


Završni rad br. 6984



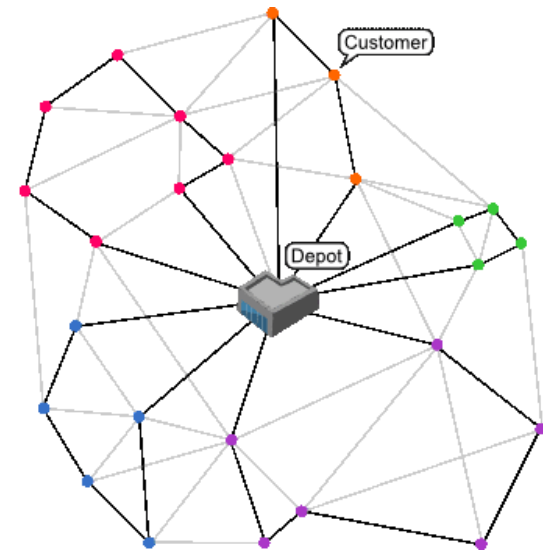
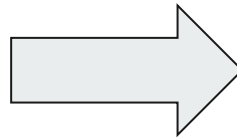
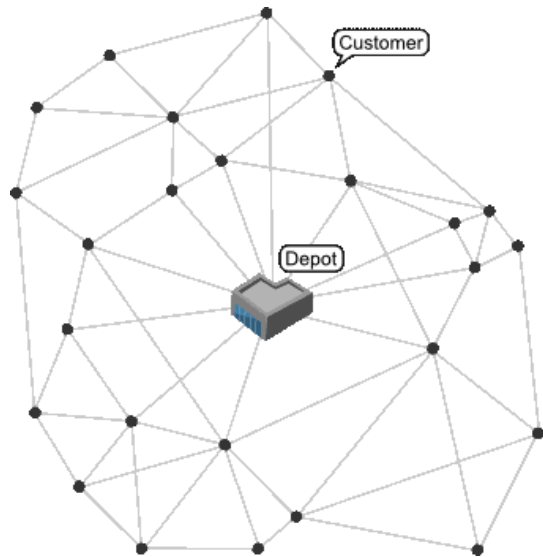
OPTIMIZACIJA PROBLEMA USMJERAVANJA VOZILA

Domagoj Lokner

Mentor: doc. dr. sc. Marko Đurasević

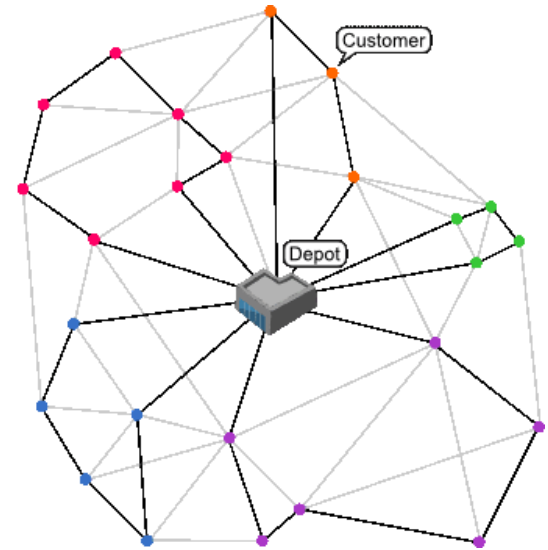


Problem usmjeravanja vozila



Problem usmjeravanja vozila

- optimizacija puta flote vozila
- problem kombinatorne optimizacije
- NP-težak problem
- George Dantzig i John Ramser 1959.
- poopćena varijanta problema trgovačkog putnika



