Problem Stabilnog Uparivanja

Seminarski rad u okviru kursa
Konstrukcija i analiza algoritama II
Matematički fakultet

Rastko Đorđević, 1091/2017
rastko_djordjevic@matf.bg.ac.rs

24. februar 2019

Sažetak

Problem stabilnog uparivanja *(eng. Stable matching problem / Stable marriage problem)* ima značaja u oblastima matematike, ekonomije i informatike. U ovom radu će biti opisan problem, algoritam za njegovo rešavanje, i implementacija datog algoritma u programskom jeziku Python.

Sadržaj

1 Postavka problema

Kod problema stabilnog uparivanja zadatak je pronaći stabilno uparivanje između dva skupa elemenata. Za svaki element oba skupa data je lista preferenci elemenata iz suprotnog skupa. Za uparivanje kažemo da nije stabilno ako važi sledeće:

1. Postoji element a skupa A koji preferira element b skupa B više od svog trenutnog para
2. Element b takođe preferira element a više od svog trenutnog para

Ovaj problem se često formuliše kao problem stabilnog braka na sledeći način: Za datih n muškaraca i n žena gde je svaka osoba ocenila sve osobe suprotnog pola, treba venčati sve muškarce i žene tako da ne postoje 2 osobe suprotnog pola koje bi radije napustile svoje trenutne partnere i oženili se jedno sa drugim. Kada ne postoji takav par osoba za skup brakova kažemo da je stabilan.

2 Gejl-Šepli algoritam

David Gejl *(eng. David Gale)* i Lojd Šepli (Lloyd Shapley) su 1962. godine dokazali da je za jednak broj muškaraca i žena uvek moguće rešiti problem stabilnog braka tako da sve osobe budu u braku i da takvi brakovi budu stabilni[2]. Njihov algoritam se sastoji od niza rundi:

U prvoj rundi svaki muškarac koji nije veren prosi ženu koju najviše voli. Potom svaka žena prihvata veridbu muškarca kojeg najviše voli od onih koji su je zaprosili, a ostale odbija.

U svakoj sledećoj rundi se dešava sledeće: Prvo svaki slobodan muškarac prosi ženu koju najviše voli od onih koje još nije prosio. Potom svaka žena koja nije verena prihvata veridbu muškarca kojeg najviše voli. Sve verene žene prihvataju veridbu samo ako više vole muškarca koji ih je zaprosio od trenutnog verenika i u tom slučaju ta veridba se raskida i taj verenik postaje slobodan.

Ovaj proces se ponavlja dok svi nisu vereni. Složenost algoritma je $O(n^2)$ gde je n broj muškaraca, što su pokazali Ivama *(eng. Iwama)* i Mijazaki *(eng. Miyazaki)*[3].

Dati algoritam garantuje da će svi biti u braku. Pretpostavimo da postoje muškarac a i žena b koji nisu u braku na kraju algoritma. Muškarac je zaprosio sve žene pa samim tim i ženu b. Pošto a i b nisu u braku to znači da žena b ili nije prihvatila njegovu prosidbu zbog nekog drugog muškarca, ili je prihvatila prosidbu pa je naknadno ostavila muškarca b zbog nekog drugog muškarca. U oba slučaja žena b mora biti u braku, što je suprotno od početne pretpostavke da žena b nije u braku.

Algoritam takođe garantuje da će su svi brakovi na kraju algoritma stabilni. Pretpostavimo da postoje muškarac a i žena b koji su u braku sa nekim drugim osobama i koji više vole jedno drugo od svojih trenutnih partnera. Pošto muškarac a više voli ženu b od svoje trenutne partnerke to znači da je zaprosio ženu b pre nego što je zaprosio svoju partnerku. Pošto je ženu b zaprosio muškarac a to znači da konačnog muškarca za kojeg se verila voli više od muškarca a što je u kontradikciji sa početnom pretpostavkom da žena b više voli muškarca a od svog trenutnog partnera.

