

Bojanje grafova prilagodljivim metaheurističkim postupcima

Diplomski rad

Dino Šantl

Mentor: Prof. dr. sc. Domagoj Jakobović

Fakultet elektrotehnike i računarstva

Zagreb, srpanj 2014.

Sadržaj

1. Uvod - definicija i analiza problema
2. Algoritmi i metode
3. Programsko rješenje
4. Rezultati
5. Zaključak

Uvod

- ▶ Korištenje metaheuristika u stvarnom problemu

Uvod

- ▶ Korištenje metaheuristika u stvarnom problemu
- ▶ Cilj: Povećanje kvalitete mobilne mreže

Uvod

- ▶ Korišćenje metaheuristika u stvarnom problemu
- ▶ Cilj: Povećanje kvalitete mobilne mreže
- ▶ *Scrambling Code* – prepoznavanje baznih stanica

Uvod

- ▶ Korištenje metaheuristika u stvarnom problemu
- ▶ Cilj: Povećanje kvalitete mobilne mreže
- ▶ *Scrambling Code* – prepoznavanje baznih stanica
- ▶ Problem: ~ 100000 baznih stanica i *samo* 512 kôdova

Uvod

- ▶ Korištenje metaheuristika u stvarnom problemu
- ▶ Cilj: Povećanje kvalitete mobilne mreže
- ▶ *Scrambling Code* – prepoznavanje baznih stanica
- ▶ Problem: ~ 100000 baznih stanica i *samo* 512 kôdova
- ▶ Pokušaj rješenja: uniformna raspodjela kodova - **neuspješno**

Modeliranje problema

- ▶ Bazne stanice modeliraju se čvorovima

Modeliranje problema

- ▶ Bazne stanice modeliraju se čvorovima
- ▶ Udaljenosti između baznih stanica – težinski bridovi

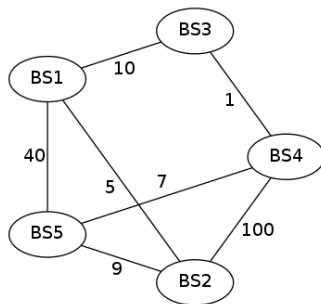
Modeliranje problema

- ▶ Bazne stanice modeliraju se čvorovima
- ▶ Udaljenosti između baznih stanica – težinski bridovi
- ▶ Model – težinski graf

Modeliranje problema

- ▶ Bazne stanice modeliraju se čvorovima
- ▶ Udaljenosti između baznih stanica – težinski bridovi
- ▶ Model – težinski graf
- ▶ Postavljanje kôdova - **problem bojanja težinskih grafova**

Slika: Primjer težinskog grafa



Definicija problema

- ▶ Ulaz: početno stanje mreže (informacije o čvorovima)

Definicija problema

- ▶ Ulaz: početno stanje mreže (informacije o čvorovima)
- ▶ Obojati graf s ciljem minimizacije sume težina konfliktnih bridova

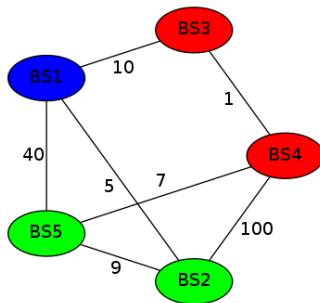
Definicija problema

- ▶ Ulaz: početno stanje mreže (informacije o čvorovima)
- ▶ Obojati graf s ciljem minimizacije sume težina konfliktnih bridova
- ▶ Najmanje 33% čvorova mora imati svoju početnu boju

Definicija problema

- ▶ Ulaz: početno stanje mreže (informacije o čvorovima)
- ▶ Obojati graf s ciljem minimizacije sume težina konfliktnih bridova
- ▶ Najmanje 33% čvorova mora imati svoju početnu boju

Slika: Primjer bojanja težinskog grafa - greška: 10



Analiza problema

- ▶ Za broj stanja (različitih bojanja): $N \leq \sum_{i=1}^M S(V(G), i)$

Analiza problema

- ▶ Za broj stanja (različitih bojanja): $N \leq \sum_{i=1}^M S(V(G), i)$
- ▶ Za $M = V(G)$ vrijedi: $N = B(M)$
 - ▶ $2^M \leq B(M) \leq M!$

Analiza problema

- ▶ Za broj stanja (različitih bojanja): $N \leq \sum_{i=1}^M S(V(G), i)$
- ▶ Za $M = V(G)$ vrijedi: $N = B(M)$
 - ▶ $2^M \leq B(M) \leq M!$
- ▶ Broj stanja prevelik za iscrpnu pretragu!