Inačice problema

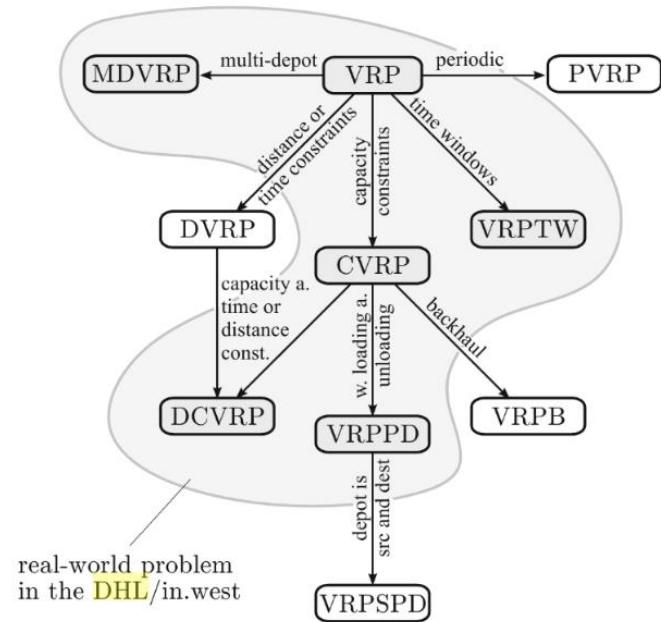
- minimizacija broja potrebnih vozila
- minimizacija troškova puta

CVRP (engl. *Capacitated VRP*)

- vozila s ograničenim kapacitetom dobara

VRPTW (engl. *VRP with Time Windows*)

- usmjeravane vozila s vremenskim prozorom





ALGORITMI

ALGORITAM UŠTEDA



ALGORITAM UŠTEDA

- Clarke i Wrightov algoritam ušteda
- pohlepna heuristika

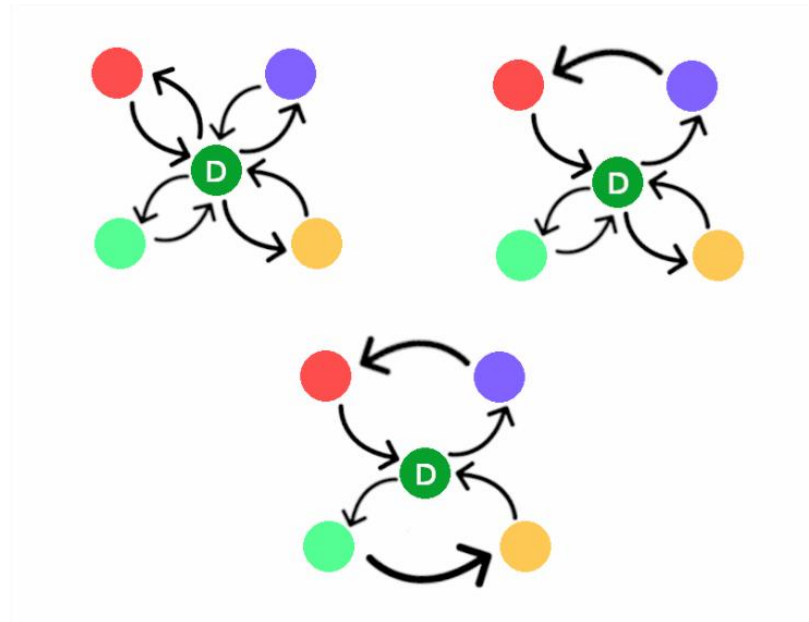
JEDNO VOZILO
PO KUPCU

$$2 \sum_{i=1}^n l(D, v_i)$$

UŠTEDA

$$\begin{aligned} s(v_i, v_j) &= 2l(D, v_i) + 2l(D, v_j) - (l(D, v_i) + l(v_i, v_j) + l(D, v_j)) \\ &= l(D, v_i) + l(D, v_j) - l(v_i, v_j) \end{aligned}$$

