



Rješavanje problema realokacije kontejnera korištenjem metaheuristika

Domagoj Lokner

Mentor: doc. dr. sc. Marko Đurasević

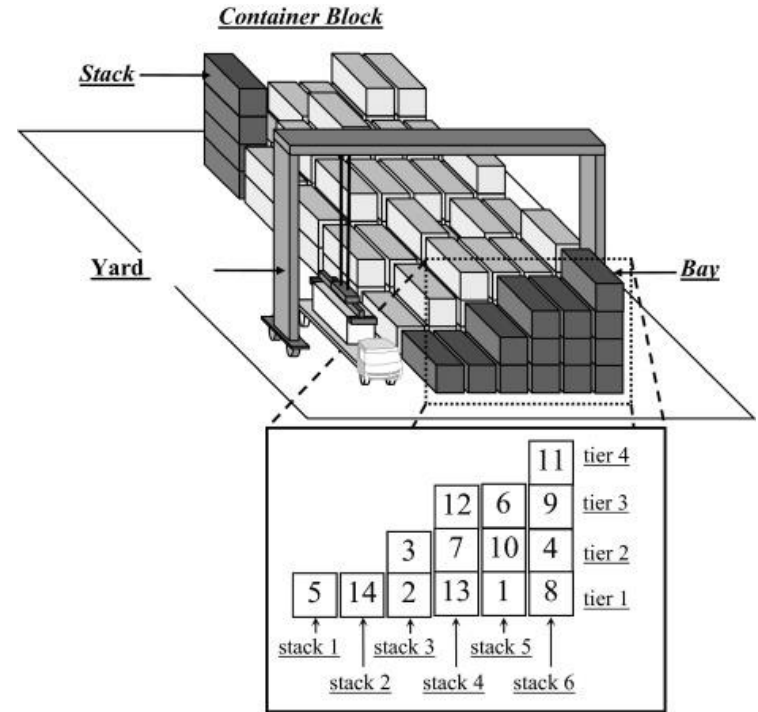
O problemu

- 90% internacionalnog transporta
- vremenska i prostorna ograničenja
- optimiranje procesa



