

# Automatski razvoj pravila raspoređivanja za probleme raspoređivanja s ograničenjima

DIPLOMSKI RAD br. 1967

Kristijan Jaklinović

mentor: prof. dr. sc. Domagoj Jakobović

voditelj: dr. sc. Marko Đurasević

Sveučilište u Zagrebu

Fakultet Elektrotehnike i Računarstva

12.07.2019.

# Sadržaj

## Problem raspoređivanja

- O problemu

- Raspoređivanje u okruženju nesrodnih strojeva

- Klasifikacija problema raspoređivanja

## ATC

### Genetsko programiranje

- Primjena GP-a na problem raspoređivanja

- Skup terminala

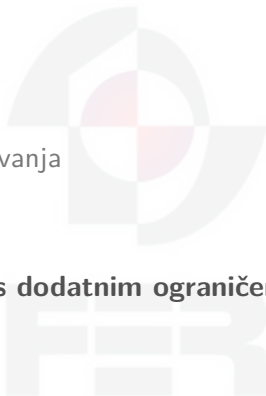
- Shema za izradu rasporeda

### Rješavanje problema raspoređivanja s dodatnim ograničenjima

- Pojedinačna ograničenja

- Kombinacije ograničenja

## Zaključak



# Problem raspoređivanja

## O problemu

Problem u kojem je **skup poslova** potrebno rasporediti na **skup resursa** na način da **minimizira** neki kriterij.

- ▶ NP-težak
- ▶ aproksimativne metode
- ▶ genetsko programiranje

# Motivacija

- ▶ nedostatak metaheurističkih postupaka
  - ▶ nisu primjenjivi kada je potrebno brzo izgraditi raspored i kada nisu dostupne sve informacije o problemu
- ▶ pravila raspoređivanja
  - ▶ automatizirano genetskim programiranjem

# Problem raspoređivanja

## Raspoređivanje u okruženju nesrodnih strojeva

Potrebno je za svaki od  $n$  **poslova** odrediti jedan od  $m$  **strojeva** na kojem će se posao izvoditi kao i redoslijed izvođenja poslova na svakom stroju.

### Okruženje nesrodnih strojeva

vrijeme obavljanja posla ovisi o stroju na kojem se taj posao izvršava.

- ▶ svojstva poslova ( $p_{ij}$ ,  $r_j$ ,  $d_j$ ,  $w_j$ )
- ▶ kriterij pojedinačnog posla -  $T_j$ 
  - ▶ vrijeme izvršavanja posla  $j$  nakon vremenskog ograničenja  $d_j$
- ▶ kriterij testiranja -  $Twt$ 
  - ▶  $Twt = \sum_{j=1}^n w_j T_j$

# Problem raspoređivanja

## Klasifikacija problema raspoređivanja

- ▶ Klasifikacija
  - ▶ S obzirom na dostupnost informacija
    - ▶ dostupne od početka
    - ▶ **kada posao uđe u sustav**
  - ▶ Trenutak konstruiranja rasporeda
    - ▶ statičko
    - ▶ **dinamičko**
  
- ▶ Raspored se gradi **paralelno s radom sustava.**

# Problem raspoređivanja

## Ograničenja

### Ograničenje

- ▶ vremenom postavljanja
- ▶ prikladnim strojevima
- ▶ privremenim prekidom rada strojeva
- ▶ redosljedom izvođenja poslova

# ATC

## *Apparent Tardiness Cost*

- ▶ heuristika, gradi raspored za vrijeme izvođenja sustava
- ▶ *Vepsalainen i Morton (1987. godine)* te *Ow i Morton (1989.)*
- ▶ koristi prioritetsnu funkciju za računanje prioriteta posla  $j$  na stroju  $i$  u trenutku  $t$

$$l_{ij}(t) = \frac{w_j}{p_{ij}} \exp \left[ - \frac{\max(d_j - p_{ij} - t, 0)}{k\bar{p}} \right]$$