ALGORITAM UŠTEDA

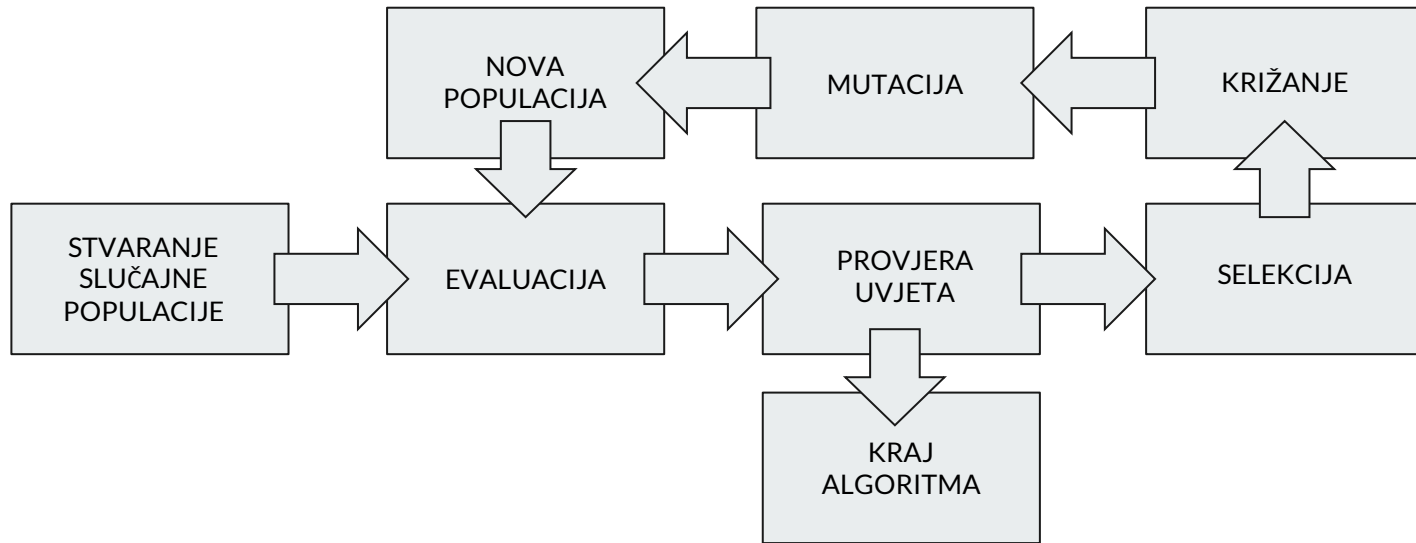




ALGORITMI

GENETSKI ALGORITAM

GENETSKI ALGORITAM

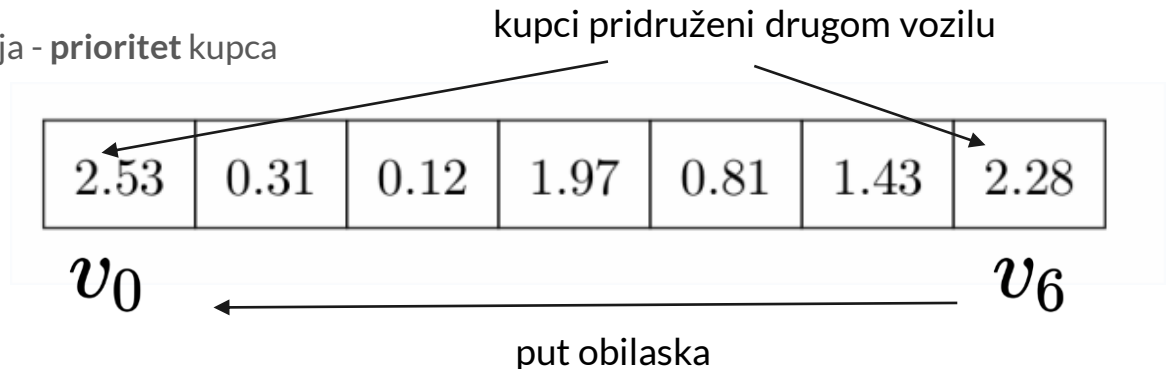




PRIKAZI RJEŠENJA

KODIRANJE SLUČAJNIM KLJUČEM

- veličina kromosoma odgovara količini kupaca
- svi brojevi u rasponu - $[0, broj_vozila>$
- indeks predstavlja kupca
 - cijeli dio broja - vozilo kojem se pridružuje kupac
 - decimalni dio broja - prioritet kupca





