SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

Navarrov algoritam za približno uspoređivanje teksta

Dario Pavlović, Matej Lopotar
Voditelj: *Krešimir Križanović*

Zagreb, svibanj 2023.

SADRŽAJ

1. Uvod

U današnje vrijeme problem uspoređivanja vrlo je čest i pojavljuje se u mnogim područjima kao što su pretraživanje teksta, prepoznavanje uzoraka i biologija. Postoji mnogo algoritama koji se bave navedenom problematikom, a jedan od njih je i Navarrov algoritam za približno uspoređivanje teksta koji će biti opisan u ovom radu. Cilj rada je opisati i usporediti Navarrov algoritam s *bit parallel sequence-to-graph alignment* algoritmom. To znači usporediti vrijeme izvođenja te potrošnju memorije za vrijeme izvođenja algoritma. Testovi će se provoditi na četiri vrste grafova: *linear graph, SNP graph, twopath graph i tangle graph*. Navedeni grafovi se razlikuju u strukturi pa ćemo tako provjeriti kako sama struktura grafa utječe na Navarorov algoritam.

1.1. Sadržaj rada

Rad govori o problemu aproksimativnom pronalasku sekvence teksta u kontekstu hypertexta, gdje je tekst reprezentiran kao graf s čvorovima koji sadrže tekst i stranicama koje sadrže alternativne putove. Cilj je pronaći najmanju udaljenost nekog uzorka sa specificiranim ograničenjem danog uzorka, gdje najmanju udaljenost predstavlja najmanji broj operacija potreban za izjednačavanjem neka dva uzorka.

Trenutno klasičan način rješavanja ovog problema je bazirano na dinamičkom programiranju gdje je onda vremenska kompleksnost $O(mn)$, m označava duljinu uzorka, a n duljinu teksta. Dodatno predloženo je još algoritama za poboljšanje kompleksnosti vremena na $O(kn)$, gdje je k maksimalan broj dozvoljenih pogrešaka.

No rad uvodi algoritam koji poboljšava i vremensku i memorijsku kompleksnost za problem aproksimativnog pronalaska sekvence u *hypertextu*, autor predlaže pristup klasični pristup dinamičkog programiranja gdje *hypertext* smatramo kao graf. Predlaže stvaranje redne dinamičke matrice, tako da ažuriramo vrijednost svakog čvora u grafu na svakom koraku iteracije nad uzorkom. Taj algoritam onda ima vremensku kompleksnost $O(m(n + e))$ i traži $O(n)$ dodatno prostora za cikličke i acikličke grafove.

U radu se diskutira o generaliziranju algoritma, tako da za čvorove možemo uzeti i

više znakova i dozvoljavamo različite vrste operacija. Demonstrirana je ta adaptivnost algoritma na više varijacija dok vremenska kompleksnost ostaje ista.

2. Navarrov algoritam

```
Search (V,E,patt)
1.      for all v∈V, C_v ← 0.
2.      for i=1 to m
3.         for all v∈V, C'_v ← g(v,i)
4.         for all v∈V, C_v ← C'_v
5.         for all (u,v)∈E, Propagate (u,v)

Propagate (u,v)
     if C_v > 1 + C_u
         C_v ← 1 + C_u
         for all z/(v,z)∈E
             Propagate (v,z)
```

Slika 2.1: Gonzalo-Navarro algoritam [3]

Na slici 2.1 prikazan je pseudokod Navarrovog algoritma. Prvi korak algoritma je inicijalizacija vrijednosti Cv koju radimo tako da za svaki čvor v u skupu čvorova V postavimo početnu vrijednost C_v na 0. C_v vrijednost predstavlja najmanju udaljenost između uzorka i poduzorka koji završava na čvoru v u hipertekstu. Ta udaljenost zove se i *Levenshteinova udaljenost* i ona predstavlja najmanji broj potrebnih operacija umetanja, brisanja i substitucije potrebnih da A i B nizovi budu jednaki. [2] Drugi korak je iteracija kroz sve znakove uzorka (od 1 do m). Zatim ažuriramo vrijednosti C'_v pomoću funkcije g koja prima dva parametra - čvor i redni broj elementa uzorka.