Formalna definicija problema

- podjela na odjeljke i stogove
- optimiziranje rada dizalice

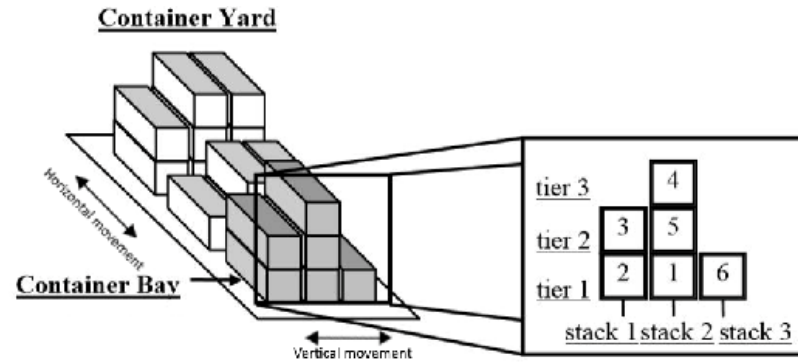




Vrste problema realokacije kontejnera

1. Broj odjeljaka
2. Prioriteti kontejnera
3. Vrijeme definicije prioriteta
4. Ograničenja realokacije

Broj odjeljaka

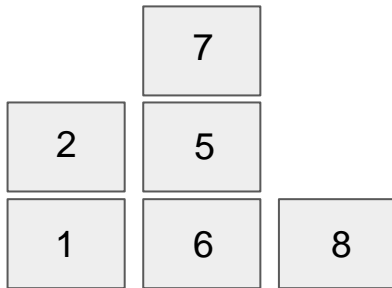


Multi-bay problem

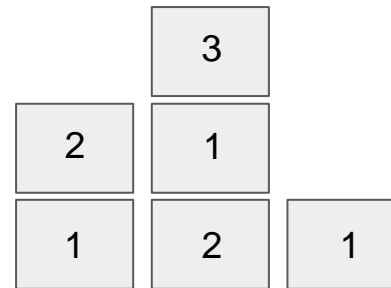
Single-bay problem



Prioriteti kontejnera



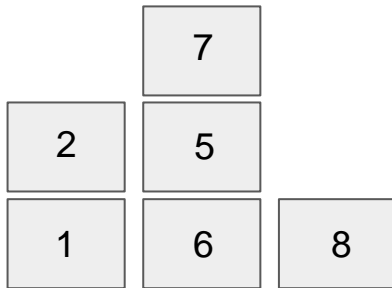
Prioritet po kontejneru



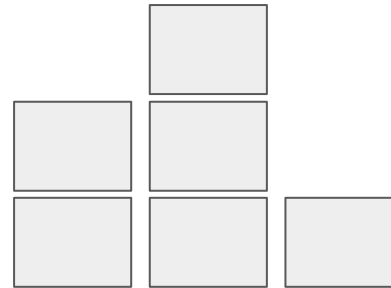
Prioritet po grupi kontejnera



Vrijeme dostupnosti prioriteta

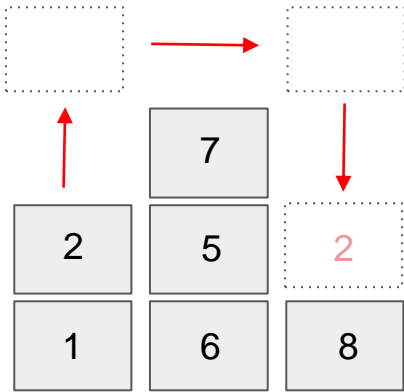


Predefinirani prioriteti

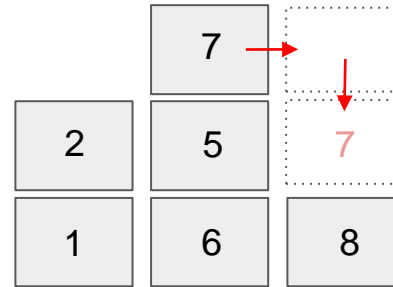


Prioriteti poznati tek u trenutku preuzimanja

Ograničenja realokacije



Samo iznad najvećeg prioriteta



Svi mogu biti realocirani

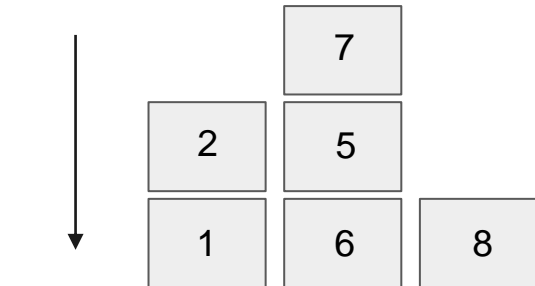
Optimiranje rada dizalice

$$\text{cost} = 0.9 * h + 0.08 * w + 0.02 * l$$

Varijabla	Opis
h	broj pomaka dizalice prema gore
w	broj horizontalnih pomaka dizalice
l	broj pomaka dizalice prema dolje

Procjena potrošnje energije

Lokacija
dostave



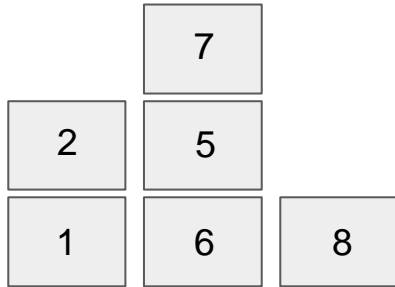


Algoritmi

1. Heuristički algoritmi
2. GRASP
3. GRH
4. Genetski algoritam



Heuristički algoritmi

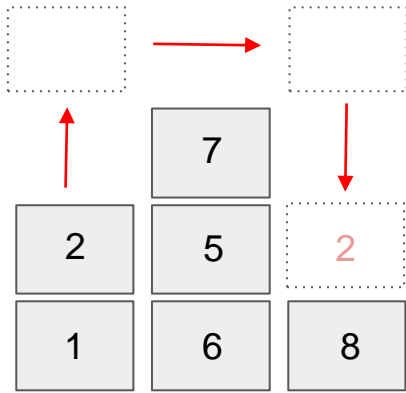


Realokacija na najniži stog

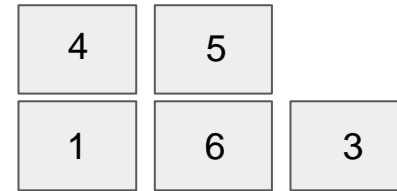


Reshuffle Index

Heuristički algoritmi

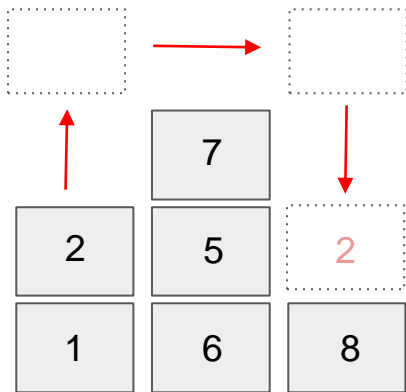


Realokacija na najniži stog

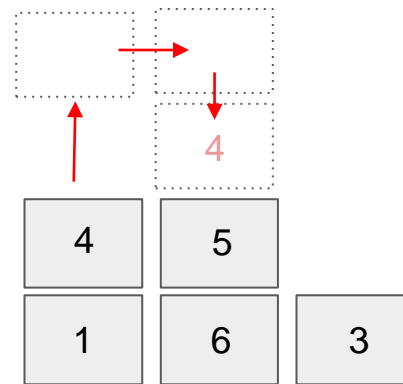


Reshuffle Index

Heuristički algoritmi



Realokacija na najniži stog



Reshuffle Index



GRASP

- **Konstrukcijska faza**
 - Jednostavan heuristički algoritam
- **Faza lokalne pretrage**
 - Najčešće realocirani kontejner
 - Generiranje moguće boljih rješenja
 - Algoritam oporavka