# ATC shema za izradu rasporeda

---

```
1:  $raspored[m][n] = []$ 
2: while broj neraspoređenih poslova  $\neq 0$  do
3:    $t \leftarrow$  trenutno vrijeme
4:    $neraspoređeniPoslovi \leftarrow$  dostupni neraspoređeni poslovi u sustavu
5:   while  $neraspoređeniPoslovi$  nije prazan do
6:     // pronadi posao s najboljim prioritetom
7:     for all  $j \in neraspoređeniPoslovi, i \in m$  do
8:        $prioritet \leftarrow I_{ij}(t)$ 
9:     end for
10:     $posao \leftarrow$  posao s najvećim prioritetom
11:     $stroj \leftarrow$  stroj na kojem će se  $posao$  najranije izvršiti
12:     $raspored[stroj] \leftarrow$  ako je stroj dostupan  $posao$ 
13:  end while
14: end while
15:  $fitnes \leftarrow$  evaluiraj  $raspored$ 
```

---

# Prednosti i mane ATC-a

## Prednosti

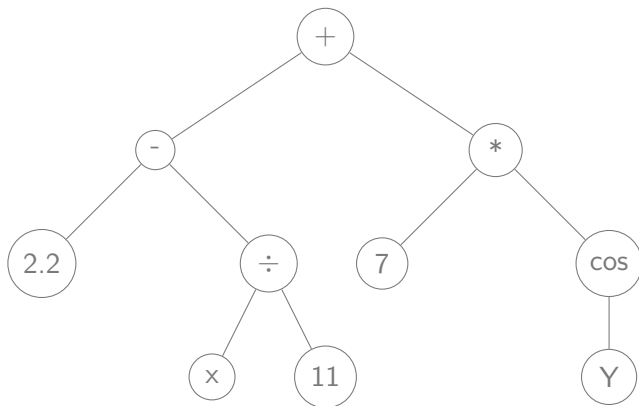
- ▶ brzina izvođenja
- ▶ primjenjiv u dinamičkim uvjetima

## Nedostaci

- ▶ teško ih je ručno izraditi
- ▶ suboptimalni rezultati
- ▶ prikladna samo za kriterije kašnjenja

# Genetsko programiranje

Automatiziran optimizacijski postupak razvoja računalnih programa ili funkcija, čija je namjena rješavanje većinom složenih problema iz područja računarstva.



$$(2.2 - \frac{x}{11}) + (7 + \cos y)$$

# Genetsko programiranje

## Primjena GP-a na problem raspoređivanja

1. definirati pravilo raspoređivanja
  - ▶ prioriteta funkcija - izračun prioriteta posla (automatizirano)
  - ▶ shema za izradu rasporeda - donošenje odluka (ručno)
2. definirati skup operatora
3. definirati skup terminala

OZNAKA	OPIS
+	operator binarnog zbrajanja
-	operator binarnog oduzimanja
*	operator binarnog množenja
÷	operator binarnog dijeljenja
<i>pos</i>	unarni operator, $pos(x) = \max(x, 0)$

**Tablica:** Skup operatora

# Skup terminala

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

# Shema za izradu rasporeda

---

- 1: **while** postoje neraspoređeni poslovi **do**
  - 2:     čekaj dok posao ne postane dostupan ili se ne završi
  - 3:     **for all** dostupne poslove i strojeve **do**
  - 4:         izračunaj prioritet  $\pi_{ij}$  posla  $j$  na stroju  $i$
  - 5:     **end for**
  - 6:     **for all** dostupne poslove **do**
  - 7:         odredi najbolji stroj (onaj za koji se postiže najbolji iznos prioriteta  $\pi_{ij}$ )
  - 8:     **end for**
  - 9:     **while** postoje poslovi čiji je najbolji stroj dostupan **do**
  - 10:         odredi najbolji prioritet takvih poslova
  - 11:         rasporedi posao s najboljim prioritetom
  - 12:     **end while**
  - 13: **end while**
-

# Automatska izrada pravila raspoređivanja GP-om

- ▶ rješavanje problema raspoređivanja GP-om već je dobro istražen u literaturi
- ▶ ali ne s dodatnim ograničenjima
- ▶ na problemima bez ograničenja GP postiže bolje rezultate od ručno izrađenih pravila raspoređivanja

Rješavanje problema  
raspoređivanja s dodatnim  
ograničenjima



# Postavke eksperimenata

- ▶ skup za testiranje i učenje
  - ▶ 60 instanci problema
- ▶  $Twt, Twt_{uk} = \sum_{i=0}^{i<60} Twt_i$
- ▶ 30 pokretanja

PARAMETAR	VRIJEDNOST
selekcija	turnirska eliminacijska
veličina turnira	3
veličina populacije	1000
vjerojatnost mutacije	0.3
kriterij zaustavljanja (broj generacija)	80
maksimalna dubina stabla	5
minimalna dubina stabla	1

**Tablica:** Parametri algoritma

# Vrijeme postavljanja

## OVP

Označava vrijeme koje je potrebno stroju za prilagodbu izvođenja sljedećeg posla.