KODIRANJE POMIČNIM ZAREZOM

- veličina kromosoma odgovara količini kupaca
- svi brojevi u rasponu - $[0, 1]$
- indeks predstavlja kupca

0.85	0.21	0.13	0.57	0.29	0.44	0.71
------	------	------	------	------	------	------

- interval $[0, 1]$ dijelimo na n jednakih dijelova, pri čemu n odgovara broju vozila
- svakom vozilu pridružujemo kupca koji se nalazi unutar njegovog intervala
- vozilo prvo obilazi kupce s manjim brojem



FUNKCIJA DOBROTE



FUNKCIJA DOBROTE

- procjena kvalitete rješenja
- ukupna cijena određenog rasporeda vozila
- vrijednost funkcije nastojimo minimizirati

$fitness = distance + 1000 \cdot overcapacity$

if *timeWidow* **then**

$fitness = fitness + overtime$

if *late* > 0 **then**

$fitness = fitness \cdot late \cdot numberOfVehicles$

end if

end if

return *fitness*



INSTANCE PROBLEMA



INSTANCE PROBLEMA

- Hombergove instance
- Tri kategorije:
 - **C** - grupirani kupci
 - **R** - uniformno raspodijeljeni kupci
 - **RC** - kombinacija gornja dva tipa

C101

VEHICLE NUMBER	CAPACITY
25	200

CUSTOMER CUST NO.	XCOORD.	YCOORD.	DEMAND	READY TIME	DUE DATE	SERVICE	TIME
0	40	50	0	0	1236	0	
1	45	68	10	912	967	90	
2	45	70	30	825	870	90	
...							



ANALIZA REZULTATA



ANALIZA REZULTATA

- instanca **tipa C**
 - C_1_4_1
- **50 jedinki** u populaciji
- **58 000 evaluacija** rješenja, odnosno **1200 evolucija**
- jedna jedinka se prenosi elitizmom iz svake populacije u slijedeću
- **K-turnirska** selekcija, pri čemu je **K=5**