Algoritam 1 GRASP algoritam

najbolje_rjesenje = 0

while *maxIter* > 0 **do**

max_iter − = 1

rjesenje = *konstrukcijska_faza*()

rjesenje = *lokalna_pretraga*(*rjesenje*)

if *rjesenje* > *najbolje_rjesenje* **then**

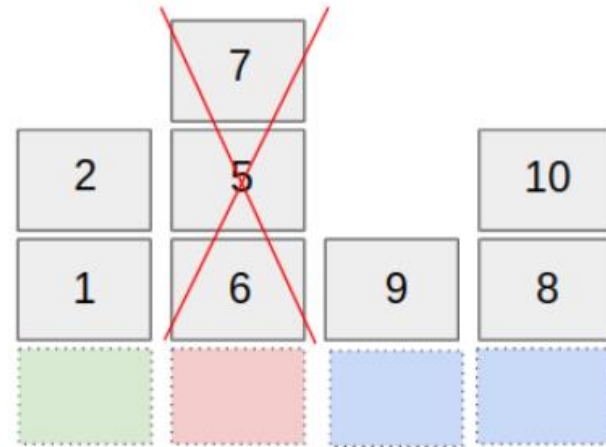
najbolje_rjesenje ← *rjesenje*

end if

end while

GRASP

- **Konstruktivska faza**
 - Jednostavan heuristički algoritam
 - Parametar k
 - Slučajan odabir



Realokacija na najnižu poziciju
uz $k=2$



GRASP

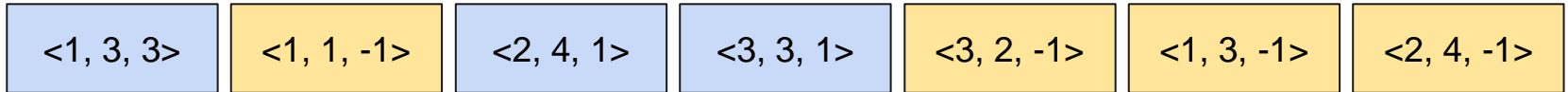
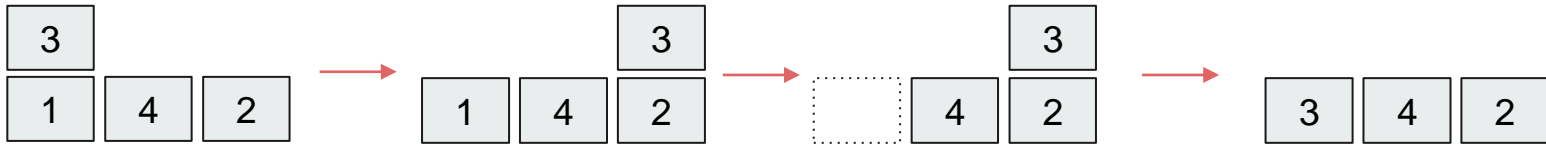
- Faza lokalne pretrage
 - Pronalazak točaka “*loše odluke*”
 - Određivanje moguće boljih realokacija
 - Algoritam oporavka

$\langle x, c, y \rangle$

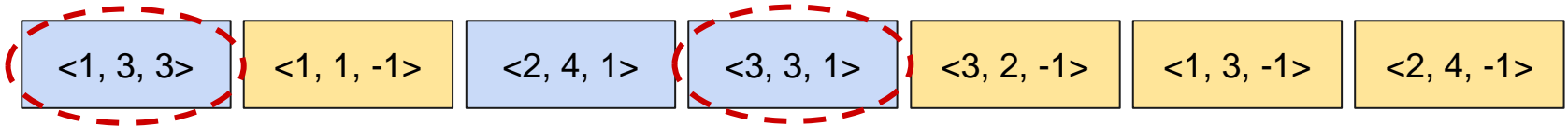
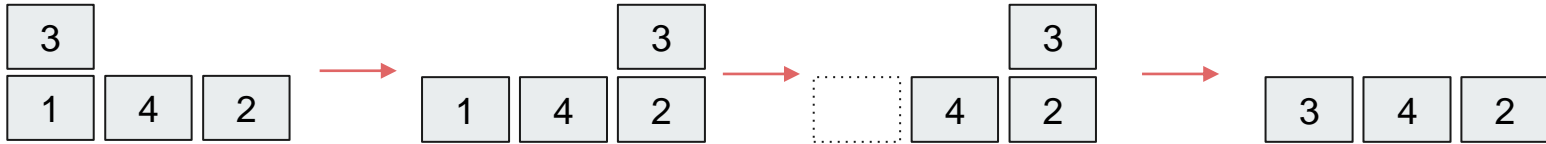
x - stoga sa kojeg premještamo
c - kontejner koji se realocira
y - stog na koji premještamo kontejner



