Problem upotpunjenja grafa do Hamiltonovog

Ante Grbić, Hermina Petric Maretić

Prirodoslovno-matematički fakultet - Matematički odjel
Seminar za teorijsko računarstvo

18. ožujka, 2013.

Sadržaj

Opis problema

Neka je zadan neusmjeren, bestežinski graf $G = (V,E)$. Potrebno je dodati minimalan broj bridova $d \in \mathbb{N}$ grafu G tako da graf postane Hamiltonov, odnosno da sadrži Hamiltonov ciklus. Jasno je da je ovaj problem NP-težak jer rješava problem traženja Hamiltonovog ciklusa.
NP-potpunost problema traženja Hamiltonovog ciklusa dokazana je 1972. u *Reducibility Among Combinatorial Problems*, Karp.

Povijest problema i dosadašnji rezultati

- Problem su neovisno predstavili Boesch, Chen i McCugh te Goodman i Hedetniemi 1970. godine.
- Objavili su algoritme koji u polinomijalnom vremenu konstruiraju optimalni pokrivač jednostavnim putevima za stablo.
- Kundu je 1976. njihove rezultate poboljšao pokazavši da se isti algoritam može izvesti u linearnom vremenu.

Povijest problema i dosadašnji rezultati

2012. godine David Gamarnik i Maxim Sviridenko računaju asimptotsko rješenje problema za rijetke grafove $G = (V, \frac{c}{n})$, gdje je $c < 1$, a $\frac{c}{n}$ označava vjerojatnost s kojom se svaki brid može pojaviti u grafu G.

Primjena problema

- Dodijeljivanje frekvencija
- Optimizacija programskog koda
- Distribuirani procesi (dodijeljivanje procesa distribuiranim procesorima)

Algoritmi

- Genetski algoritam - krizanje poretkom
- Genetski algoritam - pohlepno križanje
- Imunološki algoritam
- Mravlji algoritam
- Mravlji algoritam "Blizanac"

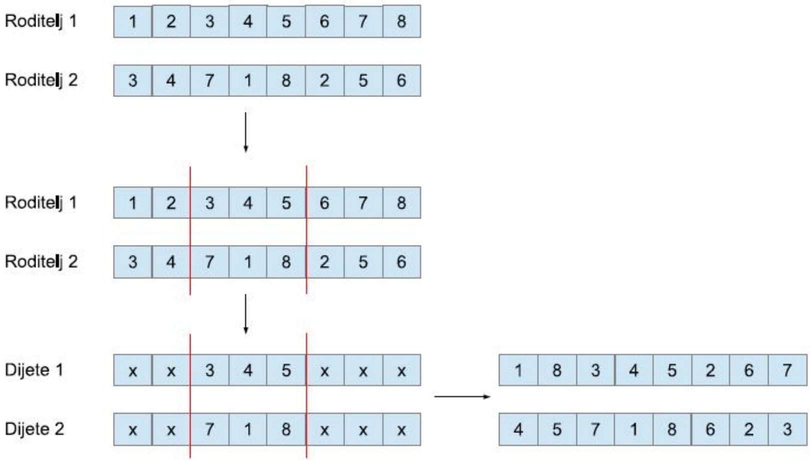
Genetski algoritam

Reprezentacija rješenja: permutacije čvorova bez čvora broj 1.

Algoritam (počinje se sa slučajno generiranom populacijom veličine *brJed*):

1. Ocijeni jedinke.
2. Odaberi dvije najbolje i sačuvaj ih u *novaPop*. Roulette wheel selekcijom odaberi $brJed - 2$ jedinki i spremi ih u *novaPop*.
3. Metodom križanja poretka križaj odabrane jedinke.
4. Mutacijom koja je implementirana kao implementacija inverzije mutiraj jedinke u *novaPop*.
5. Vjerojatnost mutacije se s vremenom smanjuje.
6. Početnu populaciju zamijeni s *novaPop*. Nastavi algoritam dok nije zadovoljen uvjet prekida.

