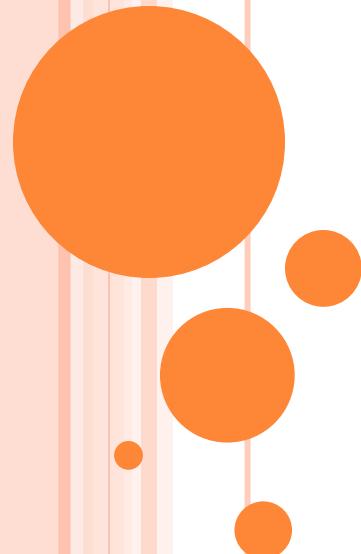


# **PRIMENA METAHEURISTIKE MRAVLJIH KOLONIJA ZA REŠAVANJE JEDNODIMENZIONOG PROBLEMA PAKOVANJA**



**MPIO 2016/2017.**

## JEDNODIMENZIONI PROBLEM PAKOVANJA (BIN PACKING PROBLEM - BPP)

- Problem pakovanja pripada klasi problema sečenja i pakovanja (*cutting and packing problems-CPP*). Opštu strukturu ove grupe problema opisao je Dyckhoff, (1990)
- Zadatak CPP: Raspolažemo kombinacijom grupe malih (paketa) i grupe velikih objekata (skladišta). Cilj je spakovati male objekte po obrascu i dodeliti ih velikim objektima.
- Primeri CPP: problem ranca (*knapsack problem*) problem utovara prtljaga (*vehicle loading problem*), problem raspoređivanja zadataka na procesore (*multiprocessor scheduling problem*), pa čak i problem raspoređivanja kapitala (*multi-period capital budgeting problem*)



## JEDNODIMENZIONI PROBLEM PAKOVANJA (BIN PACKING PROBLEM - BPP)

- BPP spada u jednostavnije probleme iz klase CPP
- Zadatak BPP: Dat je skup paketa sa pridruženim veličinama koje treba spakovati u skladišta unapred zadatog kapaciteta, tako da je broj iskorišćenih skladišta minimalan.
- BPP pripada klasi NP teških problema (Garey i Johnson, 1979).
- Egzaktne metode primenjene na BPP efikasno rešavaju instance problema manjih dimenzija, dok se za instance većih dimenzija koriste heuristike i metaheuristike.
- Najbolje rezultate daju hibridne metaheurističke metode



# PRIMENE JEDNODIMENZIONOG PROBLEMA PAKOVANJA

- **Problem utovara prtljaga** – minimizujemo broj vozila koja prevoze pakete određenih veličina do zadatih destinacija; ovde minimizujemo i troškove transporta tih vozila.
- **Problem pakovanja na traci** - pakovanje 2D paketa pravougaonog oblika na traci fikne širine u cilju minimizacije iskorišćene trake. Pravougaonici se ne smeju preklapati, a moguće je rotirati pravougaonike za 90 stepeni.
- **Primene u industriji** – papirna, čelična, drvna industrija. Sečemo materijale na određene veličine, a cilj je minimizovati materijal koji otpada (ovde je reč o 2D ili 3D BPP-u).
- **Neki problemi koji su slični BPP-u**, npr. VLSI dizajn (Very Large Scale Integration), tj. kreiranje integrisanih kola kombinovanjem velikog broja tranzistora u jedan čip (VLSI uređaj), planiranje procesorskog vremena, problemi iz ekonomije, npr. upravljanje kapitalom...