Pseudo-kod algoritma:

```
Algoritam stabilno_uparivanje(M, Ž)
Ulaz:
M - skup muškaraca M, sa uređenom listom preferiranih žena
Ž - skup žena Ž, sa uređenom listom preferiranih muškaraca
Izlaz:
SU - stabilno uparivanje elemenata iz skupova M i Ž
begin
    Inicijalizuj sve muškace i žene da budu slobodni
    while postoji slobodan muškarac m koji još nije zaprosio sve žene
        ž = prva žena na listi preferenci muškarca m koju m još nije zaprosio
        if ž je slobodna
            upari(m, ž)
        else neki par (m', ž) već postoji
            if ž više voli m od m'
                m' postaje slobodan
                upari(m, ž)
end
```

2.1 Optimalnost rešenja

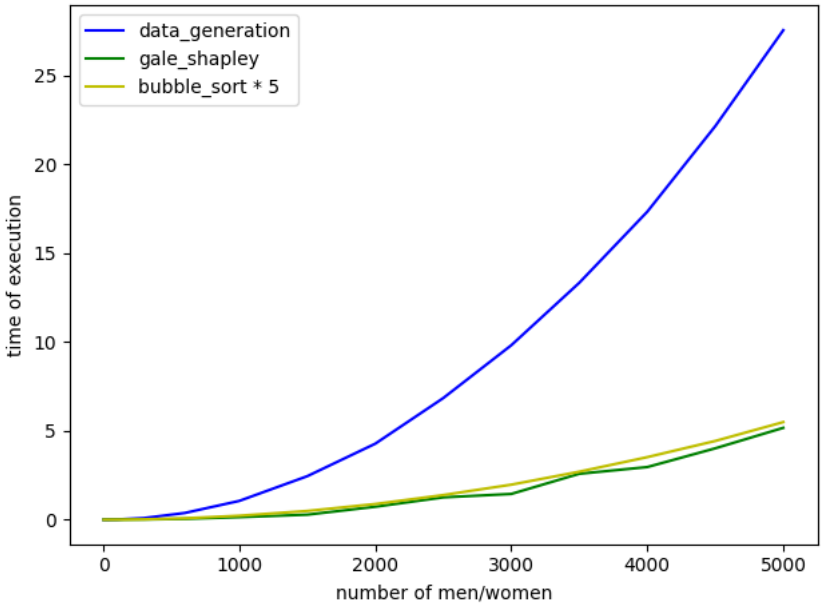
U opštem slučaju stabilno uparivanje nije jedinstveno stoga se postavlja pitanje kakvo rešenje vraća opisani algoritam? Da li je bolje za žene ili muškarce, ili je nešto između? Važi da su u prednosti osobe koje preuzmu inicijativu. Gejl-Šepli algoritam u kojem muškarci prose žene uvek vraća najbolje uparivanje za sve muškarce od svih mogućih stabilnih uparivanja. U suprotnom, ako bi žene prosile muškarce, onda bi rezultujuće uparivanje bilo najbolje od svih stabilnih uparivanja za sve žene. Optimalnost rešenja se može dokazati kontradikcijom.

Takođe treba razmotriti da li je algoritam moguće "prevariti"? Naime, ako bi neki muškarac znao liste prioriteta svih ostalih osoba, pitanje je da li bi mogao da promeni svoju listu prioriteta kako bi dobio voljenu ženu. Ispostavlja se da ovo nije moguće. Čak ni više muškaraca u koaliciji ne mogu da promene svoje preference tako da svi muškarci u koaliciji prođu bolje [1].

Sa druge strane, žene, to jest grupa koja prima prosidbe može "prevariti"algoritam. Tačnije, za žene važi da bi izmenom svoje liste preferenci mogle da dobiju boljeg muškarca nego da su bile iskrene tokom kreiranja svoje liste preferenci.

3 Implementacija

Algoritam je implementiran u programskom jeziku Python. Pošto je složenost algoritma kvadratna, algoritam je upoređen sa verzijom sortiranja mehurom *(eng. bubble sort)*. Kao što se može videti na grafu 1, izvršavanje algoritma je približno jednako izvršavanju sortiranja mehurom 5 puta. Takođe možemo primetiti da priprema podataka zahteva veliku količinu vremena i to približno 5 puta više vremena od izvršavanja samog algoritma. To je slučaj zbog izbora struktura listi preferenci, koje zahtevaju veliki broj mešanja i kopiranja ogromnih nizova. Ovo bi se moglo dodatno optimizovati ali tome nije dat ključni prioritet prilikom pisanja programa.