Križanje poretkom



(preuzeto iz (5))

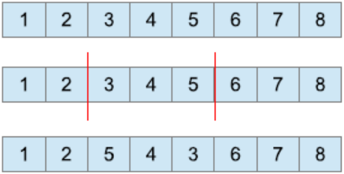
Moguće poboljšanje genetskog algoritma

Moguće poboljšanje pohlepnim operatorom križanja, koji se kao ideja pojavljuje u problemu trgovačkog putnika.
Princip je taj da svi bridovi koji su zajednički roditeljima (ciklusima) postanu osnova njihove djece. Ostatak bridova se generira slučajno.

Imunološki algoritam

1. Stvorimo inicijalnu populaciju antitijela veličine d.
2. Svako antitijelo kloniramo k puta (sada imamo (k+1)*d antitijela).
3. Svaki klon prolazi proces hipermutacije.
4. Računamo afinitete starih antitijela i hipermutiranih klonova te gradimo novu populaciju od d najboljih.

Hipermutacija



(preuzeto iz (5))

Mravlji algoritam

1. Napravimo kopiju matrice prijelaza koju će mravi mijenjati.
2. Svi mravi kreću iz istog vrha i odabiru bridove s vjerojatnošću $\frac{c_n}{\sum c_n}$ ako postoje slobodni bridovi, odnosno nasumično dodaju novi brid ako ne postoji nastavak puta.
3. Mravi evaluiraju svoja rješenja i uspoređuju ih s dosad najboljim putevima.
4. Najbolji mrav ostavlja feromone na svim već postojećim bridovima (i samo njima).
5. Trag na cijelom grafu slabi.

Mravlji algoritam - "blizanac"

- Pohlepna modifikacija mravljeg algoritma
- Mravi kreću iz vrha s najmanje susjeda
- Kad dođu do kraja puta, javlja se blizanac koji se nalazi na početku puta i nastavlja put iz početne točke
- Kad blizanac dođe do kraja puta, dodaje brid prema onom vrhu koji ima najmanje susjeda i postupak se nastavlja.

Blizanac opravdava pohlepnost algoritma!

Dokaz (1)

Neka je $x = x_1, x_2, ..., x_\mathrm{n}$ jedan optimalan ciklus i $x_\mathrm{k} = a_1$ element s minimalnim brojem susjeda. Mrav može kopirati put $x_\mathrm{k}, ..., x_{\mathrm{k+l}}$ sve dok su ti bridovi postojeći u grafu. Neka je $a_1 ... a_{\mathrm{l}+1} = x_k ... x_{k+l}$. Ako $a_{\mathrm{l}+1}$ više nema susjeda, dolazi blizanac, a ako ima, mrav bira susjeda i opet kopira sve postojeće bridove iz x u jednom smjeru. Dalje induktivno.
Primijetimo da ovim postupkom nismo poremetili optimalnost ciklusa (nismo dodali nijedan brid, a razdvojenih dijelova ima manje ili jednako kao i prije).

Dokaz (2)

Sada dolazi blizanac. On može kopirati sve postojeće i neposjećene bridove prije x_{k}, neka su to $x_{\mathrm{k-1}}, \ldots, x_{\mathrm{k-s}}$. Nakon toga ponavljamo već opisani postupak dok blizanac ne dođe do vrha bez susjeda. Mrav skače na neiskorišteni vrh s najmanjim brojem susjeda i ponavljamo gornje argumente.
Poanta je u tome da ako mrav krene od "krivog" vrha ili skoči na njega, blizanac lako ispravi problem.

Moguće poboljšanje genetskog i imunološkog algoritma

- Početnu populaciju ne generiramo nasumično, već na neki educirani način
- Koristimo mrave kako bismo generirali početne jedinke
- Drastično poboljšanje, pogotovo za genetski algoritam

