

Maximum clique korišćenjem Tabu pretrage



Projekat iz Računarske inteligencije
Matematički fakultet
Univerzitet u Beograd

Jovan Đorđević 164/17
Tamara Tomić 122/2017

SADRŽAJ



- 1) Opis problema
- 2) Brute force implementacija
- 3) AMTS implementacija
- 4) Testiranje i rezultati





Opis problema

Neka je $G = (V, E)$ neusmeren graf, gde je $V = \{1, \dots, n\}$ skup čvorova i E skup grana grafa. Klik C grafa G je podskup V takav da su svaka dva čvora povezana. Klik C je makismalan ako nije sadržan ni u jednom drugom kliku, odnosno ako je njegova kardinalnost veća od kardinalnosti svih drugih klika. Ovaj problem je NP-težak problem, koji ima primenu u rešavanju mnogih svakodnevnih problema.

Brute force implementacija



- Za implementaciju algoritma grube sile koristili smo rekurzivan pristup. Funkcija `isClique` proverava da li je trenutni podskup klik veličine `cliqueSize`, koja se prosleđuje kao argument funkcije. Ukoliko u bilo kom trenutku nađemo dva čvora koja nisu povezana vraća 0, inače vraća 1.
- Funkcija `maxCliqueBF` pokušava da doda neki čvor u trenutni podskup tako da to i dalje bude klika. Zatim rekurzivno poziva `maxCliqueBF` sa argumentima: novi čvor od kog krećemo i nova veličina (za jedan veća od trenutno najveće poznate.)

AMTS implementacija



- Adaptive multistart tabu search : ideja je da pokrećemo tabu pretragu više puta iz različitih početnih stanja i da je restartujemo kada se zaglavi. Algoritam se sastoji iz sledećih delova:
- Konstrukcija inicijalnog rešenja
- Generisanje ograničene okoline
- Odabir poteza
- Izvršavanje



Konstrukcija inicijalnog rešenja



- Polazimo od praznog skupa S i u svakom koraku dodajemo čvor $nodeToAdd$, koji se nalazi u skupu $notInS$, a ne nalazi se u skupu S (koji sadrži trenutnog kandidata za najveću kliku). Čvor koji dodajemo biramo tako da ima što manju frekvenaciju pojavljivanja u S , a što veći stepen povezanosti sa čvorovima koji se nalaze u S .
- Svaki put kada nekog kandidata ubacimo u S potrebno je da ažuriramo njegovu frekvenciju pojavljivanja (uvećati za 1) i stepen povezanosti, kao i stepen povezanosti čvorova koji nisu u S sa novim skupom S .

Generisanje ograničene okoline



- Ideja je da izlistamo sve moguće kombinacije poteza pri čemu potez predstavlja zamenu čvora u iz S čvorom v koji nije u S. Želimo da izbacimo čvor sa najmanjim stepenom povezanosti sa ostalim čvorovima u S i da ubacimo čvor sa najvećim stepenom povezanosti sa čvorovima u S. Nijedan od ovih čvorova ne sme se nalaziti u tabu listi, koja nam govori o tome koja stanja (čvorovi) su trenutno zabranjeni za pomeranje.
- Sve moguće ovakve zamene čine našu ograničenu okolinu.



Odabir poteza



- Kada generišemo ograničenu okolinu potrebno je da izaberemo određeni potez iz nje. Ukoliko okolina nije prazna nasumično bираmo jedan od poteza. U suprotnom sa malom verovatnoćom bираmo (mnogo) lošiji potez. Kada imamo novog kandidata proveramo da li je on validna k-klika i ukoliko jeste vraćamo taj skup i broj iteracija koji nam je bio potreban da ga pronademo.



