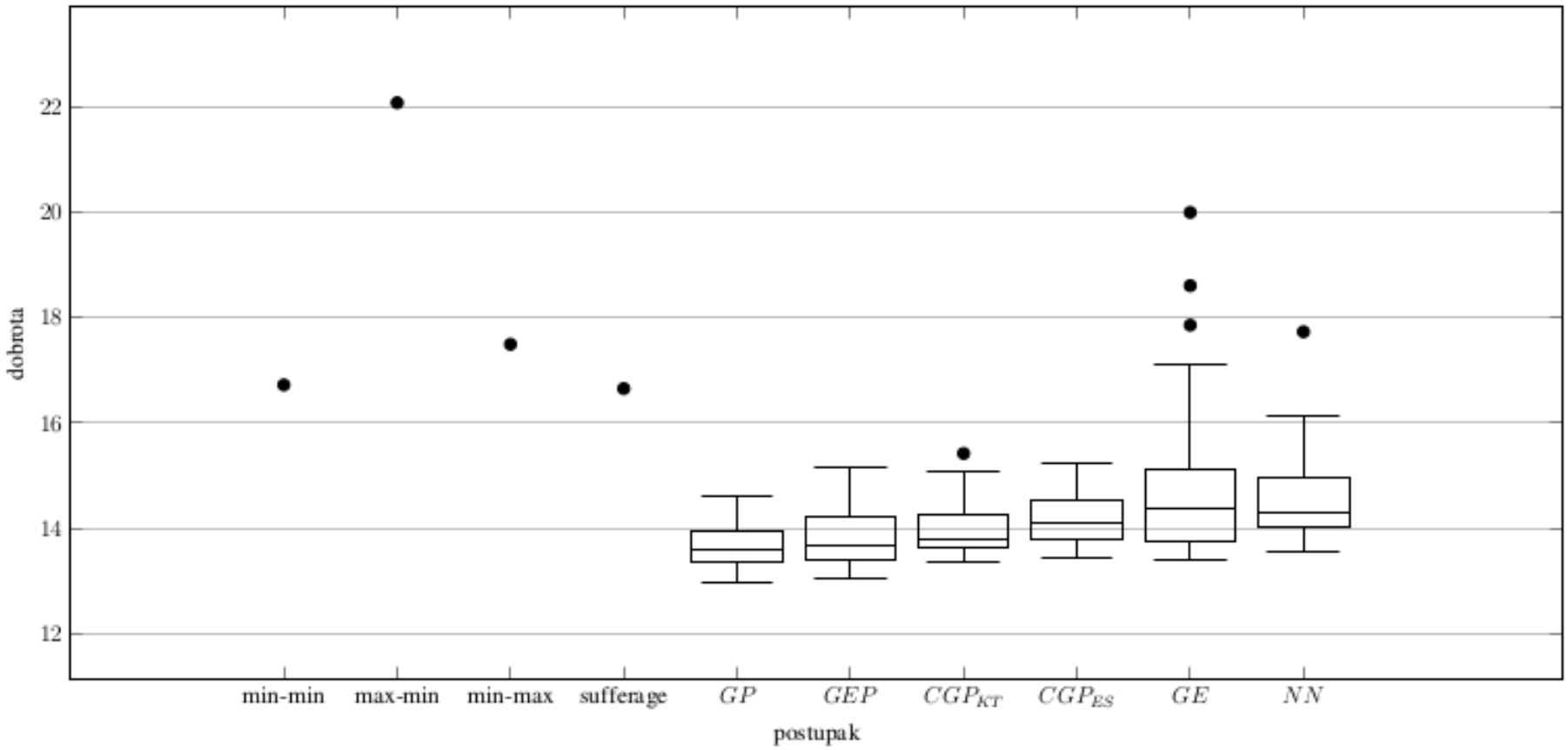
# Rezultati i analiza



- Optimiranje parametara
  - CGP – bazni algoritam (GA,ES), broj redaka(100-500), vjerojatnost mutacije(0.3-0.9), veličina populacije(500-1000,5-50)
  - GE – veličina genotipa(30-1000), vjerojatnost mutacije(0.3-0.9), veličina populacije(150-500)
  - NN – struktura mreže(1-2 skrivena sloja), vjerojatnost mutacije(0.3-0.9), veličina populacije(50-500)
- Bitno je pronaći optimalnu veličinu jedinke jer ona utječe na veličinu izraza, a time i na performanse