GRASP



GRASP



GRH

Varijabla	Opis
S	broj stogova
T	broj kontejnera na svakom stogu
C	ukupan broj kontejnera (= $S * T$)
maxHeight	najveći dopušten broj kontejnera na stogu

$$\begin{aligned}
 \text{penalty} = & \alpha \left(\frac{h_s}{\text{maxHeight}} \right) + \beta \left(\frac{l_s}{\text{maxHeight}} \right) + \text{gamma} \left(\frac{x_s}{S} \right) + \\
 & + \delta r_s + \varepsilon r_s \left(\frac{c - t_s}{C} \right) + \eta (1 - r_s) g_s + \Theta \left(\frac{k_s}{S} \right) + \mu \left(\frac{n_s}{\text{maxHeight}} \right)
 \end{aligned}$$

Varijabla	Opis
α	važnost minimiziranja podizanja
β	važnost minimiziranja spuštanja
γ	važnost minimiziranja horizontalnog pomaka
δ	važnost minimiziranja ponovnog realociranja
ε	važnost odgađanja ponovnog realociranja
η	<i>tijesnost</i> - koliko je kontejner blizu najmanjem kontejneru na stogu s
Θ	važnost realociranja što bliže mjestu istovara
μ	važnost održavanja stogova niskima

Varijabla	Opis
h_s	broj podizanja potrebnih za realokaciju na stog s
l_s	broj spuštanja potrebnih za realokaciju na stog s
x_s	broj horizontalnih pomaka potrebnih za realokaciju na stog s
r_s	postoji li kontejner većeg prioriteta na stogu s (binarna varijabla)
t_s	kontejner s najnižim indeksom na stogu s
g_s	<i>tijesnost</i> = $(t_s - c - 1)/C$ (c - indeks kontejnera koji se realocira)
k_s	broj pomaka dalje od mjesta utovara (= 0 ako se približavamo)
n_s	broj kontejnera na stogu s

Genetski algoritam

Varijabla	Opis
G	broj generacija
N	broj jedinki u generaciji
n	broj instanci problema korištenih za evaluaciju
N_r	broj nasumično stvorenih jedinki u generaciji
N_p	broj roditelja koji su odabrani u svakoj generaciji

Algorithm 2 Genetski algoritam

Require:

N - velicina populacije

N_e - broj jedinki koje prenosimo elitizmom

N_r - broj nasumično odabranih jedinki u novoj generaciji

max_iter - broj generacija

$populacija = nasumično_stvori_jedinke(N)$

for $i = 0; i < max_iter; i++$ **do**

$evaluira(j(populacija))$

$roditelji = odaberi_roditelje(populacija)$

$potomci = krianjzanje(roditelji, N - N_e - N_r)$

$nasumični_potomci = nasumično_stvori_jedinke(N_r)$

$elitni_potomci = elitizam(N_e, populacija)$

$populacija = stvori_novu_populaciju(potomci, nasumični_potomci, elitni_potomci)$

end for

return $najbolje_rjesenje(populacija)$

Optimiranje parametara GRH algoritma

$$\text{penalty} = \alpha\left(\frac{h_s}{\text{maxHeight}}\right) + \beta\left(\frac{l_s}{\text{maxHeight}}\right) + \text{gamma}\left(\frac{x_s}{S}\right) + \\ + \delta r_s + \varepsilon r_s\left(\frac{c - t_s}{C}\right) + \eta(1 - r_s)g_s + \Theta\left(\frac{k_s}{S}\right) + \mu\left(\frac{n_s}{\text{maxHeight}}\right)$$

Varijabla	Opis
α	važnost minimiziranja podizanja
β	važnost minimiziranja spuštanja
γ	važnost minimiziranja horizontalnog pomaka
δ	važnost minimiziranja ponovnog realociranja
ε	važnost odgađanja ponovnog realociranja
η	<i>tijesnost</i> - koliko je kontejner blizu najmanjem kontejneru na stogu <i>s</i>
Θ	važnost realociranja što bliže mjestu istovara
μ	važnost održavanja stogova niskima