Izvršavanje



- Za svaki graf pokrećemo AMTS pet puta u pokušaju da nađemo maksimalnu kliku. Najveći dozvoljeni broj iteracija je 10^5 . Maksimalna dubina pretrage L (broj iteracija u kojim se nije popravila funkcija evaluacije) jednaka je broju čvorova u grafu. Cilj nam je da nađemo skup S koji dostiže maksimalnu vrednost funkcije evaluacije f tako da važi: $f(S) = k \cdot (k-1) / 2$, gde je k veličina klike. Funkcija evaluacije f jednaka je broju grana koje postoje u podskupu S .

Testiranje i rezultati

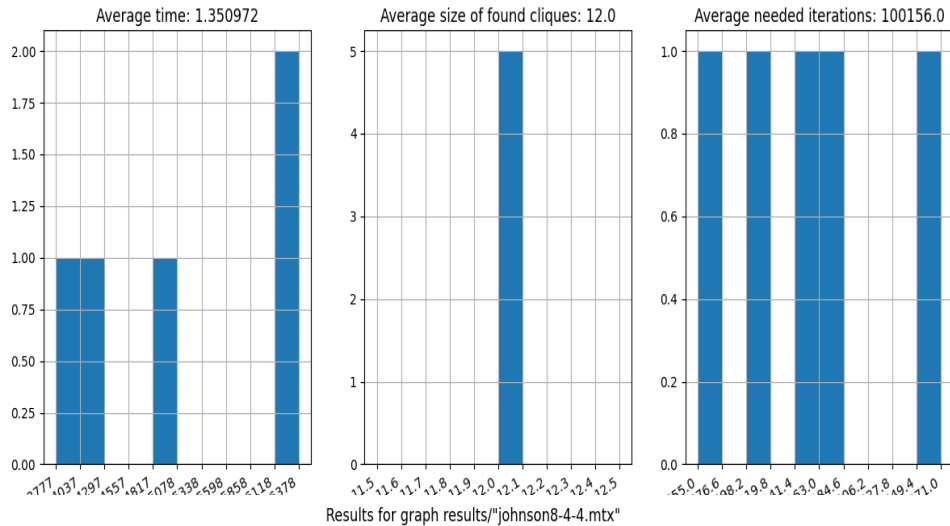


- Brute force algoritam testirali smo na 7 instanci, od čega smo za četiri instance dobili rešenje za neko realno vreme izvršavanja.
- Testiranje AMTS algoritma izvršeno je na 76/80 DIMACS test instanci. Naša verzija pokušava da nađe klike velične 3 pa nadalje do neuspeha. Za 54/76 grafova izvršavanje je uspešno i nađena je tražena veličina klike. Za 3/22 preostala grafa algoritam u nekom od pokretanja uspeva da nađe adekvatnu kliku. Za 11/22 instance ponovo smo pokrenuli algoritam sa dozvoljnih 10^8 iteracija. Time smo za 10/11 instanci barem u nekoj iteraciji našli očekivanu veličinu, dok samo za jednu instancu ovo povećanje nije bilo dovoljno da se stigne do traženog rezultata.