# Rezultati i analiza

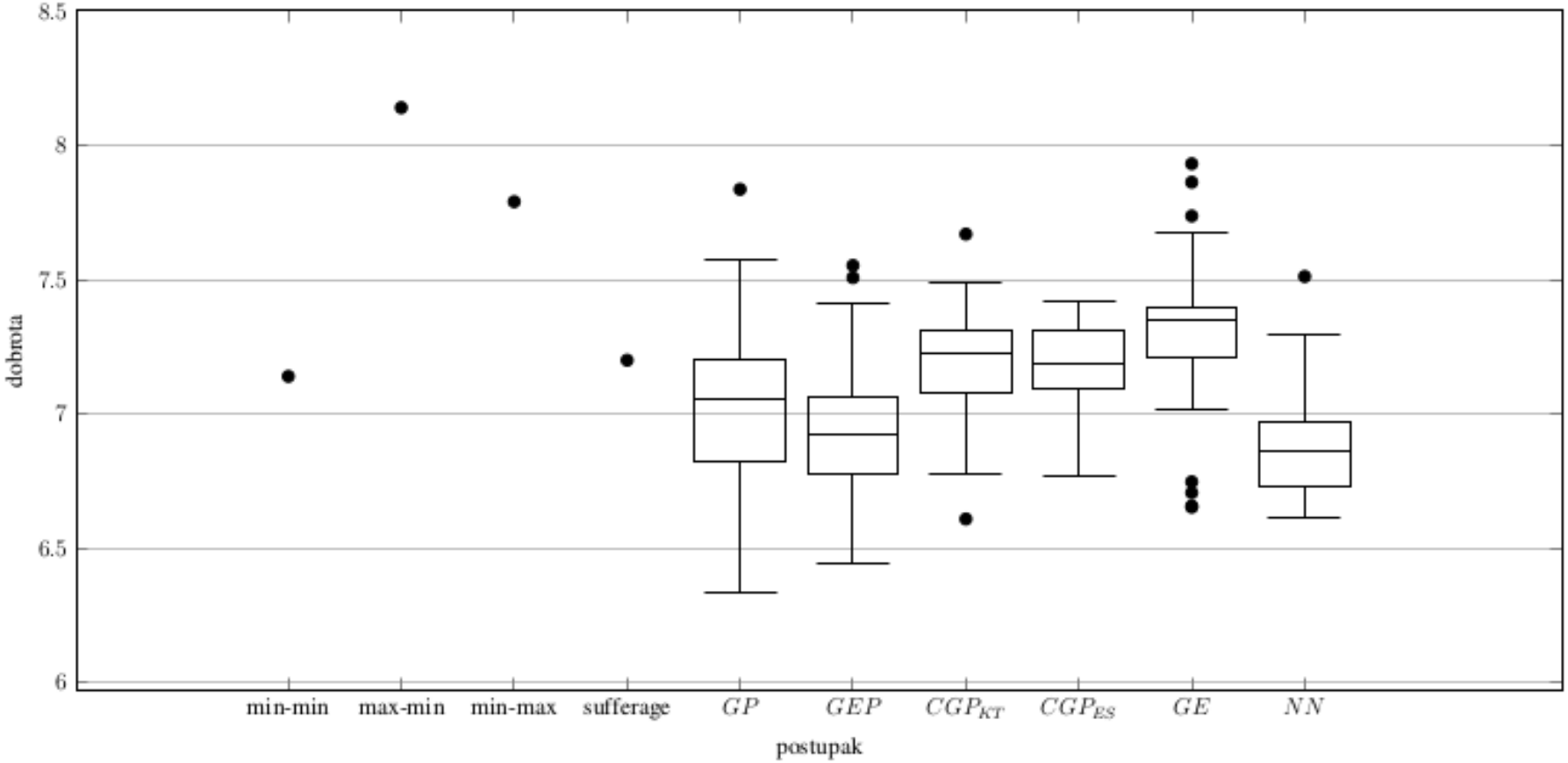





**Slika 6.4:** *Boxplot* prikaz za Twt kriterij

# Rezultati i analiza

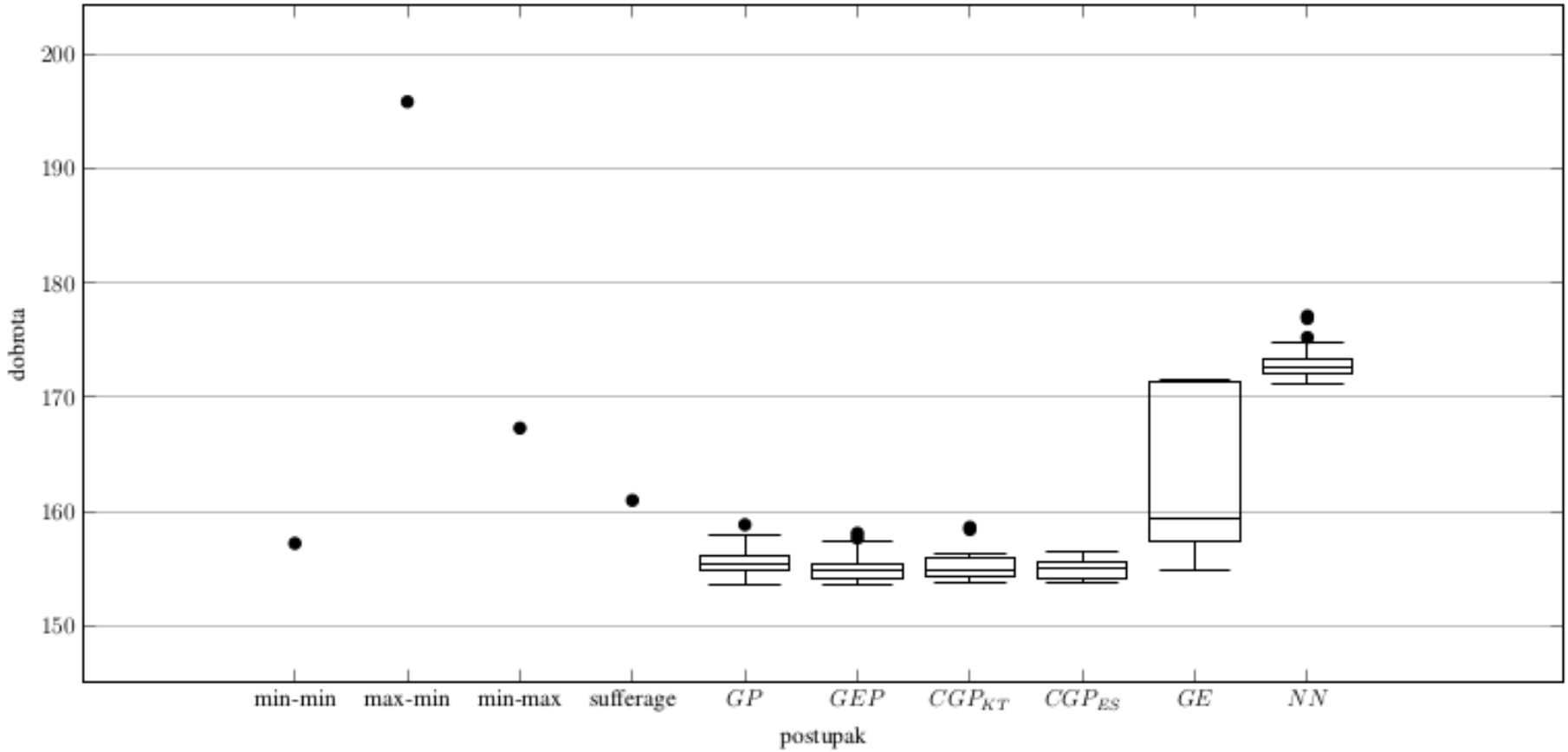





**Slika 6.3:** *Boxplot* prikaz za Uwt kriterij

# Rezultati i analiza

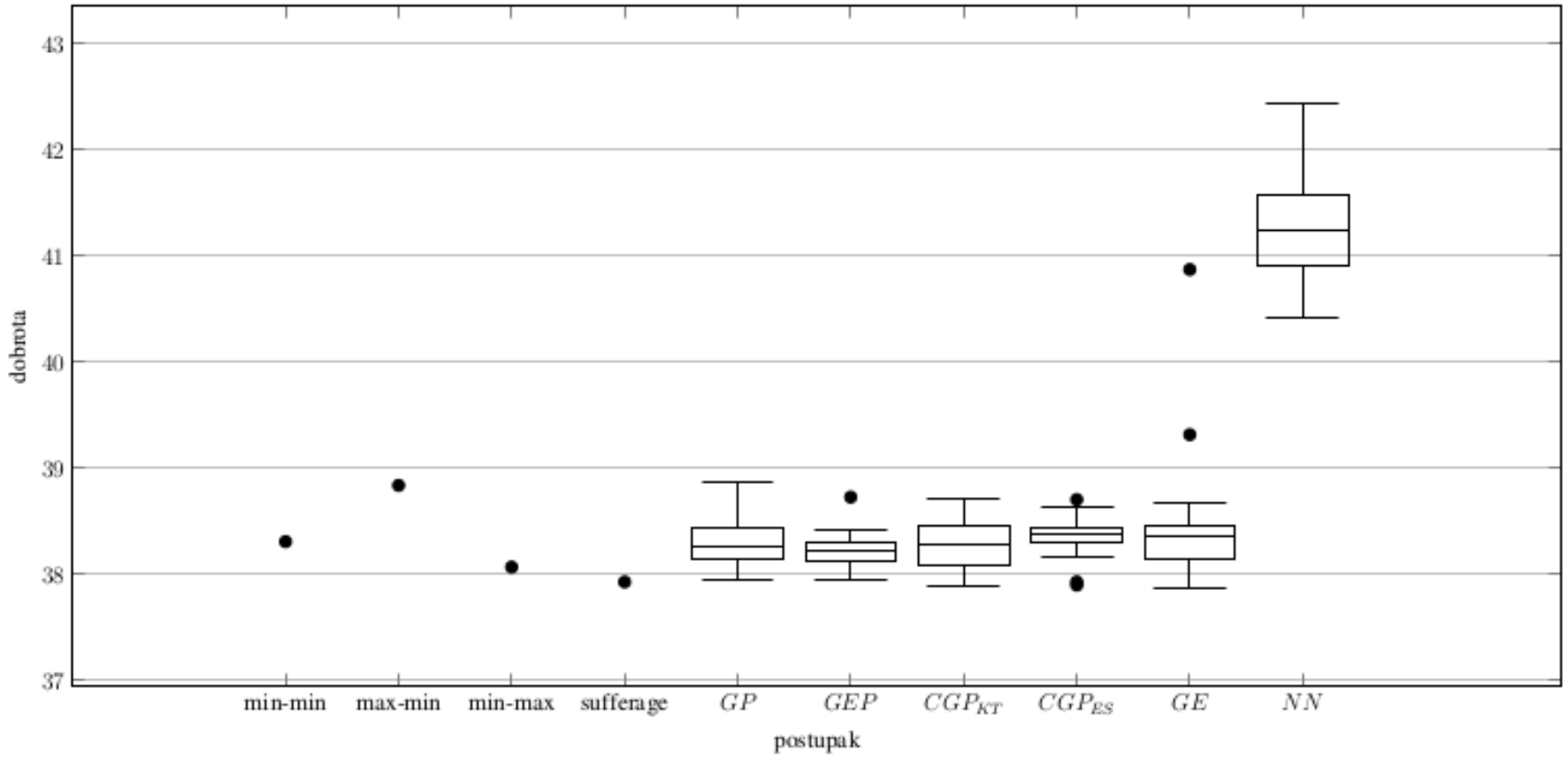





**Slika 6.2:** *Boxplot* prikaz za Ft kriterij

# Rezultati i analiza






**Slika 6.1:** *Boxplot* prikaz za Cmax kriterij

# Zaključak



- CGP i GE nadmašili postojeće heuristike te su bili konkurentni metodama klasičnog genetskog programiranja
- Neuronske mreže se nisu pokazale učinkovitima