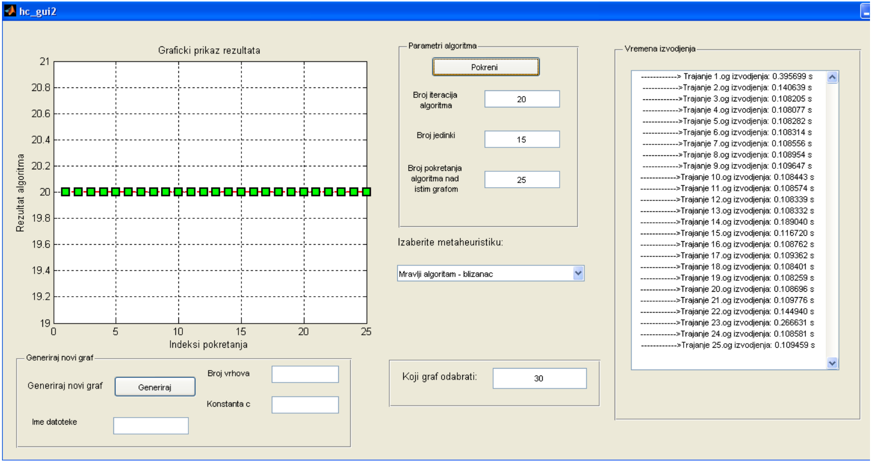
Napomena

Promatramo samo grafove oblika $G = (V, \frac{c}{n})$, gdje $\frac{c}{n}$ označava vjerojatnost s kojom se svaki brid može pojaviti u grafu G po uzoru na ranije spomenuti članak autora Gamarnik, Sviridenko: *Hamiltonian Completions of Sparse Random Graphs*.

Implementacija

- Svi algoritmi su implementirani u MATLAB-u
- Grafovi su spremljeni pomoću matrica incidencije
- Matrice su čuvane u kompresiranom sparse formatu
- Funkcije za I/O operacije sa sparse matricama su preuzete sa `http://math.nist.gov/MatrixMarket/mmio/matlab/mmiomatlab.html`

Korisničko sučelje

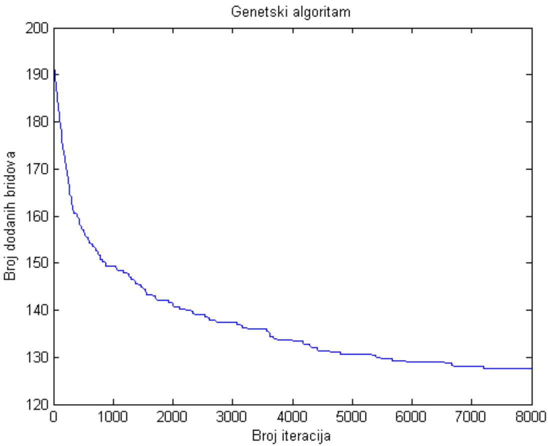


Opis testiranja

- Testiranja svih algoritama su provedena na 30 nasumično generiranih grafova od 200 vrhova i 2 nasumično generirana grafa od 1000 vrhova.
- Sva testiranja su provedena tri puta nad istim grafovima s jednakim parametrima te rezultati prikazuju usrednjenje.
- Bridovi na svim grafovima su generirani s vjerojatnošću $1/n$.
- Posebno su provedena testiranja za Mravlji algoritam "Blizanac" na nasumično generiranim grafovima od 2000, 4000, 8000 i 10000 vrhova.

Sva testiranja obavljena su na računalu s procesorom Intel Core i5, 2,3GHz i 6GB RAM memorije.