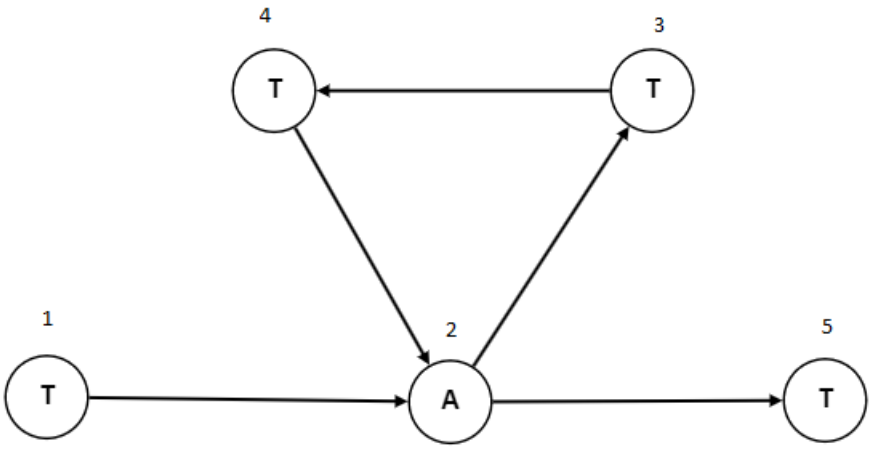
$$g(v,i) = \text{if } (patt[i] = t[v]) \text{ then } \min(\{C_u/(u,v)\in E\} \cup \{i-1\})$$
$$\text{else } 1 + \min\left(C_v, \min_{(u,v)\in E} C_u\right),$$

Slika 2.2: g gunkcija [3]

Svrha g funkcije je odrediti minimalnu udaljenost između uzorka $patt[1...i]$ i svakog podniza teksta koji završava na čvoru v. Nakon što je C'_v vrijednost izračunata potrebno je ažurirati C_v vrijednosti tako da im se dodijeli odgovarajuća C'_v vrijednost. U 5. liniji iterira se kroz sve bridove (u,v) u skupu bridova E te se da svaki brid poziva funkcija $Propagate$. U toj funkciji provjerava se je li Cv vrijednost veća od 1 + C_u gdje C_u predstavlja vrijednost izvorišnog čvora. Ako je provjera istinita C_v vrijednost se ažurira. Nakon toga rekurzivno se poziva $Propagate$ funkcija za svaki brid (v,z) koji polazi iz čvora v što nam omogućuje da se ažuriranja prošire kroz graf i da utječu na "djecu".

Važno je napomenuti da je funkcionalnost Navarrovog algoritma ovisna o topološkom poretku grafa G. Kako bi se riješio potencijalni problem implementiran je Kahnov algoritam [1] za pronalazak topološkog poretka.

Uzmimo da je p uzorak koji tražimo i G graf po kojem vršimo pretragu.



Slika 2.3: Usmjereni graf G

Na slici 2.3 prikazan je usmjereni graf G na kojem želimo pronaći uzorak $p = TTTT$. Prije prve iteracije sve C_v vrijednosti postavljene su na 0. U prvoj iteraciji tražimo znak T i za svaki čvor gledamo završava li on u tom znaku. Ako završava onda C_v vrijednost ostaje nepromijenjena (ostaje 0) jer je udaljenost 0, a ako ne završava onda se C_v vrijednost postavlja na 1. Na kraju prve iteracije čvorovi 1, 3, 4 i 5 imaju $C_v = 0$ dok čvor 2 ima $C_v = 1$. U drugoj iteraciji tražimo podniz TT.