- Moguća poboljšanja uvođenjem novih evolucijskih operatora i lokalnih operatora pretraživanja
- Problem raspoređivanja ima široku primjenu pa je važno istraživati nove pristupe njegovom rješavanju

Pitanja?

# Raspoređivanje uz pomoć pravila



| Oznaka | Opis |
|---|---|
| $pt$ | vrijeme izvođenja posla $j$ na stroju $i$ ($p_{ij}$) |
| $pmin$ | minimalno vrijeme izvođenja na svim strojevima |
| $pavg$ | srednje vrijeme izvođenja na svim strojevima |
| $PAT$ | strpljivost - vrijeme dok stroj s minimalnim vremenom izvođenja ne postane slobodan |
| $MR$ | vrijeme dok trenutni stroj ne postane slobodan |
| $age$ | vrijeme koje je posao proveo u sustavu |
| $dd$ | vremenski trenutak kad posao mora završiti ($d_j$) |
| $w$ | težina (prioritet) posla |
| $SL$ | dopuštena odgoda uz brzinu dotičnog stroja - $max(d_j - p_{ij} - time, 0)$ |

**Tablica 2.4:** Ulazni terminali prioritetne funkcije

# Raspoređivanje uz pomoć pravila



**Algoritam 1** Meta-algoritam za raspoređivanje zasnovano na prilagodljivim pravilima

```
1: while postoje neraspoređeni poslovi do
2:     čekaj dok posao ne postane dostupan ili se posao završi;
3:     for sve dostupne poslove i sve strojeve do