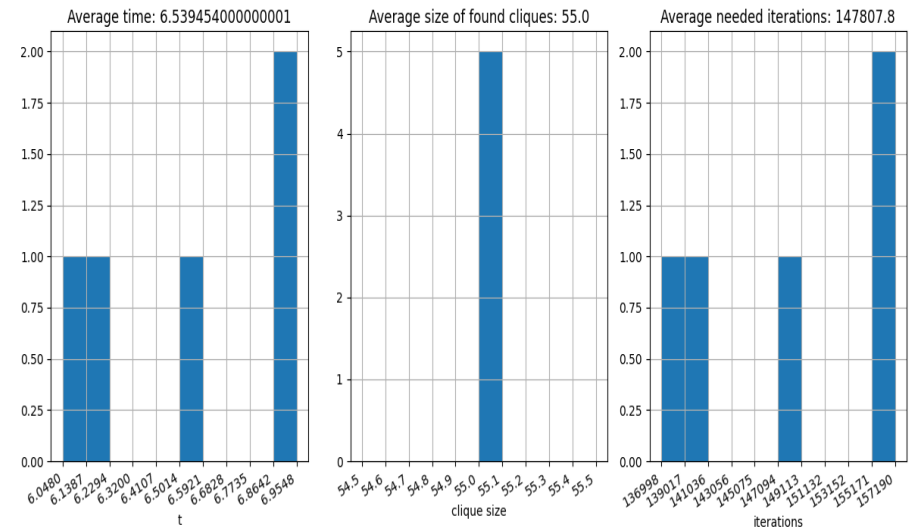
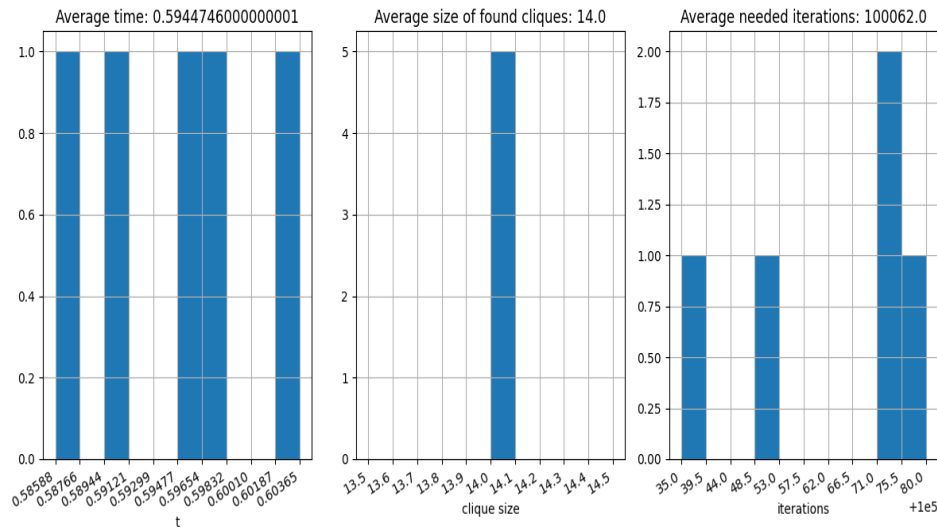
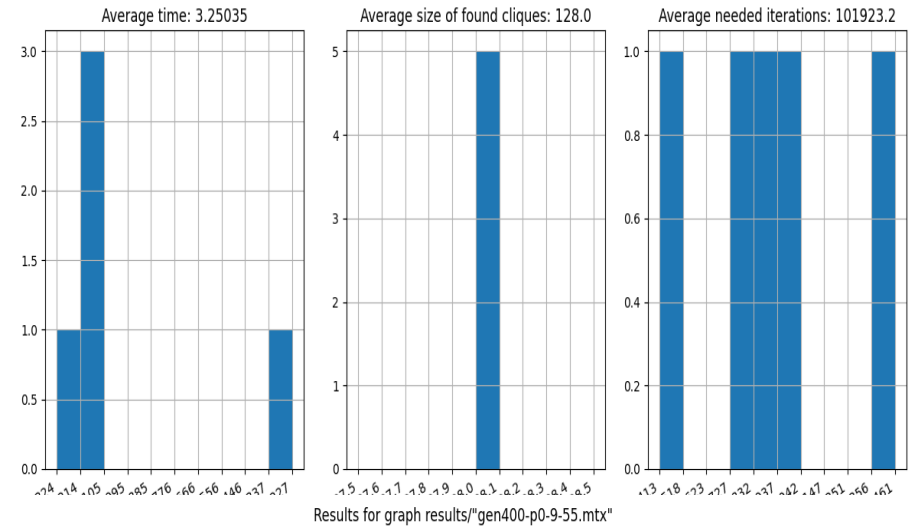
Prikaz rezultat



Results for graph results/"c-fat200-1.mtx"



Results for graph results/"hamming8-2.mtx"





KRAJ