Kromosom genetskog algoritma



Pregled rezultata



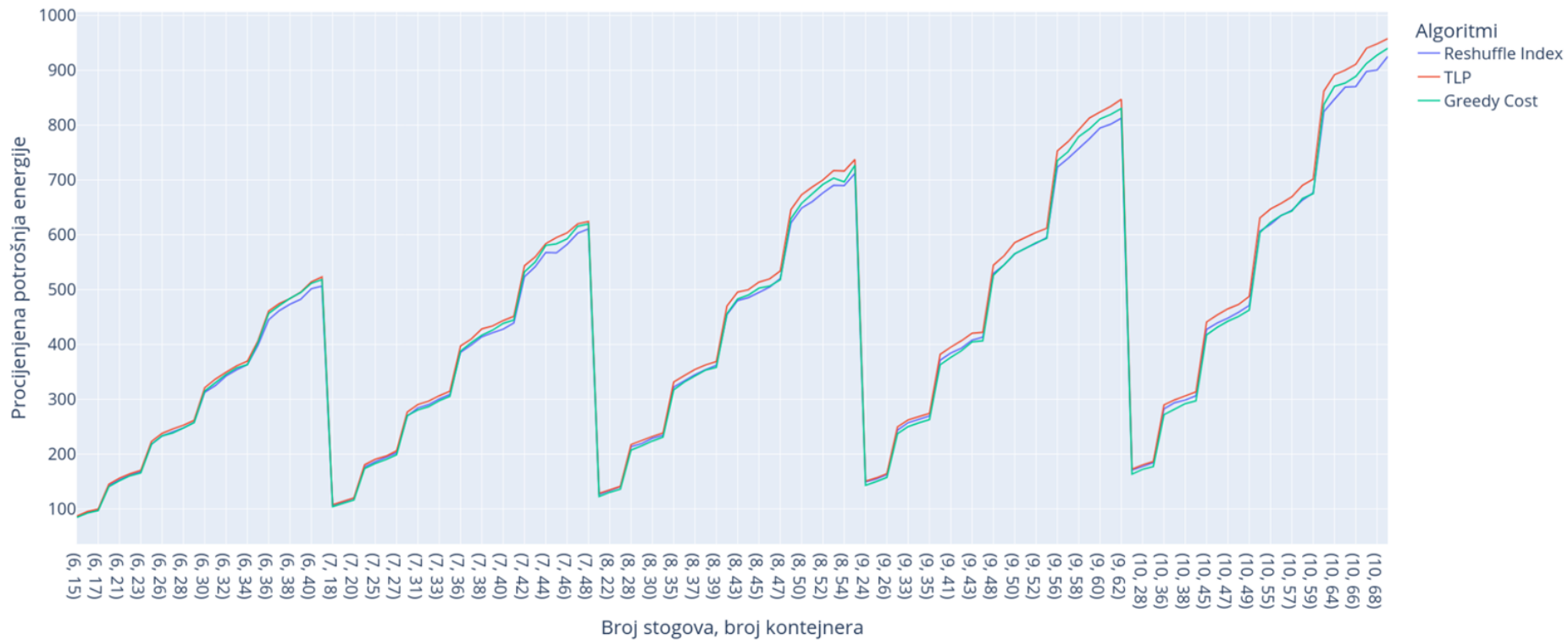
Instance problema

10	54					
6	9	35	49	24	26	31
6	34	42	23	3	13	12
6	28	52	51	50	53	48
6	11	14	10	8	41	22
1	45					
6	29	33	17	21	38	36
6	20	7	39	37	16	1
5	15	18	43	40	25	
6	44	6	4	2	5	47
6	46	30	19	54	32	27