# DEFINICIJA BPP

Dati su:

- konačan skup paketa  $L = \{1, \dots, n\}$
- veličine paketa  $I_j, j = 1, \dots, n$
- konstanta  $C$  (kapacitet svakog skladišta)
- broj raspoloživih skladišta  $N$

**Cilj:** spakovati pakete u što manji broj otvorenih skladišta  
bez prekoračenja kapaciteta skladišta.



# MATEMATIČKA FORMULACIJA BPP

Matematička formulacija problema koristi binarne promenljive:

$$x_{ij} = \begin{cases} 1, & \text{ako je paket } j \text{ u skladištu } i \\ 0, & \text{inače} \end{cases} \quad y_i = \begin{cases} 1, & \text{ako je skladište } i \text{ upotrebljeno} \\ 0, & \text{inače} \end{cases}$$

Minimizujemo funkciju  $z = \sum_{i=1}^N y_i$

Pri uslovima:  $\sum_{j=1}^n l_j x_{ij} \leq C y_i, \quad \forall i = 1, 2, \dots, N$

$$\sum_{i=1}^N x_{ij} = 1, \quad \forall j = 1, 2, \dots, n$$

$$x_{ij} \in \{0, 1\}, \forall i, j$$

$$y_i \in \{0, 1\}, \quad \forall i$$



# DOSADAŠNJE METODE REŠAVANJE PROBLEMA

- **Greedy (pohlepne) heuristike – FFD, BFD, NFD, WFD, (Dyckhoff , 1990)**  
Zajednička ideja: paketi se ređaju u nerastućem poretku (po veličini), a koriste različite strategije

**FFD** (First Feasible Descreasing)– paket se smešta u prvo skladište u koje može stati.

**BFD** (Best Feasible Descreasing)– paket se smešta u prvo skladište u koje može stati, a koje ima najmanje preostalog slobodnog prostora.

**NFD** (Next Feasible Descreasing)– paketi se smeštaju u otvoreno skladište dok god mogu u njega stati.

**WFD** (Worst Feasible Descreasing) – paket se smešta u prvo skladište u koje može stati, a koje ima najviše preostalog slobodnog prostora.

Heuristike se zaustavljaju kada su svi paketi raspoređeni.

# DOSADAŠNJE METODE REŠAVANJE PROBLEMA

- **Metaheuristike:**

Hibridni genetski algoritam grupisanja (Hybrid Grouping Genetic Algorithm - HGGA), Falkenauer (1996)

Heuristika redukcije (Reduction Procedure - MTP), Martello and Toth, (1999)

Optimizacija pomoću mravljih kolonija (Ant Colony Optimization - ACO), Levine i Ducatelle (2004).
- **Predlog nove metode:**Kombinacija metoda ACO i heuristike lokalnog pretraživanja (Local Search -LS), (Rajačić, 2013), po uzoru na Levine i Ducatelle (2004).



# HEURISTIKA FFD

(PAKET SE SMEŠTA U PRVO SKLADIŠTE U KOJE MOŽE STATI)

---

## *Algoritam 2.1: Pseudo kod – FFD*

---

*Poređaj pakete nerastuće;*

**Ponavljam:**

**For svaki paket do**

**For svako skladište do**

**If paket može stati u skladište**

**Spakuj paket;**

**Pređi na sledeći paket;**

**End if**

**End for**

**If paket ne može stati u skladište**

**Otvori novo skladište;**

**Spakuj paket u njega;**

**End if**

**End for**

**Sve dok: Nisu svi paketi spakovani;**

**Izlaz: Paketi spakovani u skladišta.**

---



# HEURISTIKA BFD

(PAKET SE SMEŠTA U PRVO SKLADIŠTE U KOJE MOŽE STATI, A KOJE IMA NAJMANJE PREOSTALOG SLOBODNOG PROSTORA)

---

#### *Algoritam 2.4: Pseudo kod – WFD*

---

*Poređaj pakete nerastuće;*

**Ponavljam:**

**For svaki paket do**

**For svako skladište do**

*Izračunaj preostali prostor u skladištu;*

**If** skladište ima min slobodnog prostora  
i paket može stati u njega

*Stavi paket;*

*Izračunaj preostali slobodan prostor u skladištu;*

**End if**

**End for**

**If** paket ne može stati u skladište

*Otvori novo skladište;*

*Stavi paket u njega:*

**End if**

**End for**

**Sve dok:** Nisu svi paketi raspoređeni;

**Izlaz:** Paketi raspoređeni u skladišta.

---



# HEURISTIKA NFD

(PAKETI SE SMEŠTAJU U OTVORENO SKLADIŠTE DOK GOD MOGU U NJEGA STATI)