Slika 1: Poređenje efikasnosti

Iako optimizacija nije bila najveći prioritet, uzeta je u obzir pri implementaciji algoritma. Na maksimalnom broju mesta korišćeni konstrukti u pythonu koji se direktno prevode do C koda umesto konstrukta kod kojih to nije slučaj.

4 Primene

Algoritam za pronalazak stabilnog uparivanja se može primeniti u raznim situacijama. Najpoznatija primena je verovatno uparivanje diplomiranih studenata medicine i njihovih prvih bolnica.

Još jedna važna primena je uparivanje korisnika interneta i servera koji distribuiraju servise. Mreže za dostavu sadržaja *(eng. Content delivery networks)* koje čine ključni deo interneta izvršavaju ovaj algoritam svakih desetak sekundi kako bi omogućili svim korisnicima brz pristup željenom sadržaju.

4.1 Praktična pitanja

Razmatraćemo praktične probleme primene Gejl-Šepli algoritma nad problemom upisivanja studenata u univerzitete sa kojim su se i borili autori originalnog rada. Ovo je malo izmenjena verzija početnog problema, gde svaki element skupa A može biti uparen sa više elemenata skupa B. Rešenje ovog problema je veoma slično već opisanom algoritmu. Nas zanima koliko je ovaj algoritam zaista upotrebljiv za spajanje studenata sa željenim univerzitetima i obratno.

Prvi očigledan problem je što algoritam zahteva dosta komunikacije napred i nazad ako bi se pratio doslovno. Jednostavno rešenje bi bilo skupiti sve liste preferenci na jedno mesto i potom izvršiti algoritam nad svim

podacima. Problem sa ovim rešenjem je što bi bila neohodna ogromna količina vremena koji bi morao da uloži svaki od učesnika kako bi napravio iscrpnu listu preferenci svih univerziteta i obratno. Tako da bi se morao napraviti neki kompromis, na primer kandidati mogu poslati zahteve za samo svojih 5 omiljenih izbora.

Takođe je bitno napomenuti da nakon što je aplikant odbijen on ne mora odmah aplicirati za svoj sledeći univerzitet. Konačni izbor se neće promeniti ako aplikant preskoči narednih nekoliko rundi. Ovo znači da ceo proces neće morati da stane nakon što neki od studenata bude odbijen od svih 5 omiljenih univerziteta.

Pored tehničkih problema koji su do sada razmatrani, postoji i veliki broj problema vezanih za dodatne uslove koji nisu uzeti u obzir u algoritmu. Na primer neki studenti će pre izabrati određen univerzitet sa stipendijom ali im je previše skup bez stipendije. Neki univerziteti takođe uzimaju u obzir i ukupan sastav polazne generacije, stoga se njihove preference menjaju sa svakim primljenim studentom.

Iako se neki od pomenutih problema mogu lako rešiti izmenom algoritma, kod nekih od problema to nije slučaj. Uprkos tome što može doći do otežavajućih okolnosti pri primeni algoritma, on nam daje i neke bitne osobine. Aplikanti ne gube svoje niže izbore dok im se razmatraju omiljeni izbori. Takođe univerziteti su sigurni da aplikanti koji su na njihovoj listi čekanja nemaju druge univerzitete na koje bi radije otišli, stoga ako im se ukaže prilika prihvatiće ponudu da odu na dati univerzitet.

Literatura

[1] L. E. Dubins and D. A. Freedman. Machiavelli and the gale-shapley algorithm. *The American Mathematical Monthly*, 88(7):485–494, 1981.

[2] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[3] K. Iwama and S. Miyazaki. A survey of the stable marriage problem and its variants. In *International Conference on Informatics Education and Research for Knowledge-Circulating Society (icks 2008)*, pages 131–136, Jan 2008.