200 vrhova

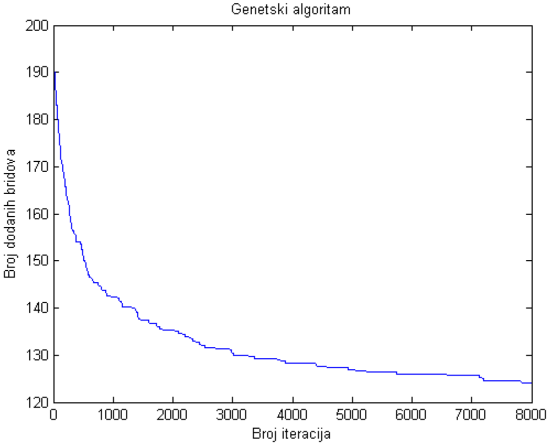


Elitizam: 2

Trajanje izvođenja: 52.63 sekunde

Najbolji rezultat: 127.667

200 vrhova

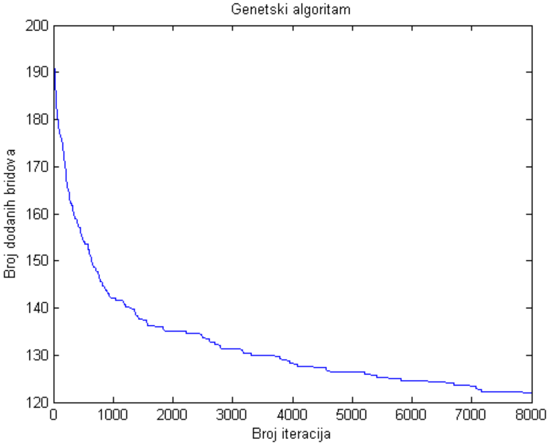


Elitizam: 4

Trajanje izvođenja: 47.5709 sekunde

Najbolji rezultat: 124

200 vrhova

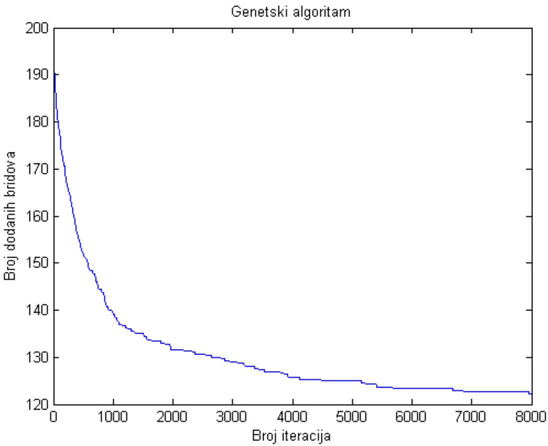


Elitizam: 6

Trajanje izvođenja: 42.68 sekunde

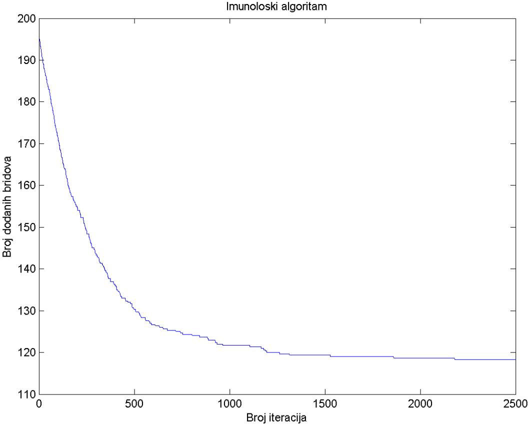
Najbolji rezultat: 122