Analiza problema

- ▶ Optimizacijski problem (nije problem odluke)

Analiza problema

- ▶ Optimizacijski problem (nije problem odluke)
- ▶ Problem je **NP-težak**

Analiza problema

- ▶ Optimizacijski problem (nije problem odluke)
- ▶ Problem je **NP-težak**
- ▶ *No free lunch* teorem?

Analiza problema

- ▶ Optimizacijski problem (nije problem odluke)
- ▶ Problem je **NP-težak**
- ▶ *No free lunch* teorem?
- ▶ Nameće se korištenje (meta)heuristika

Korišteni algoritmi

- ▶ Pohlepni algoritam

Korišteni algoritmi

- ▶ Pohlepni algoritam
- ▶ Agentski algoritam

Korišteni algoritmi

- ▶ Pohlepni algoritam
- ▶ Agentski algoritam
- ▶ Simulirano kaljenje

Korišteni algoritmi

- ▶ Pohlepni algoritam
- ▶ Agentski algoritam
- ▶ Simulirano kaljenje
- ▶ Evolucijska strategija

Korišteni algoritmi

- ▶ Pohlepni algoritam
- ▶ Agentski algoritam
- ▶ Simulirano kaljenje
- ▶ Evolucijska strategija
- ▶ Genetsko kaljenje

Pohlepni algoritam

- Obilazak čvorova i dodjela boja

Pohlepni algoritam

- ▶ Obilazak čvorova i dodjela boja
- ▶ Pronalazak relativno dobre vrijednosti funkcije cilja

Pohlepni algoritam

- ▶ Obilazak čvorova i dodjela boja
- ▶ Pronalazak relativno dobre vrijednosti funkcije cilja
- ▶ Vrste sortiranja: COL, FIT, LDO, SDO, SDOLDO, RND

Pohlepni algoritam

- ▶ Obilazak čvorova i dodjela boja
- ▶ Pronalazak relativno dobre vrijednosti funkcije cilja
- ▶ Vrste sortiranja: COL, FIT, LDO, SDO, SDOLDO, RND
- ▶ Vrste bojanja: ABW, MC, MF, SWAP, RND

Pohlepni algoritam

- ▶ Obilazak čvorova i dodjela boja
- ▶ Pronalazak relativno dobre vrijednosti funkcije cilja
- ▶ Vrste sortiranja: COL, FIT, LDO, SDO, SDOLDO, RND
- ▶ Vrste bojanja: ABW, MC, MF, SWAP, RND
- ▶ Pohlepan algoritam ne pronađe nužno globalni minimum

Agentski algoritam



- ▶ Agent - nalazi se na nekom čvoru i donosi odluku o novoj boji

Agentski algoritam



- ▶ Agent - nalazi se na nekom čvoru i donosi odluku o novoj boji
- ▶ Boduje se prema najvećoj težini konfliktnog brida

Agentski algoritam



- ▶ Agent - nalazi se na nekom čvoru i donosi odluku o novoj boji
- ▶ Boduje se prema najvećoj težini konfliktnog brida
- ▶ Sortiranje agenata prema bodovima

Agentski algoritam



- ▶ Agent - nalazi se na nekom čvoru i donosi odluku o novoj boji
- ▶ Boduje se prema najvećoj težini konfliktnog brida
- ▶ Sortiranje agenata prema bodovima
- ▶ Odluka za agenta:
 - ▶ Ostajanje na čvoru
 - ▶ Slučajni susjed
 - ▶ *Najteži* susjed

Simulirano kaljenje

- ▶ Dva rješenja: staro X i novo S

Simulirano kaljenje

- ▶ Dva rješenja: staro X i novo S
- ▶ Ako je novo bolje sigurno se prihvaća

Simulirano kaljenje

- ▶ Dva rješenja: staro X i novo S
- ▶ Ako je novo bolje sigurno se prihvaća
- ▶ Ako je rješenje lošije prihvaća se s određenom vjerojatnošću
 - ▶ $P(X = S) = e^{-\frac{f(S) - f(X)}{\tau}}$