---

## *Algoritam 2.3: Pseudo kod – NFD*

---

*Poređaj pakete nerastuće;*

*Ponavljam:*

*For svaki paket do*

*If paket može stati u trenutno otvoreno skladište*

*Stavi paket;*

*Else*

*Otvori novo skladište i smesti paket u njega;*

*Zatvori prethodno skladište;*

*End for*

*Sve dok: Nisu svi paketi raspoređeni;*

*Izlaz: Paketi raspoređeni u skladišta.*

---



# HEURISTIKA WFD

(PAKET SE SMEŠTA U PRVO SKLADIŠTE U KOJE MOŽE STATI, A KOJE IMA NAJVIŠE PREOSTALOG SLOBODNOG PROSTORA)

---

## Algoritam 2.4: Pseudo kod – WFD

---

Poređaj pakete nerastuće;

Ponavljam:

    For svaki paket do

        For svako skladište do

            Izračunaj preostali prostor u skladištu;

            If skladište ima max slobodnog prostora  
                i paket može stati u njega

                Stavi paket;

                Izračunaj preostali slobodan prostor u skladištu;

            End if

        End for

        If paket ne može stati u skladište

            Otvori novo skladište;

            Stavi paket u njega:

        End if

    End for

Sve dok: Nisu svi paketi raspoređeni;

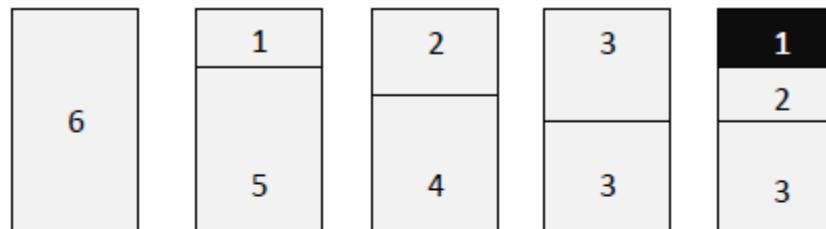
Izlaz: Paketi raspoređeni u skladišta.



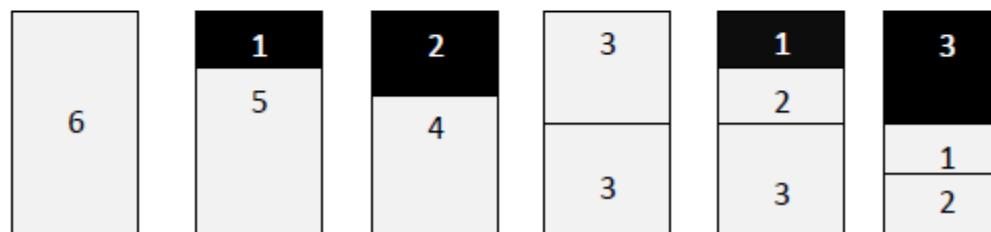
# PRIMER KORIŠĆENJA HEURISTIKA

Neka je dat niz paketa  $L = \{4,1,2,5,3,2,3,6,3\}$  i neka su skladišta kapaciteta 6. Paketi se redaju u nerastućem poretku  $L = \{6,5,4,3,3,3,2,2,1\}$ . Nakon toga, različite heuristike smeštaju pakete na različite načine. Od značaja je napomenuti da je teorijski optimum 5.

FFD:



NFD:



Metode BFD i WFD su za uzeti primer imale isti rezultat kao FFD.



## PRILAGOĐAVANJE ACO ALGORITMA BPP-U

Specifičnost BPP je da generalno imamo više paketa istih veličina

Veličina paketa je najvažnije svojstvo, tako da nije bitan redni broj paketa već veličina

Svaki paket možemo identifikovati sa njegovom veličinom: „**paket i**“ znači „**paket veličine i**“ . Ako veličina paketa nije ceo broj?