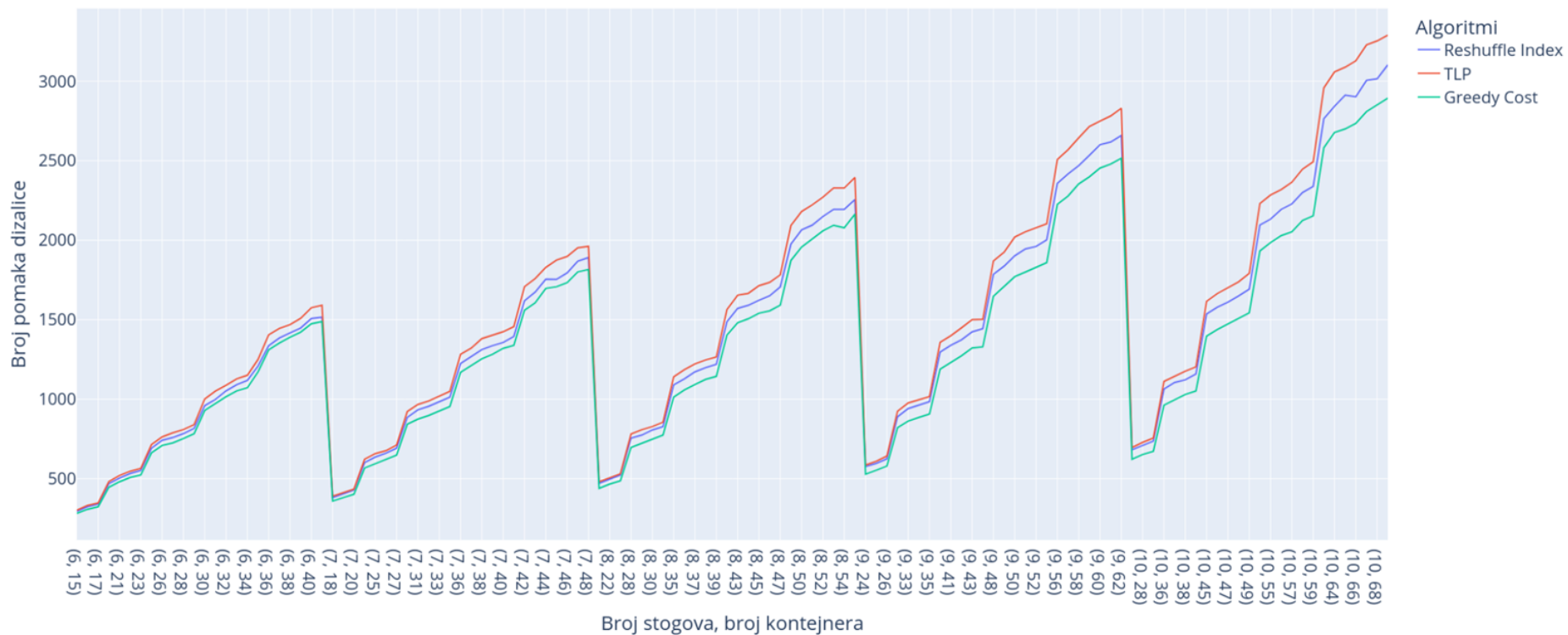


Heurističké metode

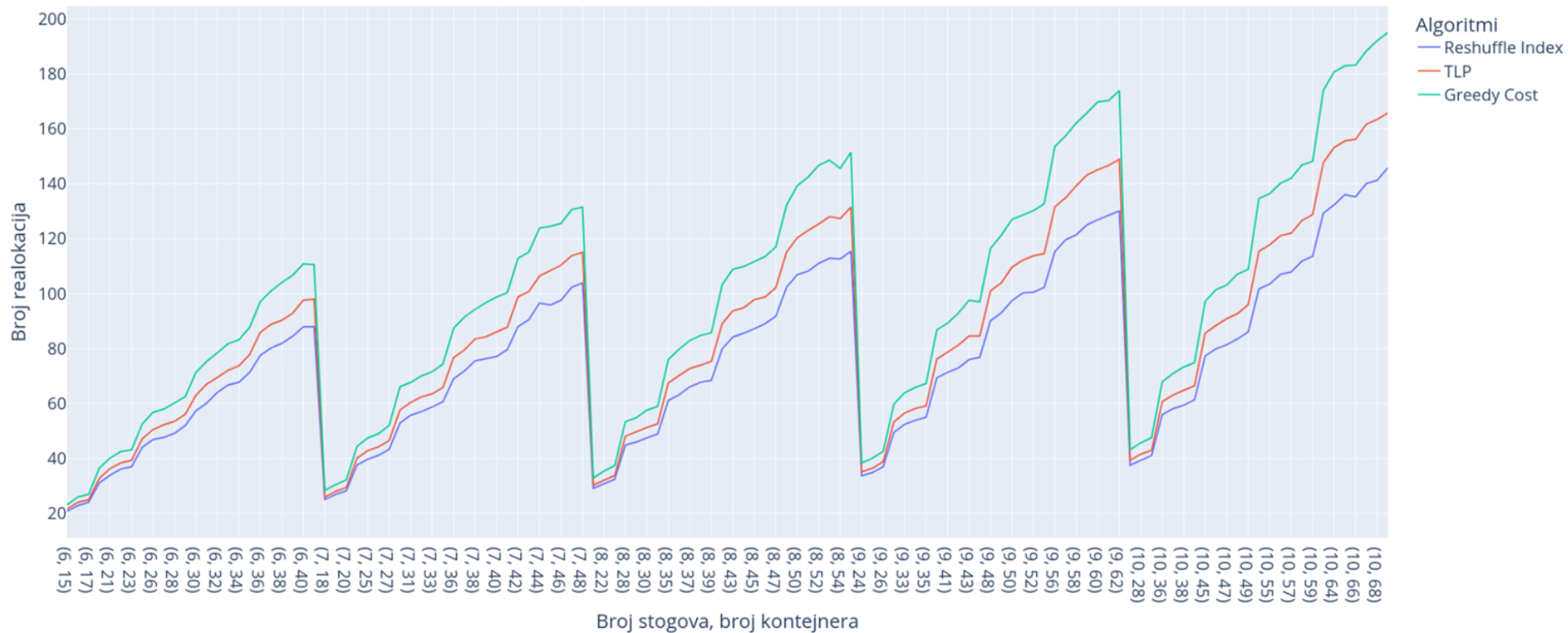
Procijenjena potrošnja energije



Broj pomaka dizalice



Broj realokacija





Metaheuristički algoritmi



Parametri metaheurističkih algoritama

GRASP