200 vrhova



Elitizam: 8
Trajanje izvođenja: 40.01 sekunde
Najbolji rezultat: 122.333

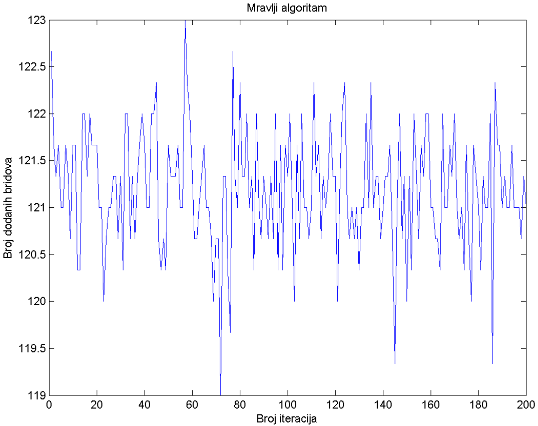
200 vrhova



Trajanje izvođenja: 100.11 sekundi

Najbolji rezultat: 118.33

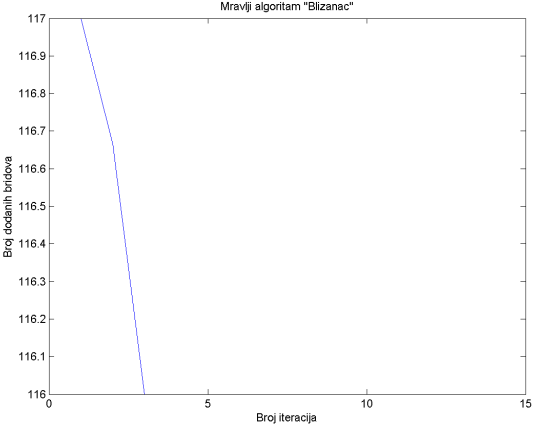
200 vrhova



Trajanje izvođenja: 24.41 sekunde

Najbolji rezultat: 118

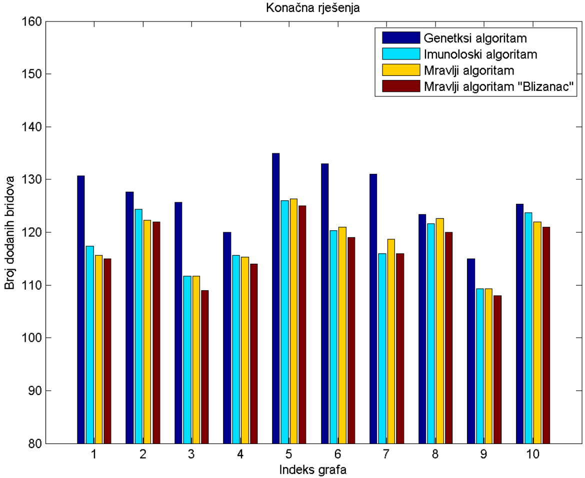
200 vrhova



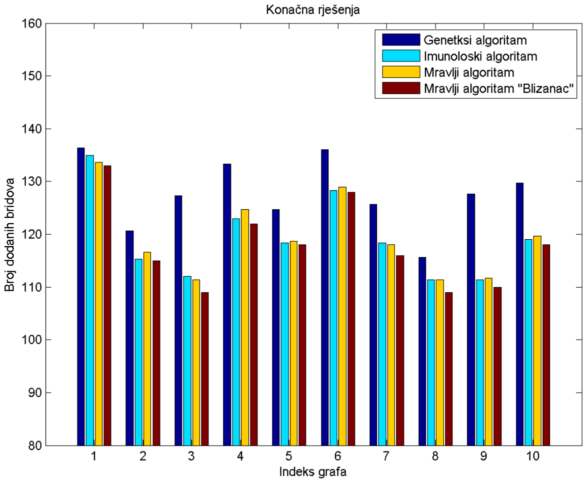