- **Definicija traga feromona za BPP:**
- Koristi se sledeći pristup za definiciju matrice feromona:  
 $\tau(i,j)$  je pristrasnost (poželjnost) da se paket *i* i paket *j* nađu u istom skladištu.
- **Prednosti:**
  - Matrica feromona je kompaktna.
  - Matrica predstavlja šablon za pakovanje, jer čuva informacije o “dobrim” pakovanjima, pojačavajući veze između veličina.
  - Kada se pronađe dobar šablon, moći će da se koristi više puta tokom rešavanja problema.

## Pojačavanje feromona i funkcija prilagođenosti

- Ideja je da se feromon pojačava svaki put kada se paketi  $i$  i  $j$  nađu zajedno u istom skladištu najboljem rešenju  $s_{best}$

$$\tau(i,j) = \rho \times \tau(i,j) + m \times f(s_{best})$$

- $m$  = broj koji označava koliko puta su se paketi  $i$  i  $j$  našli u istom skladištu u najboljem rešenju
- Funkcija prilagođenosti je pokazatelj kvaliteta rešenja:

$$f(s) = \frac{\sum_{i=1}^N (F_i/C)^z}{N}$$

- $F_i$  je popunjenoć skladišta  $i$
- Parametar  $z$  potpomaže raznovrsnost rešenja (optimalno  $z=2$ ).



# Pojačavanje feromona i funkcija prilagođenosti

*Da li ostavljanje feromona dozvoliti samo najboljem mravu?*

- Korišćenje samo najboljeg mrava za pojačavanje feromona čini pretragu previše usmerenom. Ako se primenjuje ova strategija, biće podržane samo kombinacije skladišta koje se često javljaju u dobrom rešenjima.
- Postoji nekoliko načina da se balansira između diversifikacije i usmeravanja pretraživanja. Jedan od načina je odabir između najboljeg mrava u tekućoj iteraciji  $s_{Lb}$  i globalno najboljeg mrava  $s_{Gb}$ .
- Najpogodnije je uvesti parametar  $\gamma$  koji nagoveštava koliko koristimo  $s_{Lb}$  dok ne upotrebimo  $s_{Gb}$  ponovo.



# IZGRADNJA REŠENJA

- Svaki mrav počinje sa praznim skladištem i skupom paketa koji treba smestiti.
- Mrav konstruiše rešenje tako što puni sva skladišta dok svi paketi nisu spakovani.
- U slučaju da nijedan od preostalih paketa ne može stati u trenutno otvoreno skladište, ono se zatvara i otvara se novo.

## Koji paket smestiti u otvoreno skladište?

Odluka se zasniva na informaciji iz heuristike, ali i na informaciji iz feromona.

Verovatnoća da će u mrav  $k$  izabrati da paket  $j$  smesti u otvoreno skladište  $b$  nakon parcijalnog rešenja  $s$  je definisana sa:

$$p_k(s, b, j) = \begin{cases} \frac{[\tau_b(j)] \times [\mu(j)]^\beta}{\sum_{g \in J_k(s, b)} [\tau_b(g)] \times [\mu(g)]^\beta}, & \text{ako } j \in J_k(s, b) \\ 0 & \text{inače} \end{cases}$$

$J_k(s, b)$  je skup paketa koji mogu stati u trenutno otvoreno skladište  $b$ . To su paketi koji su ostali nakon kreiranja parcijalnog rešenja  $s$ , a koji se mogu smestiti u  $b$

$\mu(j)$  je vrednost heuristike za paket  $j$ ,

$\tau_b(j)$  je vrednost feromona za paket  $j$  koji se nalazi u skladištu  $b$

# IZGRADNJA REŠENJA

## Kako izračunati vrednost feromona?

Vrednost feromona za paket  $j$  koji se nalazi u skladištu  $b$  i predstavlja sumu svih vrednosti feromona između paketa  $i$  i  $j$ , a koji se već nalaze u skladištu  $b$ , podeljena sumom veličina svih paketa u skladištu  $b$ .

$$\tau_b(j) = \begin{cases} \frac{\sum_{i \in b} \tau(i, j)}{|b|}, & \text{ako je } b \neq \emptyset \\ 1, & \text{inače} \end{cases}$$

## Kako izračunati vrednost heuristike?

Heuristika se vodi idejom da ćemo uvek izabrati veliki paket za smeštanje pre nego mali. Dakle, heuristički odabir paketa je u direktnoj vezi sa njegovom veličinom.