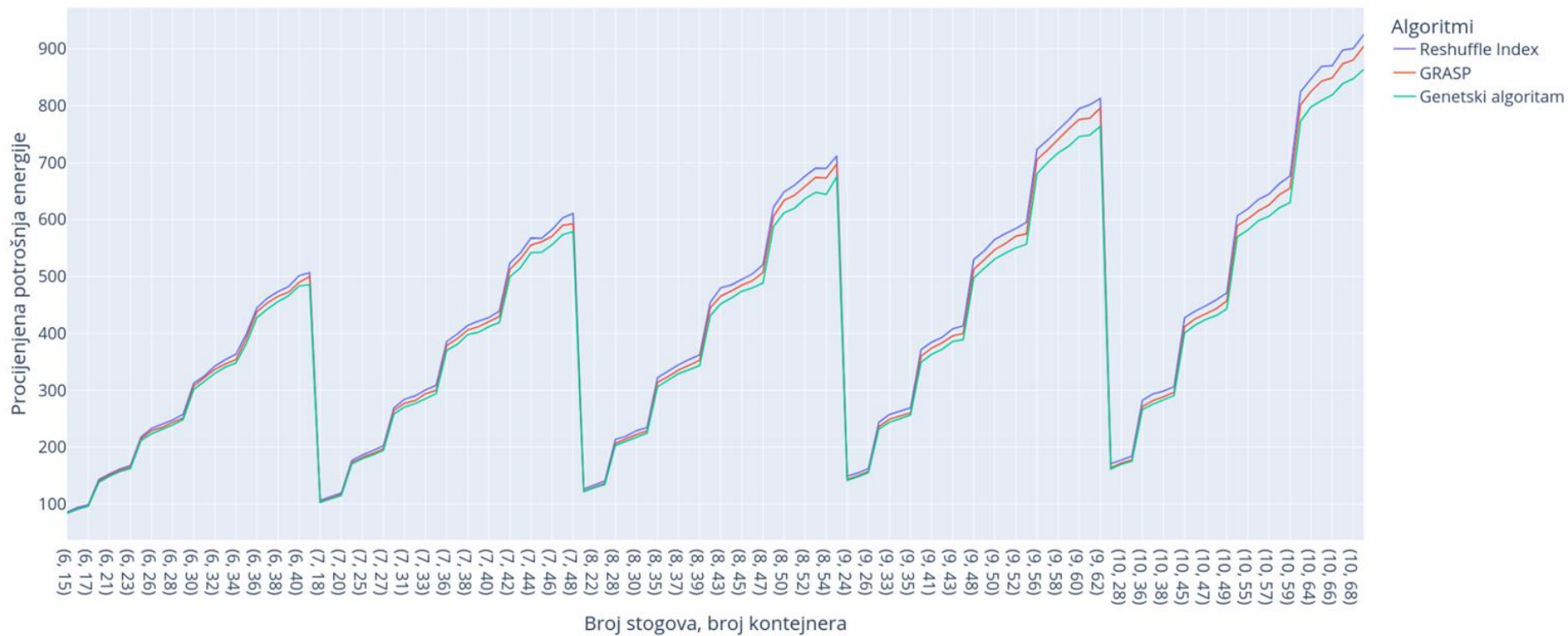
- **Konstrukcijska faza**
 - Na najnižu poziciju
- **Faza lokalne pretrage**
 - Najjeftinija realokacija
- **Algoritam oporavka**
 - Reshuffle Index

Genetski algoritam

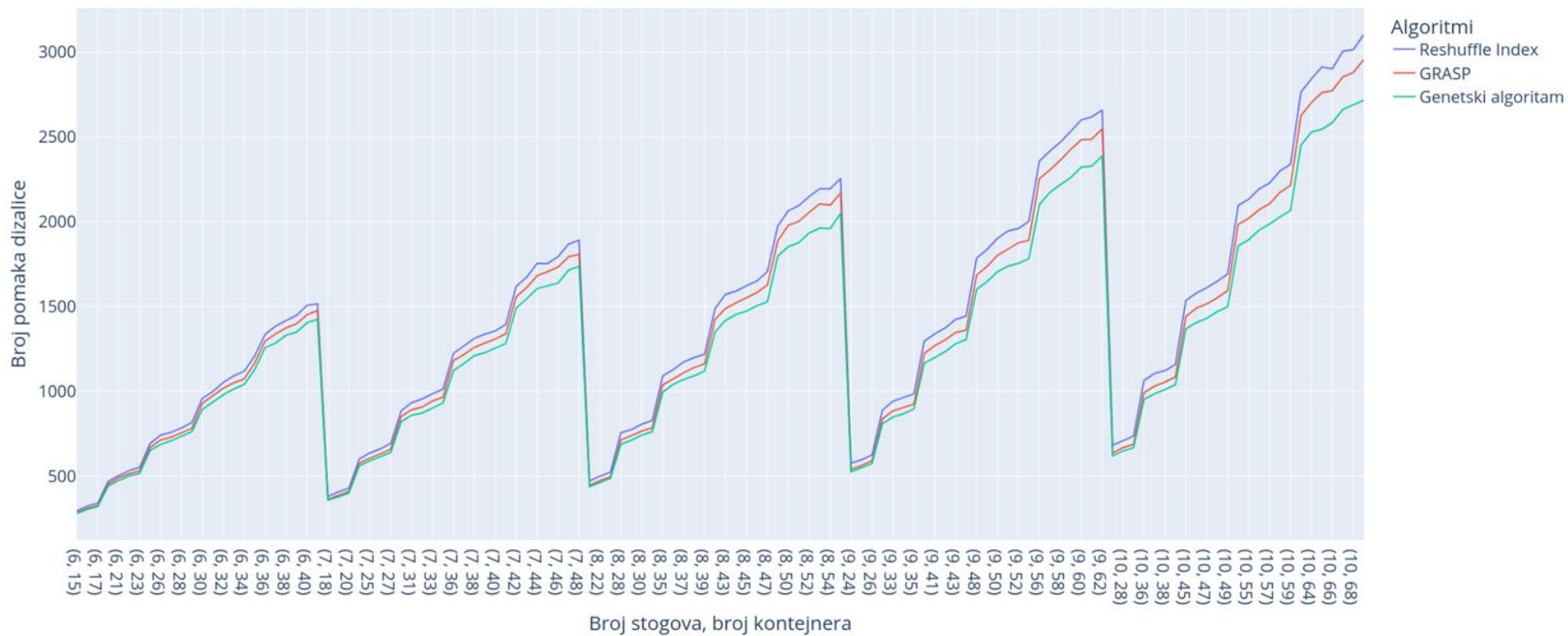
- **3 kolekcije optimiranih parametara**