4:         izračunaj prioritet π_ij posla j na stroju i;
5:     end for
6:     for sve dostupne poslove do
7:         odredi najbolji stroj (onaj za koji se postiže najbolji iznos prioriteta π_ij);
8:     end for
9:     while postoje poslovi čiji je najbolji stroj dostupan do
10:        odredi najbolji prioritet od svih takvih poslova;
11:        rasporedi posao s najboljim prioritetom;
12:    end while
13: end while
```

# Raspoređivanje uz pomoć pravila



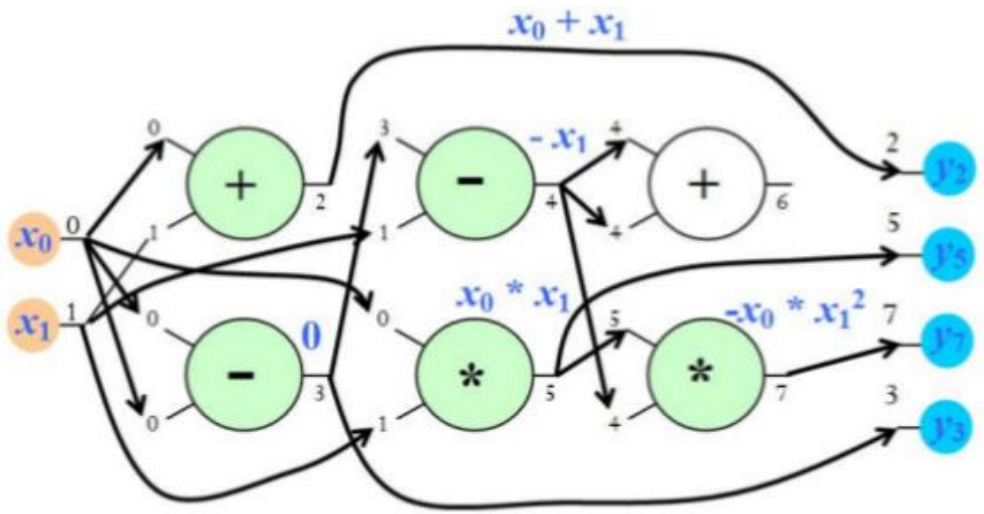
- $\pi_{ij} = pt - MR + pmin - \frac{pt - pmin - SL - MR}{p_t}$