Uzimamo  $\mu(j) = j$

## LOKALNO PRETRAŽIVANJE

- U mnogim aplikacijama za rešavanje NP teških problema, ACO je kombinovan sa LS
- Local search vrši pretragu u okolini nekog inicijalnog rešenja, koje dobijamo pomoću ACO
- **Ideja:** Prazni se *loc* najmanje punih skladišta čime je određen broj paketa u inicijalnom rešenju je oslobođen. Pokušava se zamena  
**2 smeštena sa 2 slobodna paketa ili**  
**2 smeštena sa 1 slobodnim paketom ili**  
**1 smešten sa 1 slobodnim paketom.**
- Uslov: skladišta koja nisu oslobođena moraju biti više popunjena nego pre zamene, da bi se zamena prihvatile.
- Ostali slobodni paketi koji nisu nigde dodati se ubacuju pomoću neke od postojećih heuristika NFD, FFD, BFD, WFD.
- Procedura se ponavlja dok se ne popravi funkcija prilagođenosti (ako je moguće).
- Pomoću ovako dobijenih rešenja (lokalnih optimuma) pojačavamo trag feromona.

## PRIMER LOKALNOG PRETRAŽIVANJA

Neka je kapacitet skladišta jednak 10 i neka je izabrana heuristika kojom će preostali paketi da se smeštaju FFD. Parametar *loc* u primeru ima vrednost 2.

- *Rešenje pre primene lokalnog pretraživanja*

| 6 3 | 7 2 | 4 3 | 5 2 | 3 5 | 4 4 |

U svakom skladištu (između svaka dva znaka |), smešteni su paketi određenih veličina. U prvom skladištu se nalaze paketi veličina 6 i 3. Kapacitet skladišta je 10, broj iskorišćenih skladišta je 6.

Gubitak u skladištu definišemo kao preostali prostor nakon smeštanja paketa u njega do popunjavanja kapaciteta skladišta. Tako u prvom skladištu imamo gubitak 1, a ukupan gubitak je jednak 12.

- *Otvaramo dva najmanje puna skladišta (to su skladišta 3 i 4) i oslobađamo pakete:*

Preostala skladišta: | 6 3 | 7 2 | 3 5 | 4 4 |

Slobodni paketi:      5, 4, 3, 2



# PRIMER LOKALNOG PRETRAŽIVANJA

- *Vršimo zamenu:*

Prvo skladište:  $| \underline{6} \ 3 | \longrightarrow | \underline{6} \ 4 |$  Ostalo {5, 3, 3, 2}.

Drugo skladište:  $| \underline{7} \ 2 | \longrightarrow | \underline{7} \ 3 |$  Ostalo {5, 3, 2, 2}.

Treće skladište:  $| \underline{3} \ 5 | \longrightarrow | \underline{5} \ 5 |$  Ostalo {3, 3, 2, 2}.

Četvrto skladište:  $| \underline{4} \ 4 | \longrightarrow | \underline{4} \ 4 |$  Ostalo {3, 3, 2, 2}.

- *Smeštamo ostale pakete pomoću FFD algoritma:*

Četvrto skladište:  $| \underline{4} \ 4 | \longrightarrow | \underline{4} \ 4 \ 2 |$  Ostalo {3, 3, 2}.

Novo skladište: {3, 3, 2}  $\longrightarrow | \underline{3} \ 3 \ 2 |$  Ostalo Ø.