Trajanje izvođenja: 1.17 sekundi

Najbolji rezultat: 116

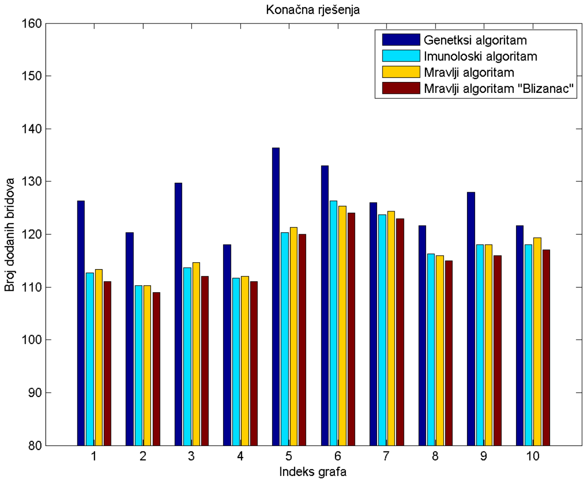
200 vrhova - usporedba rezultata



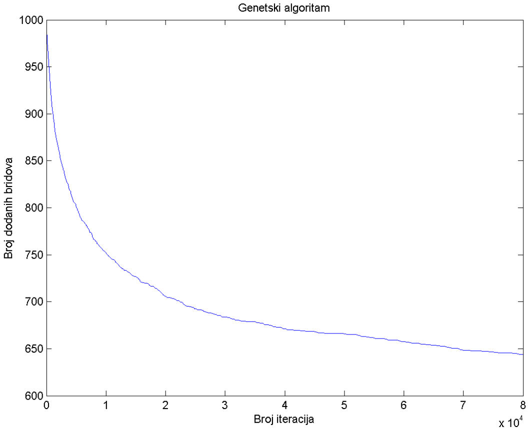
200 vrhova - usporedba rezultata



200 vrhova - usporedba rezultata



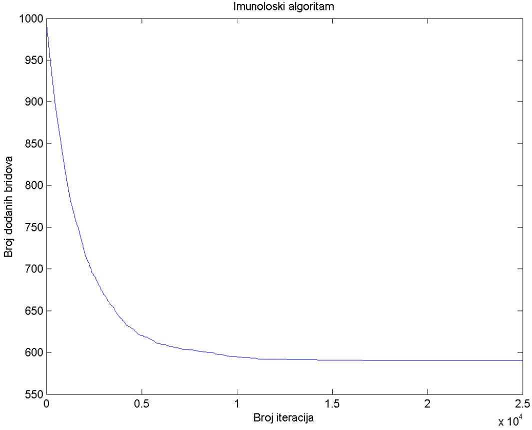
1000 vrhova



Trajanje izvođenja: 5136.6 sekundi

Najbolji rezultat: 644

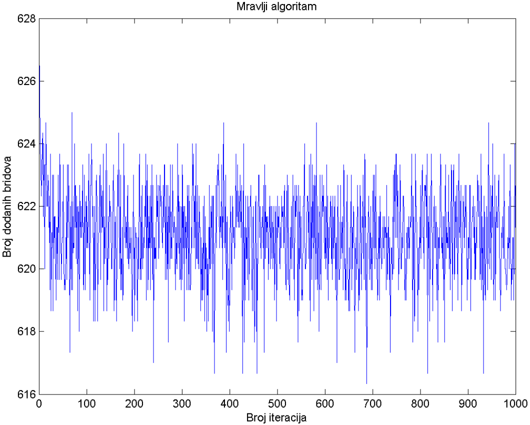
1000 vrhova



Trajanje izvođenja: 3045.6 sekundi