Simulirano kaljenje

- ▶ Dva rješenja: staro X i novo S
- ▶ Ako je novo bolje sigurno se prihvaća
- ▶ Ako je rješenje lošije prihvaća se s određenom vjerojatnošću
 - ▶ $P(X = S) = e^{-\frac{f(S) - f(X)}{T}}$
- ▶ Temperatura T pada s vremenom

Evolucijska strategija

- ▶ Genetski algoritam – smisao operatora križanja? – loši rezultati

Evolucijska strategija

- ▶ Genetski algoritam – smisao operatora križanja? – loši rezultati
- ▶ Evolucijska strategija (μ, λ) – kreiraju se nova rješenja, stara se brišu

Evolucijska strategija

- ▶ Genetski algoritam – smisao operatora križanja? – loši rezultati
- ▶ Evolucijska strategija (μ, λ) – kreiraju se nova rješenja, stara se brišu
- ▶ Samo operator mutacije - vrste bojanja

Genetsko kaljenje

- ▶ Kombinacija simuliranog kaljenja i genetskog algoritma

Genetsko kaljenje

- ▶ Kombinacija simuliranog kaljenja i genetskog algoritma
- ▶ Komunikacija između *jedinki* modelira se zajedničkom energijom koja se ravnopravno podijeli

Genetsko kaljenje

- ▶ Kombinacija simuliranog kaljenja i genetskog algoritma
- ▶ Komunikacija između *jedinki* modelira se zajedničkom energijom koja se ravnopravno podijeli
- ▶ Energija: koliko rješenje može biti lošije, a da se prihvati

Tuneliranje i postotak bojanja

- ▶ *SWAP* vrsta bojanja

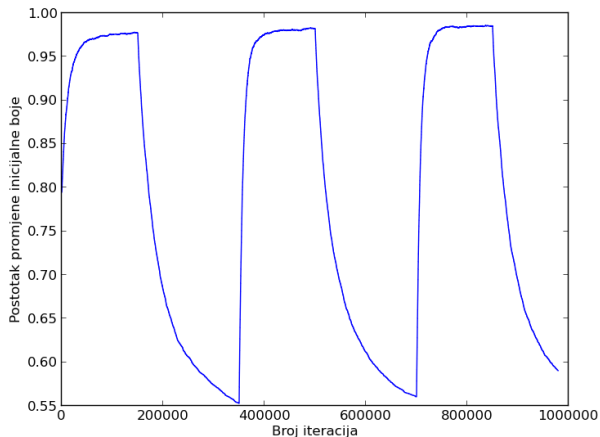
Tuneliranje i postotak bojanja

- ▶ *SWAP* vrsta bojanja
 - ▶ Odabire se točka u prostoru čija vrijednost funkcije cilja približno jednaka trenutnoj

Tuneliranje i postotak bojanja

- ▶ *SWAP* vrsta bojanja
 - ▶ Odabire se točka u prostoru čija vrijednost funkcije cilja približno jednaka trenutnoj
- ▶ Postotak bojanja - u operatore se ugrađuje preferiranje početne boje
 - ▶ Agresivni i blagi operatori

Minimizacija postotka promjene početne boje



Izlučivanje znanja

- ▶ Da li postoji funkcija $f(\text{COL}, \text{FIT}, \text{LDO}, \text{SDO})$ po kojoj se čvorovi sortiraju, a da je rezultat pohlepnog algoritma optimalan?

Izlučivanje znanja

- ▶ Da li postoji funkcija $f(\text{COL}, \text{FIT}, \text{LDO}, \text{SDO})$ po kojoj se čvorovi sortiraju, a da je rezultat pohlepnog algoritma optimalan?
- ▶ Slučajno sortirati čvorove

Izlučivanje znanja

- ▶ Da li postoji funkcija $f(\text{COL}, \text{FIT}, \text{LDO}, \text{SDO})$ po kojoj se čvorovi sortiraju, a da je rezultat pohlepnog algoritma optimalan?
- ▶ Slučajno sortirati čvorove
- ▶ Obojati nekom vrstom bojanja