- *Konačno rešenje:*

$| \underline{6} \ 4 | \ 7 \ 3 | \ 5 \ 5 | \ 4 \ 4 \ 2 | \ 3 \ 3 \ 2 |$

Broj iskorišćenih skladišta je 5, a gubitak 2.

- *Ponavljanje procedure:*

Nema mogućeg napretka; stajemo.



## EKSPERIMENTALNI REZULTATI

Testiranja su izvršena na standardnim ORLIB instancama .

80 test instanci, označene sa uN\_M

Kapacitet skladišta je 150

N paketa sa veličinama koje su slučajno izabrane u intervalu 20-100

N=120, 250, 500, 1000

Za svako N generisano je M instanci

# EKSPERIMENTALNI REZULTATI – PODEŠAVANJE PARAMETARA

Parametar

*br\_mrava:*

<i>br_mrava</i>	<i>u120</i>	<i>u250</i>	<i>u500</i>	<i>u1000</i>
10	0.057	0.083	0.458	1.285
15	0.068	0.099	0.567	1.659
20	0.078	0.161	1.103	1.457
25	0.083	0.140	0.733	1.862

Tabela 4.1: Vreme izvršavanja programa u zavisnosti od parametra *br\_mrava* dano u sekundama.

Parametar  $\theta$ :

$\theta$	<i>u120</i>	<i>u250</i>	<i>u500</i>	<i>u1000</i>
1	0.077	0.092	0.606	1.516
2	0.065	0.088	1.351	4.329
5	0.083	0.158	12.749	25.574
10	0.104	0.234	38.013	473.703

Tabela 4.2 : Vreme izvršavanja programa u zavisnosti od parametra  $\theta$  dano u sekundama.

Parametar  $\gamma$  :

$\gamma$	<i>u120</i>	<i>u250</i>	<i>u500</i>	<i>u1000</i>
1	0.057	0.083	0.442	1.171
2	0.062	0.094	0.687	1.784
3	0.063	0.099	0.504	1.301
5	0.058	0.099	0.614	1.175

Tabela 4.3 : Vreme izvršavanja programa u zavisnosti od parametra  $\gamma$  dano u sekundama.

# EKSPERIMENTALNI REZULTATI – PODEŠAVANJE PARAMETARA

Parametar  $\rho$ :

$\rho$	$u120$	$u250$	$u500$	$u1000$
0.10	0.068	0.109	0.474	1.275
0.25	0.063	0.104	0.613	1.124
0.50	0.062	0.110	0.401	1.177
0.75	0.063	0.112	0.453	1.197
0.95	0.062	0.097	0.518	0.986

Tabela 4.4: Vreme izvršavanja programa u zavisnosti od parametra  $\rho$  dano u sekundama.

Parametar  $loc$ :

$loc$	$u120$	$u250$	$u500$	$u1000$
2	0.066	0.103	1.050	4.076
3	0.063	0.110	0.586	1.873
4	0.068	0.100	0.482	1.142
5	0.071	0.172	0.626	1.294
10	0.083*	0.174	0.896	1.170

Tabela 4.5: Vreme izvršavanja programa u zavisnosti o parametru  $loc$  dano u sekundama.

Vrednosti ostalih parametara su optimalne.

- \* označava da nije dobiten teorijski optimum



# POREĐENJE HEURISTIKA

Iskorišćena skladišta:

	<i>Opt</i>	FFD	NFD	BFD	WFD
<i>u120</i>	<b>48</b>	<i>Opt + 1</i>	<i>Opt + 19</i>	<i>Opt + 1</i>	<i>Opt + 2</i>
<i>u250</i>	<b>99</b>	<i>Opt + 1</i>	<i>Opt + 39</i>	<i>Opt + 1</i>	<i>Opt + 2</i>
<i>u500</i>	<b>198</b>	<i>Opt + 2</i>	<i>Opt + 82</i>	<i>Opt + 2</i>	<i>Opt + 2</i>
<i>u1000</i>	<b>399</b>	<i>Opt + 4</i>	<i>Opt + 164</i>	<i>Opt + 4</i>	<i>Opt + 4</i>

Tabela 4.7 : Broj iskorišćenih skladišta u svakoj heuristici.