# Kartezijsko genetsko programiranje



Genotip:

0 0 1 1 0 0 1 3 1 2 0 1 0 4 4 2 5 4 2 5 7 3




$$y_2 = x_0 + x_1$$
$$y_5 = x_0 * x_1$$
$$y_7 = -x_0 * x_1^2$$
$$y_3 = 0$$

**Slika 3.3:** Prikaz rješenja kartezijskog genetskog programiranja

# Kartezijsko genetsko programiranje



| Oznaka | Opis |
|---|---|
| $+$ | operator binarnog zbrajanja |
| $-$ | operator binarnog oduzimanja |
| $*$ | operator binarnog množenja |
| $/$ | sigurno dijeljenje: $/(a,b) = \begin{cases} 1 & \text{if } \lvert b \rvert < 0.000001 \\ \text{a / b} & \text{else} \end{cases}$ |
| $POS$ | unarni operator: $POS(a) = max(a, 0)$ |

**Tablica 3.1:** Funkcije unutarnjih čvorova

# Gramatička evolucija



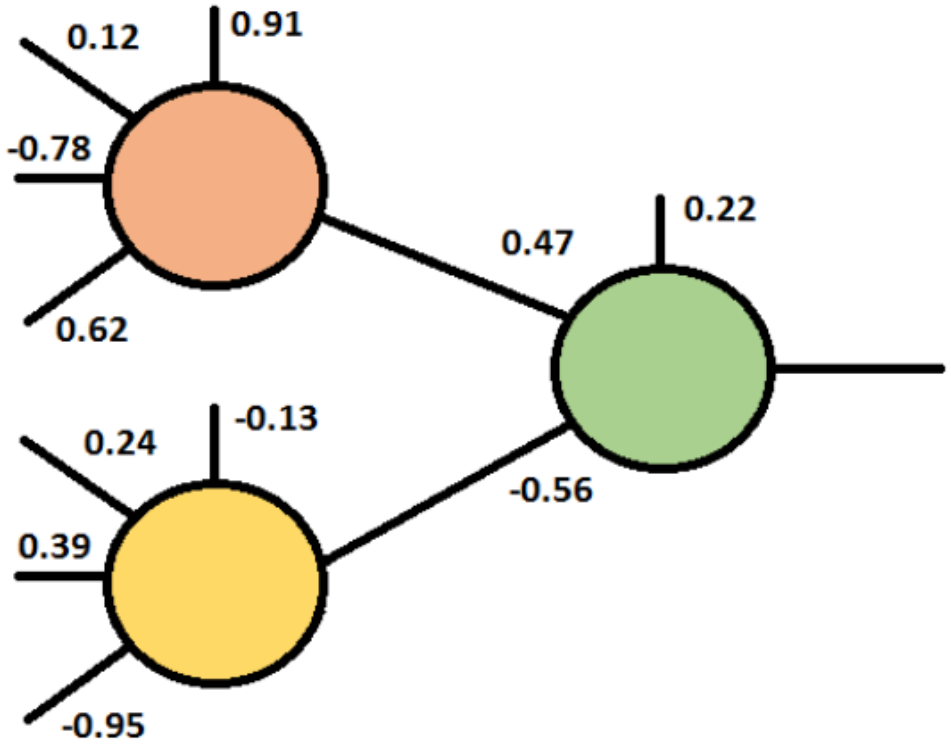
| prikaz | izraz |
|---|---|
| | <expr> |
| 0 | <expr> <op> <expr> |
| 7 | <subexpr> <op> <expr> |
| 5 | <var> <op> <expr> |
| 1 | [T_dd] <op> <expr> |
| 4 | [T_dd] + <expr> |
| 4 | [T_dd] + <subexpr> |
| 2 | [T_dd] + <func> ( <expr> ) |
| 3 | [T_dd] + pos ( <expr> ) |
| 0 | [T_dd] + pos ( <expr> <op> <expr> ) |
| 1 | [T_dd] + pos ( <subexpr> <op> <expr> ) |
| 3 | [T_dd] + pos ( <var> <op> <expr> ) |
| 0 | [T_dd] + pos ( [T_pt] <op> <expr> ) |
| 1 | [T_dd] + pos ( [T_pt] - <expr> ) |
| 7 | [T_dd] + pos ( [T_pt] - <subexpr> ) |
| 5 | [T_dd] + pos ( [T_pt] - <var> ) |
| 3 | [T_dd] + pos ( [T_pt] - [T_SL] ) |