## Prilagodba ATC-a → ATCS

$$l_{ij}(t, l) = \frac{w_j}{p_{ij}} \exp \left[ - \frac{\max(d_j - p_{ij} - t, 0)}{k_1 \bar{p}} \right] \exp \left[ \frac{s_{lj}}{k_2 \bar{s}} \right]$$

OZNAKA	OPIS
$l$	indeks posljednje obavljenog posla na stroju
$s_{lj}$	vrijeme postavljanja posla $j$ nakon posla $l$
$\bar{s}$	srednja vrijednost vremena postavljanja
$k_2$	parametar

# Prilagodba GP-a

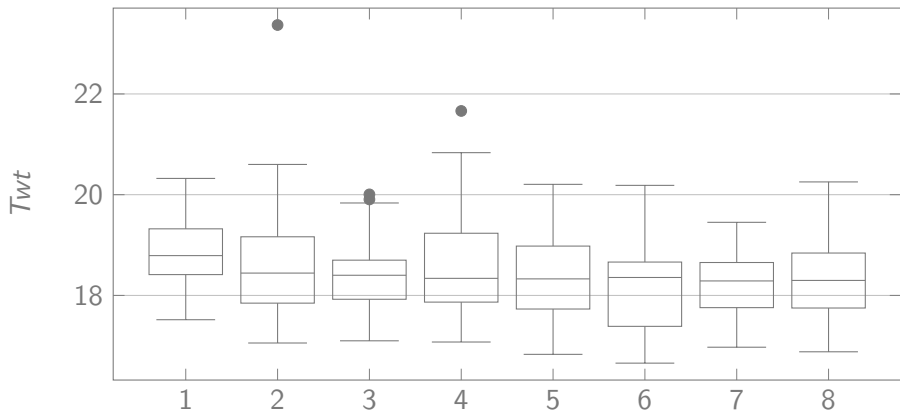
## Vrijeme postavljanja

OZNAKA	OPIS
<i>smin</i>	minimalna vrijednost vremena postavljanja
<i>sAvg</i>	srednja vrijednost vremena postavljanja
<i>setMac</i>	vrijeme postavljanja posla na stroju

**Tablica:** Dodatni terminali

- ▶ shemu nije potrebno prilagođavati

# Rezultati



1 bez dodatnih terminala 2 *smin* 3 *sAvg* 4 *setMac* 5 *smin* i *sAvg* 6 *smin* i *setMac* 7 *sAvg* i *setMac* 8 *smin* i *sAvg* i *setMac*

► ATCS:  $k_1 = 0.95$ ,  $k_2 = 0.1$ ,  $T_{wt} = 21.8318$

# Ograničenje prikladnih strojeva

OPS

Za svaki stroj definira poslove koji se na njemu mogu izvoditi.

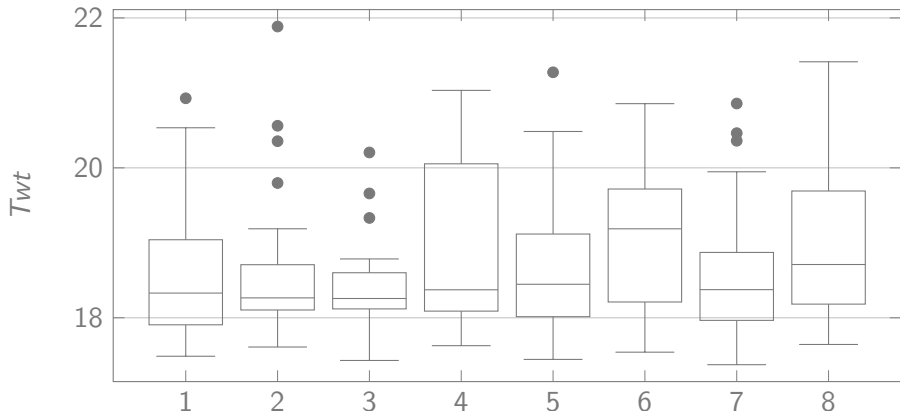
## Prilagodba ATC-a

- ▶ preskačemo računanje prioriteta za kombinacije neprikladnih parova strojeva i poslova