Samo se C_v vrijednost čvora 4 neće promijeniti i ostat će na 0 jer je moguće dobiti podniz TT tako da završimo u čvoru 4. Ostali čvorovi imat će C_v vrijednost iznosa 1. U trećoj iteraciji tražimo podniz TTT. C_v vrijednost čvora 1 tada raste na 2 i C_v vrijednost čvora 4 raste na 1. Ostalim čvorovima C_v vrijednost ostaje nepromijenjena. U zadnjoj, četvrtoj iteraciji tražimo cijeli niz $TTTT$. U tom se slučaju C_v vrijednost prva 3 čvora povećava za 1 pa vrijednosti redom iznose: 3, 2, 2. Čvorovi 4 i 5 imaju na kraju $C_v = 1$. Važno je napomenuti da se računa C'_v vrijednost koja se na kraju iteracije sprema u C_v.

3. Rezultati

Vrsta grafa	Navarro memorija (KB)	Navarro vrijeme (s)	Bit-parallel memorija (KB)	Bit-parallel vrijeme (s)
Linear	$1.14 * 10^7$	68.23	1844.98	3.18
Twopath	$1.10 * 10^7$	902.53	18413.10	52.63
SNP	$1.08 * 10^7$	170.52	4191.48	6.06
Tangle	$1.08 * 10^7$	114.61	1992.72	32.87

Tablica 3.1: Tablični prikaz rezultata

Tablica 3.1 prikazuje rezultate dobivene za 4 vrste grafova za Navarrov algoritam i bit prallel sequence-to-graph algoritam. Promatrane su potrošnja memorije i vrijeme izvođenja pojedinih algoritama. Iz rezultata je vidljivo da je twopath graf najkompleksniji i da je za njega potrebno najviše vremena. Općenito, iz tablice je vidljivo da je Navarrov algoritam je puno sporiji od bit parallel sequence-to-graph algoritma za svaki graf, a i potrebno mu je puno više memorije za izvođenje.

4. Zaključak

Cilj rada je opisati i usporediti Navarrov algoritam s *bit parallel sequence-to-graph alignment* algoritmom. To znači usporediti vrijeme izvođenja te potrošnju memorije za vrijeme izvođenja algoritma. Memorijski i vremenski testovi provedeni su za 4 vrste grafova, a rezultati su prikazani u tablici 3.1. Iz dobivenih rezultata dolazimo do zaključka da je naša implementacija Navarrovog algoritma puno sporija od bit parallel sequence-to-graph algoritma i da zahtijeva puno više memorije. Iako su oba algoritma imala savršenu točnost (100%) zaključujemo da je bit parallel sequence-to-graph algoritam memorijski i vremenski optimalniji za korištenje i pronalazak uzoraka.

Naš cilj je uspješno izvršen - uspjeli smo implementirati Navarrov algoritam, izračunati vrijeme izvođenja algoritma i potrebnu memoriju za izvedbu na sve 4 vrste grafova. Što se tiče rezultata, možemo reći da je očekivano da bit parallel sequence-to-graph radi bolje od naše implementacije Navarrovog algoritma.

5. Literatura

[1] URL https://www.geeksforgeeks.org/topological-sorting-indegree-based-solution/.

[2] Heikki Hyyrö, Kimmo Fredriksson, i Gonzalo Navarro. Increased bit-parallelism for approximate and multiple string matching. *ACM J. Exp. Algorithmics*, 10: 2.6–es, dec 2005. ISSN 1084-6654. doi: 10.1145/1064546.1180617. URL https://doi.org/10.1145/1064546.1180617.

[3] Gonzalo Navarro. Improved approximate pattern matching on hypertext. *Theoretical Computer Science*, 237(1):455–463, 2000. ISSN 0304-3975. doi: https://doi.org/10.1016/S0304-3975(99)00333-3. URL https://www.sciencedirect.com/science/article/pii/S0304397599003333.

[3] [1] [2]

6. Sažetak

Navarrov algoritam je algoritam koji služi za uspoređivanje teksta i moguće ga je koristiti u bioinformatici. Implementirani Navarrov algoritam uspoređen je s bit parallel sequence-to-graph algoritam. Testovi su provedeni na četiri različite vrste grafova: *linear, twopath, SNP i tangle*. Oba algoritma su imala točnost iznosa 100%, ali naša implementacija Navarrovog algoritma radila je puno sporije s većim utroškom memorije zbog čega dolazimo do zaključka da je bit parallel sequence-to-graph algoritam bolji za korištenje.