Najbolji rezultat: 590

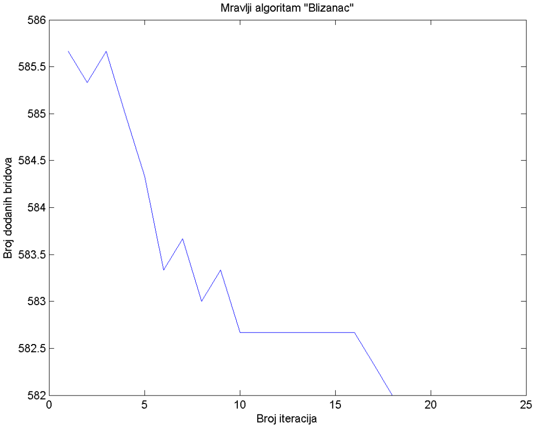
1000 vrhova



Trajanje izvođenja: 2399.5 sekunde

Najbolji rezultat: 611

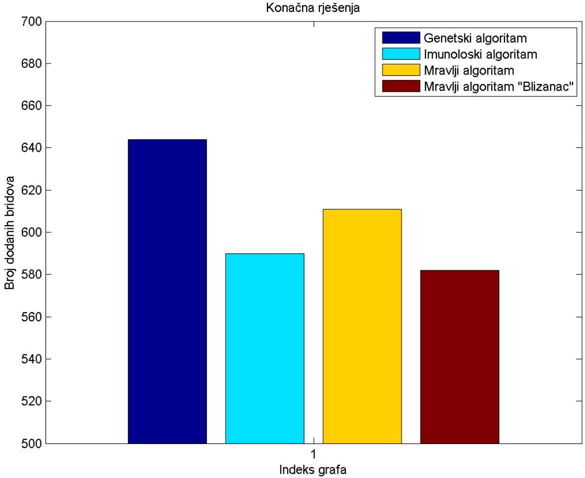
1000 vrhova



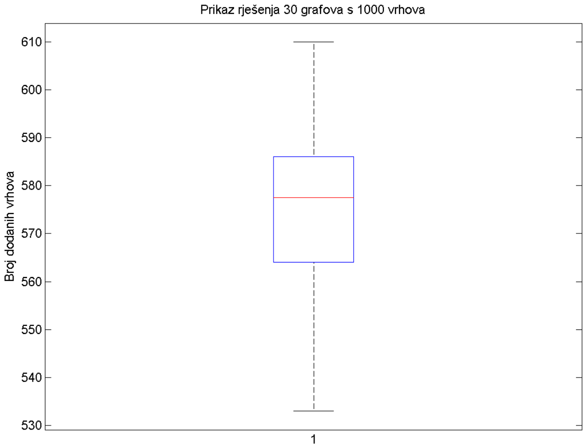
Trajanje izvođenja: 24.51 sekundi

Najbolji rezultat: 582

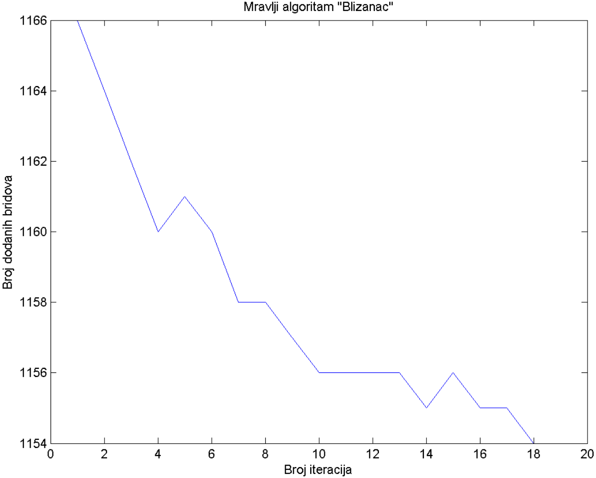
1000 vrhova - usporedba rezultata



Rezultati mravljeg algoritma "Blizanac"