Poređenje brzina izvršavanja:

	<i>u120</i>	<i>u250</i>	<i>u500</i>	<i>u1000</i>
FFD	<b>0.531033</b>	<b>0.597954</b>	<b>0.786479</b>	<b>1.098585</b>
NFD	0.608417	0.674830	0.798720	1.142213
BFD	5.284068	5.578800	7.497821	14.670705
WFD	5.163053	5.368559	6.881902	12.545984

Tabela 4.6 : Rezultati upoređivanja vremena izvršavanja heuristika.  
Vreme dano u milisekundama.



# EFIKASNOST ALGORITMA U ZAVISNOSTI OD HEURISTIKE

	<i>Opt</i>	<i>ACO (FFD)</i>	<i>ACO (NFD)</i>	<i>ACO (BFD)</i>	<i>ACO (WFD)</i>
<b>u120</b>	<b>48</b>	<i>Opt</i>	<i>Opt</i>	<i>Opt</i>	<i>Opt</i>
<b>u250</b>	<b>99</b>	<i>Opt + 2</i>	<i>Opt + 2</i>	<i>Opt</i>	<i>Opt</i>
<b>u500</b>	<b>198</b>	<i>Opt</i>	<i>Opt + 3</i>	<i>Opt + 1</i>	<i>Opt + 1</i>
<b>u1000</b>	<b>399</b>	<i>Opt</i>	<i>Opt</i>	<i>Opt + 3</i>	<i>Opt + 2</i>

Tabela 4.9 : Broj skladišta ACO algoritma u zavisnosti od početne informacije.

	<i>u120</i>	<i>u250</i>	<i>u500</i>	<i>u1000</i>
ACO (FFD)	<b>49.949217</b>	81.303023	648.139368	2843.421072
ACO (NFD)	52.020834	<b>79.765569</b>	<b>491.532009</b>	<b>1089.491660</b>
ACO (BFD)	51.352208	104.329493	814.436775	1201.873814
ACO (WFD)	51.321411	98.105162	855.232495	1147.719506

Tabela 4.8 : Rezultati upoređivanja vremena izvršavanja ACO algoritma. Vreme, dato u milisekundama.

Poređenje brzina zvršavanja:

Rezultati poređenja : ACO (NFD) < ACO (FFD) < ACO (WFD) < ACO (BFD)

# POREĐENJE SA DRUGIM PRISTUPIMA

Iskorišćena skladišta:

	<i>u120</i>	<i>u250</i>	<i>u500</i>	<i>u1000</i>
ACO (FFD)	<i>Opt</i>	<i>Opt + 2</i>	<i>Opt</i>	<i>Opt</i>
ACO (NFD)	<i>Opt</i>	<i>Opt + 2</i>	<i>Opt + 3</i>	<i>Opt</i>
HGGA	<i>Opt + 2</i>	<i>Opt + 3</i>	<i>Opt</i>	<i>Opt</i>
MTP	<i>Opt + 2</i>	<i>Opt + 12</i>	<i>Opt + 44</i>	<i>Opt + 78</i>

Tabela 4.11 : Broj iskorišćenih skladišta.

Poređenje brzina izvršavanja:

	<i>u120</i>	<i>u250</i>	<i>u500</i>	<i>u1000</i>
ACO (FFD)	<b>0.049949</b>	0.081303	0.648139	2.843421
ACO (NFD)	<b>0.052021</b>	<b>0.079765</b>	<b>0.491532</b>	<b>1.089492</b>
HGGA	3.339587	2.050825	9.977732	52.317715
MTP	1.848113	2.325678	15.090400	69.615887

Tabela 4.10 : Rezultati upoređivanja ACO algoritma sa drugim algoritmima. Vreme dato u sekundama