**Slika 4.2:** Prikaz rješenja gramatičke evolucije

- Broj u genotipu označava indeks produkcijskog pravila za najlijeviji nezavršni simbol
- Dobiveni izraz se pretvori u prefiksni sustav oznaka uz pomoć *shunting-yard* algoritma
- Moguće dobiti nevaljano rješenje

# Neuronske mreže






| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.91 | 0.12 | -0.78 | 0.62 | -0.13 | 0.24 | 0.39 | -0.95 | 0.22 | 0.47 | -0.56 |

**Slika 5.4:** Prikaz rješenja neuronske mreže

# Neuronske mreže



- Operatori križanja:
  - Križanje slojeva
  - Križanje neurona
  - Uniformno križanje neurona
- Operatori mutacije:
  - Skaliranje težine
  - Skaliranje neurona
  - Mutiranje neurona
  - Mutiranje N neurona
  - Gaussovo mutiranje težine

# Rezultati i analiza



| Metoda | Cmax | Ft | Uwt | Twt |
|---|---|---|---|---|
| min-min | 38.31 | 157.20 | 7.14 | 16.72 |
| max-min | 38.84 | 195.89 | 8.14 | 22.07 |
| min-max | 38.07 | 167.30 | 7.79 | 17.49 |
| sufferage | 37.93 | 160.97 | 7.20 | 16.65 |
| GP | 37.95,38.26,38.87 | 153.64,155.31,158.84 | **6.33**,7.06,7.84 | **12.96**,**13.60**,**14.62** |
| GEP | 37.95,**38.22**,38.73 | **153.53**,**154.83**,158.09 | 6.44,6.93,7.55 | 13.06,13.69,15.14 |
| CGP, k-turn. sel. | 37.88,38.27,**38.71** | 153.72,154.93,158.62 | 6.61,7.23,7.67 | 13.38,13.81,15.42 |
| CGP, evol. strat. | 37.90,38.39,38.71 | 153.75,155.06,**156.48** | 6.77,7.19,**7.42** | 13.44,14.09,15.24 |
| GE | **37.86**,38.36,40.88 | 154.78,159.43,171.53 | 6.65,7.35,7.93 | 13.41,14.37,19.99 |
| NN | 40.41,41.25,42.44 | 171.23,172.56,177.13 | 6.62,**6.86**,7.51 | 13.57,14.29,17.7 |

**Tablica 6.13:** Pregled rezultata svih metoda. Svaka ćelija sadrži minimum,medijan i maksimum dobrote.