## Prilagodba GP-a

- ▶ prilagođena shema za izradu rasporeda
- ▶ dodatni terminali
  - ▶ *emfj* - broj strojeva na kojima se posao može izvršiti
  - ▶ *rjfm* - broj dostupnih poslova koji se mogu izvršiti na stroju
  - ▶ *amfj* - broj dostupnih strojeva koji mogu izvršiti posao

# Rezultati



**1** bez dodatnih terminala, **2** *amjf*, **3** *emfj*, **4** *rjfm*, **5** *amfj* i *emfj*, **6** *amfj* i *rjfm*, **7** *emfj* i *rjfm*, **8** *amfj* i *emfj* i *rjfm*

► ATC:  $k = 0.45$ ,  $T_{wt} = 20.3027$

# Privremeni prekid u radu strojeva

## OPP

Definira vremenska razdoblja u kojima je stroj u nemogućnosti izvoditi poslove.

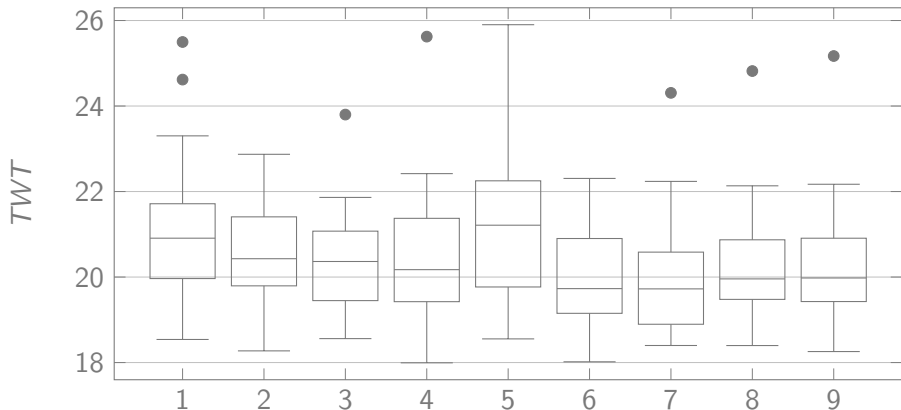
### Prilagodba ATC-a

- ▶ preskačemo one kombinacije poslova i strojeva za koje će se dogoditi prekid za vrijeme izvođenja posla

### Prilagodba GP-a

- ▶ prilagođena shema za izradu rasporeda
- ▶ dodatni terminali
  - ▶ *bwhde* - hoće li se dogoditi prekid tijekom izvođenja posla
  - ▶ *norb* - broj preostalih prekida
  - ▶ *tube* - vrijeme do završetka prekida
  - ▶ *tunb* - vrijeme do sljedećeg prekida

# Rezultati



1 bez dodatnih terminala, 2 *bwhde*, 3 *norb*, 4 *tube*, 5 *tunb*, 6 *bwhde* i *tube*, 7 *bwhde* i *tunb*, 8 *bwhde* i *norb* i *tube*, 9 *bwhde* i *norb* i *tube* i *tunb*

► ATC:  $k_1 = 1.5$ ,  $Twt = 22.525$



# Ograničenje redosljedom izvođenja

## ORI

Za svaki posao definira poslove koji se moraju izvršiti prije njega, tj. njegove prethodnike.