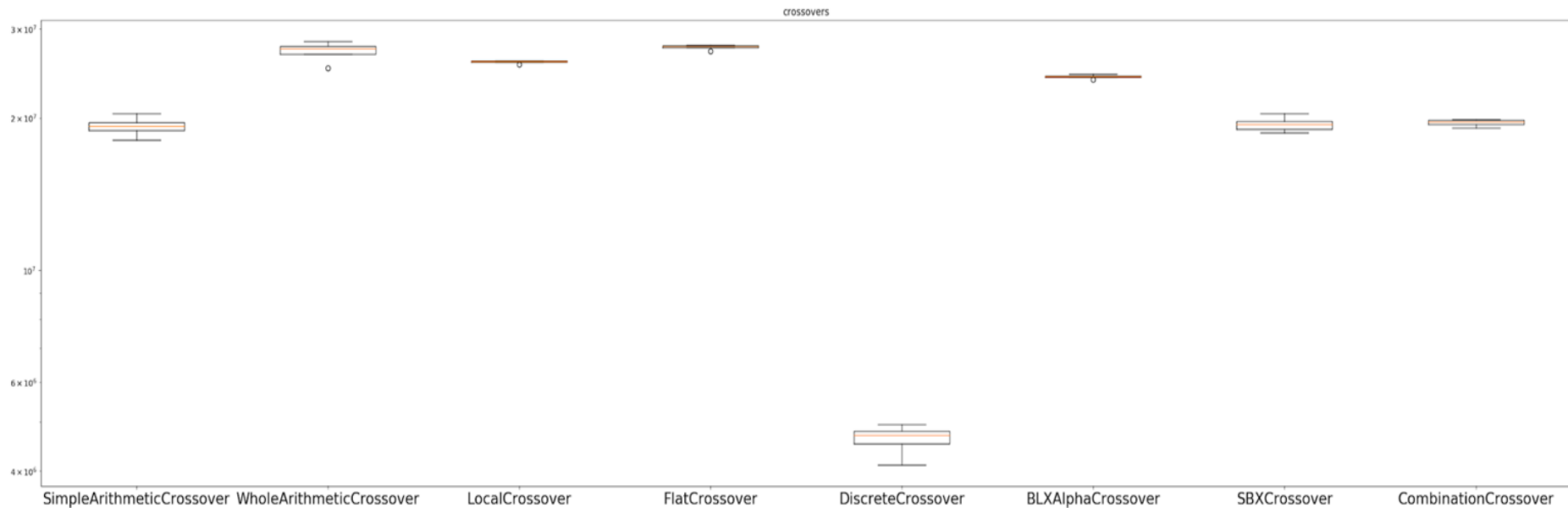


ODREĐIVANJE OPERATORA

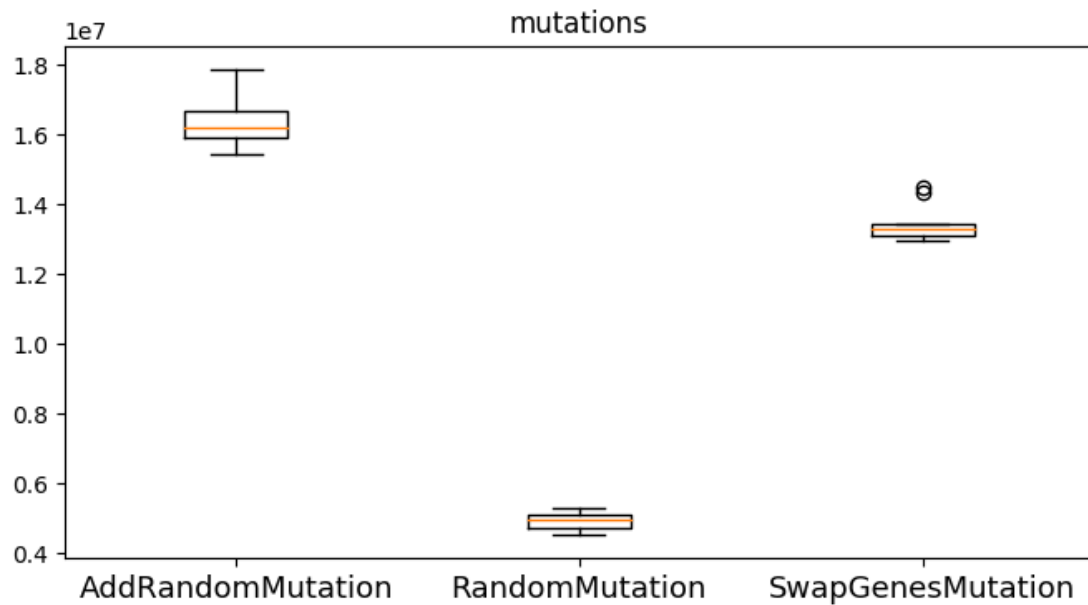
1. KRIŽANJE
2. MUTACIJA
3. VJEROJATNOST MUTACIJE



KRIŽANJA