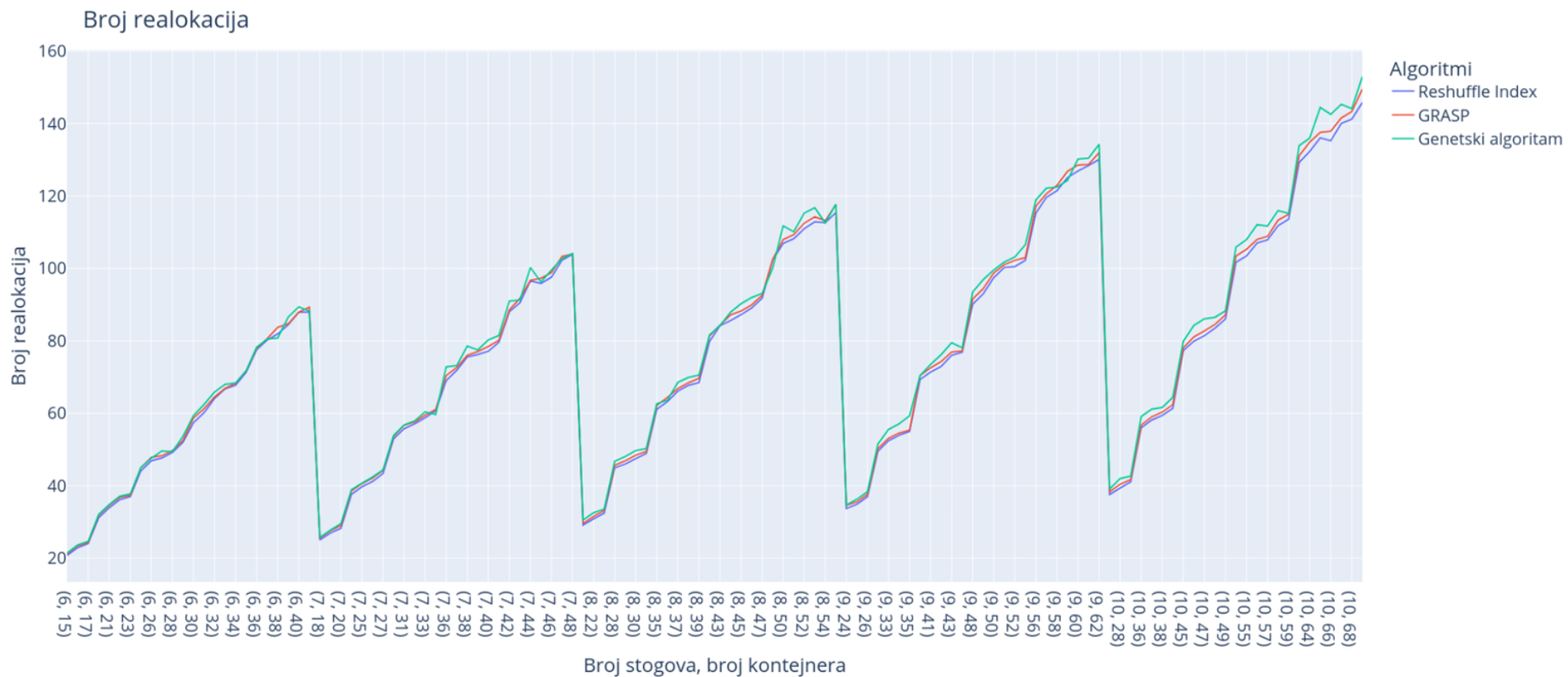
	veličina populacije - N	generacije - G	instance za evaluaciju - n
<i>GA1</i>	50	100	20
<i>GA2</i>	75	100	20
<i>GA3</i>	50	200	20

Procijenjena potrošnja energije



Broj pomaka dizalice





Zaključak



- **Optimiranje potrošnje dizalice**
 - Ušteda energije i vremena
- **Veće instance problema zahtijevaju kompleksnije algoritme**
- **Nastavak istraživanja:**
 - Različite instance problema i različita ograničenja
 - Kompleksniji heuristički algoritmi
 - Dodati parametri GRH algoritma

Hvala na pažnji!