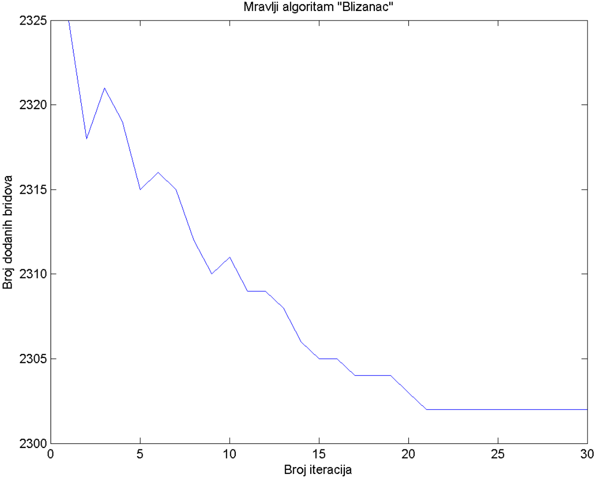


2000 vrhova



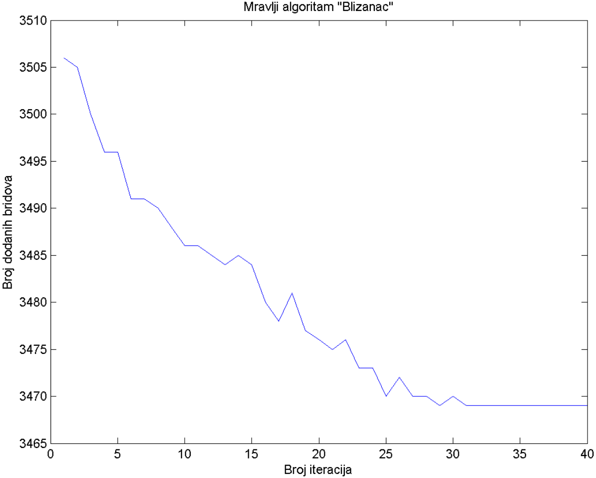
Trajanje izvođenja: 99.7 sekundi

4000 vrhova



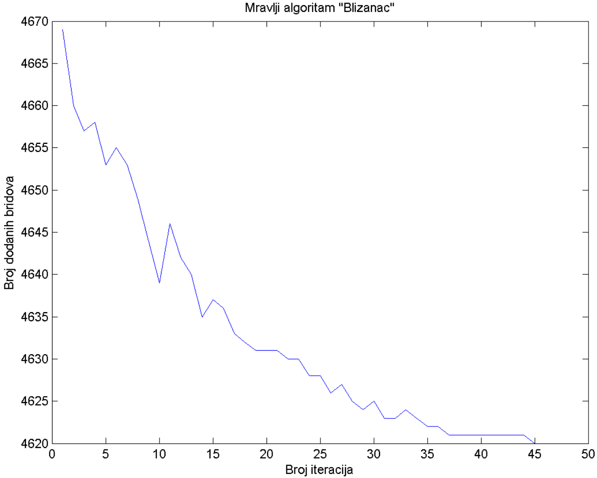
Trajanje izvođenja: 567.4 sekundi

6000 vrhova



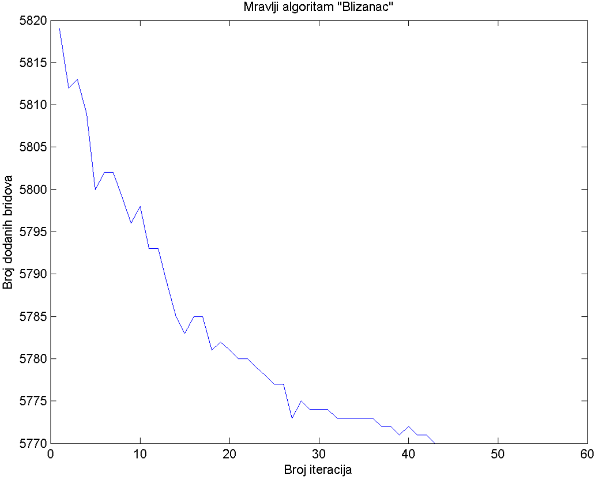
Trajanje izvođenja: 1786.2 sekunde

8000 vrhova



Trajanje izvođenja: 4147.5 sekunde

10000 vrhova



Trajanje izvođenja: 8796.2 sekunde

Zaključak

- Genetski algoritam se pokazuje presporim i nedovoljno točnim za pristup ovom problemu.
- Imunološki algoritam nudi blago poboljšanje u odnosu na genetski u aspektu brzine i osjetno poboljšanje u aspektu točnosti.
- Mravlji algoritam vrlo brzo pronađe solidno rješenje, ali vrlo teško konvergira u točnije.
- Daleko najbolje rezultate daje modifikacija mravljeg algoritma koja vrlo brzo konvergira u najbolja pronađena rješenja.

Zaključak

Daljnja poboljšanja:

- Implementacija vlastite sparse strukture prilagođene potrebama naših algoritama.
- Paralelizacija algoritama.
- Eventualno bolje prilagođeni operatori križanja za genetski algoritam.

Literatura

1. Darrell Whitley, Nam-Wook Yoo: Modeling Simple Genetic Algorithms for Permutation Problems, C.Sc. Department, Colorado State University
2. David Gamarnik, Maxim Sviridenko: Hamiltonian Completions of Sparse Random Graphs, MIT, 2012.
3. D. S. Franzblau, A.Raychaudhuri: Optimal Hamiltonian completions and path covers for trees, and a reduction to maximum flow; Department of Mathematics, CUNY, College of Staten Island, USA, 1999.
4. http://en.wikipedia.org/wiki/Hamiltonian_completion
5. Marko Čupić: Prirodom inspirirani optimizacijski algoritmi. Metaheuristike; FER, Sveučilište u Zagrebu
6. http://math.nist.gov/MatrixMarket/mmio/matlab/mmiomatlab.html