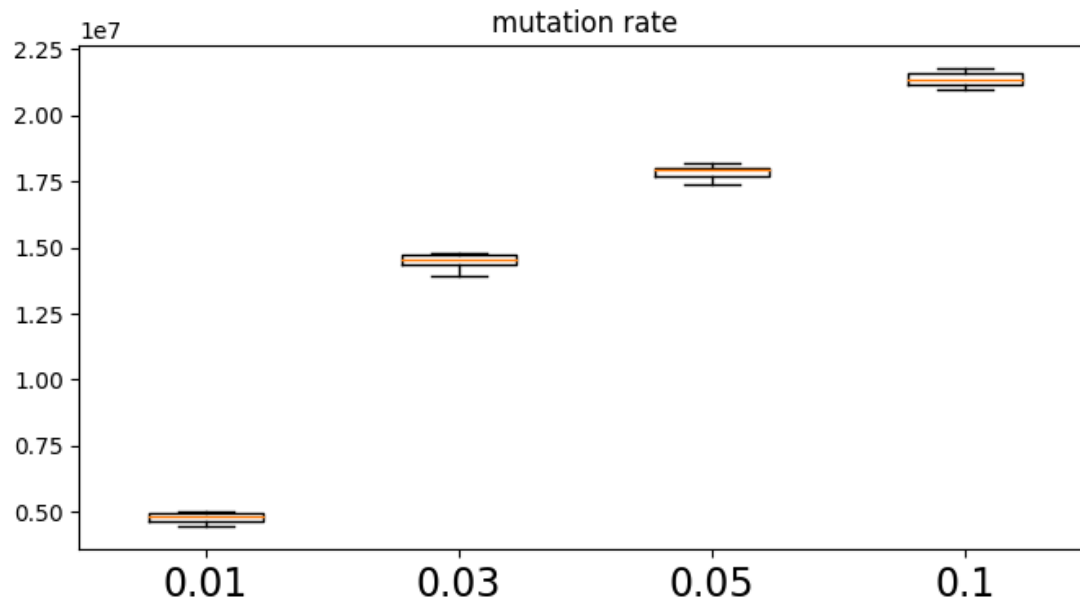


MUTACIJE





VJEROJATNOST MUTACIJE



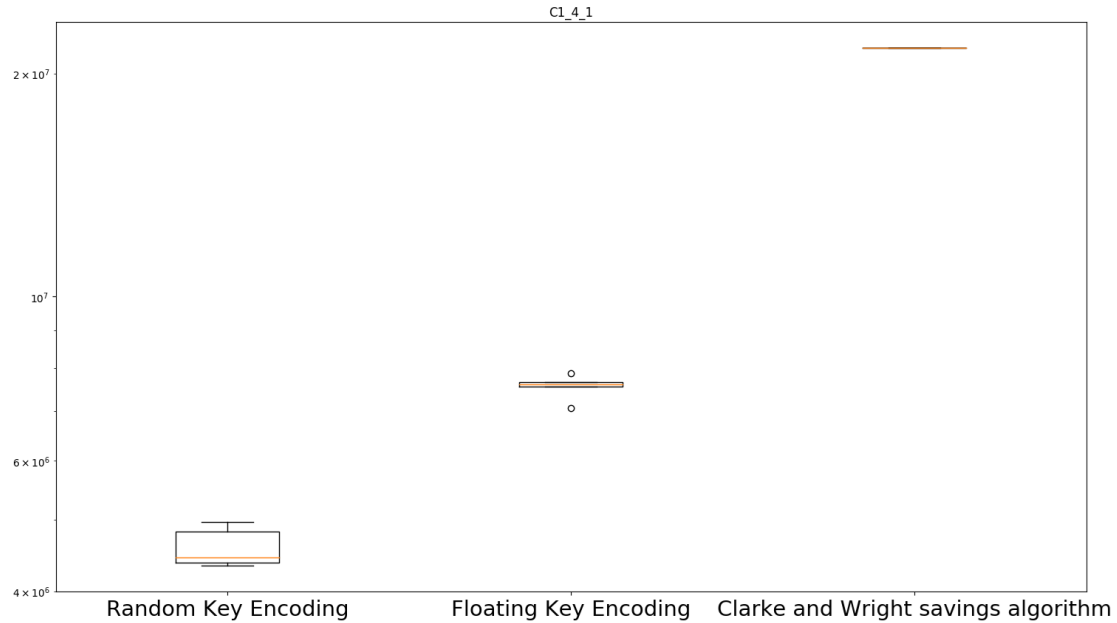


USPOREDBA REZULTATA NA RAZLIČITIM INSTANCAMA PROBLEMA