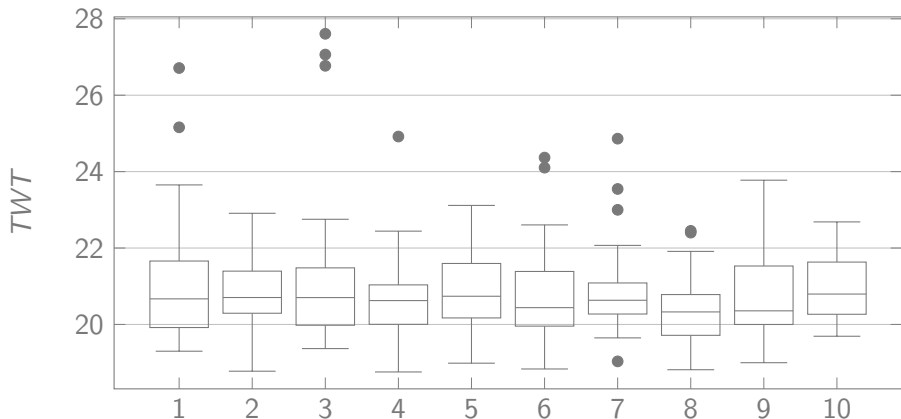
## Prilagodba ATC-a

- ▶ preskačemo one poslove čiji prethodnici nisu raspoređeni

## Prilagodba GP-a

- ▶ prilagođena shema za izradu rasporeda
- ▶ dodatni terminali
  - ▶ *nous* - broj neraspoređenih sljedbenika trenutno razmatranog posla
  - ▶ *avss* - prosječni *slack* svih sljedbenika trenutno razmatranog posla,  $slack = d_j - (t + p_{ij})$
  - ▶ *mss* - minimalan *slack* svih sljedbenika trenutno razmatranog posla
  - ▶ *mxss* - maksimalan *slack* svih sljedbenika trenutno razmatranog posla
  - ▶ *wsl* - težinsko kašnjenje svih sljedbenika trenutno razmatranog posla
  - ▶ *scl* - broj svih čvorova stabla sljedbenika za dani posao
  - ▶ *mscl* - najveća dubina stabla sljedbenika za dani posao

# Rezultati



**1** bez dodanih terminala, **2** *nous*, **3** *avss*, **4** *mss*, **5** *wsl*, **6** *avss* i *mss*, **7** *avss* i *nous*, **8** *mss* i *nous*, **9** *ass* i *nous* i *wsl*, **10** *mss* i *ass* i *wsl* i *nous*

► ATC:  $k_1 = 1.3$ ,  $T_{wt} = 20.1955$

# Kombinacije ograničenja

	OVP OPS	OVP OPP	OPS OPP	OPP ORI
ATC				
<i>Twt</i>	27.6076	30.0478	40.9037	39.717
GP				
<i>min</i>	24.2597	23.2614	28.8009	26.9227
<i>median</i>	27.58485	<b>25.26985</b>	<b>33.046</b>	<b>31.2314</b>
<i>max</i>	46.5309	29.5905	39.0432	36.9033
GP - bez dodatnih terminala				
<i>min</i>	23.8205	23.761	30.5045	29.8797
<i>median</i>	<b>26.7524</b>	26.06835	34.09265	33.5573
<i>max</i>	30.9706	30.6188	85.9731	42.7583

# Kombinacije ograničenja

	OVP OPS OPP	OVP OPP ORI	OVP OPS ORI
ATC			
<i>Twt</i>	<b>37.8632</b>	55.954	56.9899
GP			
<i>min</i>	35.9829	34.3962	46.7345
<i>median</i>	41.96735	<b>42.8429</b>	51.5876
<i>max</i>	52.4947	47.7727	58.6026
GP - bez dodatnih terminala			
<i>min</i>	37.9792	39.796	46.3836
<i>median</i>	42.3346	43.8096	<b>51.47405</b>
<i>max</i>	47.8872	52.1177	58.5926

# Kombinacije ograničenja

	OVP OPS OPP ORI
ATC	
<i>Twt</i>	108.335
GP	
<i>min</i>	70.4215
<i>median</i>	81.98605
<i>max</i>	91.1927
GP - bez dodatnih terminala	
<i>min</i>	76.6059
<i>median</i>	<b>80.7225</b>
<i>max</i>	106.617

# Zaključak

## Što je napravljeno

- ▶ istražen pristup rješavanju problema genetskim programiranjem
- ▶ prilagođen GP problemu kao i dodatnim ograničenjima
- ▶ implementirana ATC heuristika i prilagođena ograničenjima
- ▶ genetsko programiranje uspoređeno s ATC heuristikom

## Zaključak

- ▶ GP generalno bolji od ATC-a
- ▶ potrebna optimizacija terminala

PITANJA ?

# Primjer jedinke

```
<Tree size="36">+ + / + pmin PAT w - - w - MR dd - * w age / pmin  
MR + - + pmin + SL pmin pt pos - / dd MR - MR dd </Tree>
```

$$\frac{pmin + pat}{w} + (w - (MR - dd) - (w * age - \frac{pmin}{MR}))$$
$$+(pmin + SL + pmin - pt) + pos(\frac{dd}{MR} - (MR - dd))$$