Izlučivanje znanja

- ▶ Da li postoji funkcija $f(\text{COL}, \text{FIT}, \text{LDO}, \text{SDO})$ po kojoj se čvorovi sortiraju, a da je rezultat pohlepnog algoritma optimalan?
- ▶ Slučajno sortirati čvorove
- ▶ Obojati nekom vrstom bojanja
- ▶ Uzeti najbolje rezultate kao ulaz za regresiju

Izlučivanje znanja

- ▶ Da li postoji funkcija $f(\text{COL}, \text{FIT}, \text{LDO}, \text{SDO})$ po kojoj se čvorovi sortiraju, a da je rezultat pohlepnog algoritma optimalan?
- ▶ Slučajno sortirati čvorove
- ▶ Obojati nekom vrstom bojanja
- ▶ Uzeti najbolje rezultate kao ulaz za regresiju
- ▶ Korištenje regresije (odabir modela ili simbolička regresija)

Izlučivanje znanja

- ▶ Da li postoji funkcija $f(\text{COL}, \text{FIT}, \text{LDO}, \text{SDO})$ po kojoj se čvorovi sortiraju, a da je rezultat pohlepnog algoritma optimalan?
- ▶ Slučajno sortirati čvorove
- ▶ Obojati nekom vrstom bojanja
- ▶ Uzeti najbolje rezultate kao ulaz za regresiju
- ▶ Korištenje regresije (odabir modela ili simbolička regresija)
- ▶ Fokus je na čvorovima (ne na grafu)

Izlučivanje znanja

- ▶ Rezultati nisu pokazali postojanje funkcije koja kombinira vrste sortiranja
- ▶ Najveći utjecaj ima *SDO*, a nakon toga *LDO* – što se poklapa s prijašnjim istraživanjima
- ▶ Ako se greška minimizira lokalno *MF*, tada i *FIT* ima utjecaj

Programsko rješenje

- ▶ *Java* programski jezik

Programsko rješenje

- ▶ *Java* programski jezik
- ▶ Programsko okruženje za razvoj algoritama za ovaj specifičan problem

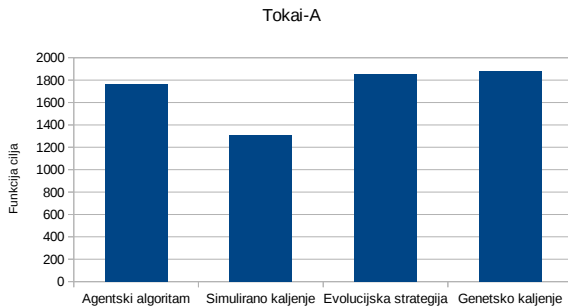
Programsko rješenje

- ▶ *Java* programski jezik
- ▶ Programsko okruženje za razvoj algoritama za ovaj specifičan problem
- ▶ Razni formati za čitanje grafova i spremanje rezultata, podrška za populacijske algoritme...

Minimizacija funkcije cilja

Graf	Funkcija cilja	Postotak promjene početne boje
Tokai	1195.255	65%
Kansai	4104.29	66%

Usporedba algoritama



Usporedba algoritama

Tablica: Standardna devijacija

Algoritam	Standardna devijacija
Agentski algoritam	166.5007100085
Simulirano kaljenje	2.3047581941
Evolucijska strategija	7.855956339
Genetsko kaljenje	0.1745653573

Zaključak

- ▶ Problem se može efikasno rješavati metaheuristikama

Zaključak

- ▶ Problem se može efikasno rješavati metaheuristikama
- ▶ Važno je ugraditi specifične operatore i namjestiti parametre

Zaključak

- ▶ Problem se može efikasno rješavati metaheuristikama
- ▶ Važno je ugraditi specifične operatore i namjestiti parametre
- ▶ Rezultati pokazuju da se ovakav pristup može koristiti za klasičan problem bojanja grafova

Zaključak

- ▶ Problem se može efikasno rješavati metaheuristikama
- ▶ Važno je ugraditi specifične operatore i namjestiti parametre
- ▶ Rezultati pokazuju da se ovakav pristup može koristiti za klasičan problem bojanja grafova
- ▶ Moguća poboljšanja i ubrzanja

Hvala na pažnji!

Pitanja?