- križanje - `DiscreteCrossover`
- mutacija - `RandomMutation`
- vjerojatnost mutacije - 1%

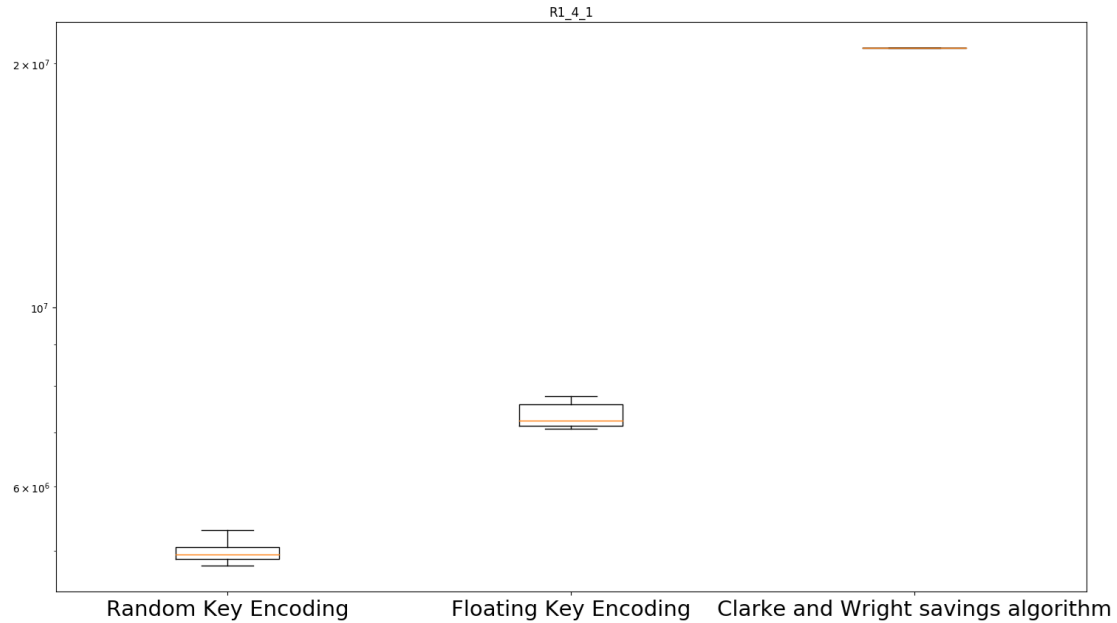


C INSTANCA (C_1_4_1)



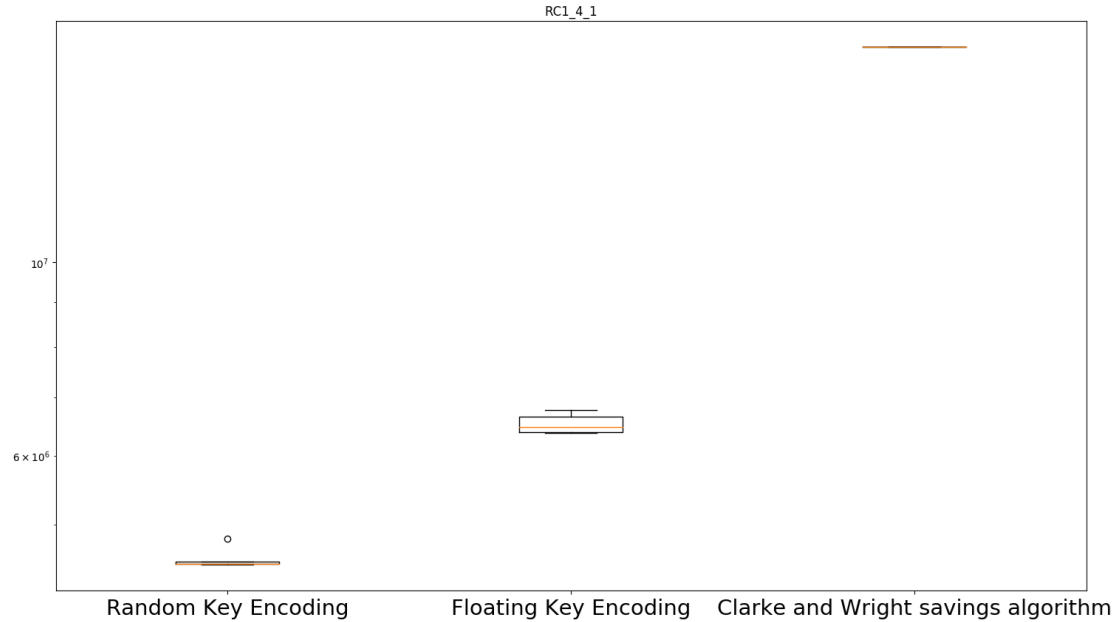


R INSTANCA (R_1_4_1)





RC INSTANCA (RC_1_4_1)





USPOREDBA REZULTATA NA RAZLIČITIM INSTANCAMA PROBLEMA

algoritam	vrijednost	C	R	RC
RKE	minimum	4 329 882	4 792 491	4 493 631
	prosjek	4 587 390	4 996 640	4 570 151
	maksimum	4 967 740	5 306 096	4 819 033
FPE	minimum	7 072 007	7 081 652	6 364 000
	prosjek	7 557 567	7 360 116	6 525 733
	maksimum	7 875 331	7 764 609	6 761 640
C-W	rezultat	21 652 588	20 895 398	17 678 605



ZAKLJUČAK

- **genetski algoritam** rezultira boljim rezultatima naspram algoritma ušteta
- operatori - **DiscreteCrossover** i **RandomMutation (1%)**
- prikaz - **Kodiranje slučajnim ključem**
- **Poboljšanja algoritma**
 - isprobati različite prikaze
 - uvođenje strategije za inicijalizaciju početne populacije



DODACI



ALGORITAM UŠTEDA

Algorithm 1 Clarke and Wright savings algorithm

Korak 1: Izračunaj uštede za sve parove (i, j) za koje vrijedi $i, j \in V$ i $i \neq j$

Korak 2: Sortiraj izračunate uštede silazno od najveće prema najmanjoj

Korak 3: Za svaku uštedu $s(i, j)$, iz sortiranog niza ušteda, uključi poveznicu (i, j) u rutu ako niti jedno ograničenje neće biti narušeno i ako vrijedi:

1. Niti i , niti j nisu već dio postojeće rute. Tada stvori novu rutu koja će sadržavati vezu (i, j)
2. Ili, samo je jedan klijent uključen u rutu i nalazi se na *vanjskom*¹ dijelu te rute. Tada dodaj vezu (i, j) na tu rutu.
3. Ili, su oba klijenta, i i j , dio postojećih ruta te su obadva *vanjski* dio ruta. Tada povežujemo te dvije rute vezom (i, j) .

Korak 4: Svi kupci koji su preostali, to jest nisu pridodani niti jednoj rutu, pridjeljuju se vozilima koja će poslužiti samo njih i vratiti se u depo, drugim riječima stvaraju zasebnu rutu.
