SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO-MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

PROBLEM STABILNOG BRAKA

EMPIRIJSKA I TEORIJSKA ANALIZA

Oblikovanje i analiza algoritama

Lea Markušić

Nositelj kolegija: doc. dr. sc. Matej Mihelčić

4.1.2024.

Sadržaj

1. Uvod

Problem stabilnog braka je osnovna verzija problema stabilnog sparivanja, gdje su skupovi koje treba spariti, u smislu heteroseksualnog braka, skup žena W={w_1, ... w_n} i skup muškaraca M={m_1, ... m_n}. Primijetimo da su skupovi jednakobrojni, *n* žena i *n* muškaraca. Svaka žena rangira muškarce po preferenciji sparivanja, na vrhu rang-liste nalazi se njoj najpoželjniji muškarac, dok je na dnu liste najmanje poželjan. Analogno tome, svaki muškarac rangira žene. Cilj je upariti žene i muškarce u *stabilno sparivanje*, odnosno tako da ako žena i muškarac nisu u paru, barem jedan od njih preferira svog pripadajućeg partnera.

1.1. Osnovni pojmovi

Navedimo nekoliko definicija koje će se koristiti u cijelom radu:

Definicija 1: **Sparivanje S** je skup uređenih parova iz W × M takvih da se svaki element $w_i \in W$ i svaki element $m_i \in M$ nalaze u točno jednom uređenom paru skupa S.

Definicija 2: Ako žena i muškarac nisu u paru, a oboje preferiraju jedno drugo više nego trenutnog partnera, tada oni čine **blokirajući par**, tj. **nestabilnost**.

Neka je S jedno sparivanje koje sadrži parove (w_1, m_1) i (w_2, m_2). Pretpostavimo da žena w_1 preferira m_2 nad m_1 i m_2 preferira w_1 nad w_2. Tada bi (w_1, m_2) činili nestabilnost, odnosno htjeli bi prekinuti zaruke sa trenutnim partnerima kako bi mogli biti zajedno u paru.

Definicija 3: **Stabilno sparivanje** je sparivanje bez nestabilnosti. Ako postoji nestabilnost, tada je to sparivanje **nestabilno**.

1.2. Uvodni primjeri

Pokažimo primjer stabilnog sparivanja.

Primjer 1: Neka su dani skup žena W = {Manda, Ema, Lucija, Ana} i skup muškaraca M = {Oskar, David, Branko, Hrvoje} sa rang-listama preferencija:

Manda: Branko, Hrvoje, Oskar, David
Ema: David, Hrvoje, Oskar, Branko
Lucija: Branko, David, Hrvoje, Oskar
Ana: Branko, Hrvoje, David, Oskar

Oskar: Manda, Ana, Lucija, Ema
David: Manda, Lucija, Ana, Ema
Branko: Lucija, Ana, Manda, Ema
Hrvoje: Lucija, Manda, Ema, Ana

Tada stabilno sparivanje čine parovi:

Lucija i Branko
Manda i Hrvoje
Ana i David

Ema i Oskar

Može se provjeriti da to sparivanje ne sadrži nestabilnosti. Na primjer, iako Ana preferira Branka i Hrvoja nad svojim partnerom Davidom, Branko preferira svoju partnericu Luciju nad Anom i Hrvoje preferira svoji partnericu Mandu nad Anom, tako da ni Ana i Branko ni Ana i Hrvoje nisu blokirajući par.

Preciznije, ovo stabilno sparivanje je jedinstveno.

Pretpostavimo da su zadnja dva para Ema i David i Ana i Oskar. Tada bi Ana i David bili blokirajući par zato što nisu upareni, a Ana preferira Davida nad Oskarom i David preferira Anu nad Emom. U tom slučaju sparivanje nebi bilo stabilno.

Stabilno sparivanje ne mora biti jedinstveno. Pokažimo to idućim primjerom.

Primjer 2: Neka su dani skup žena W = {Monica, Phoebe, Rachel} i skup muškaraca M = {Chandler, Joey, Ross} sa rang-listama preferencija:

Monica: Chandler, Joey, Ross
Phoebe: Joey, Ross, Chandler
Rachel: Ross, Chandler, Joey

Chandler: Phoebe, Rachel, Monica
Joey: Rachel, Monica, Phoebe
Ross: Monica, Phoebe, Rachel

Tada postoje čak tri stabilna sparivanja:

Sparivanje 1	Sparivanje 2	Sparivanje 3
Monica i Chandler	Phoebe i Chandler	Rachel i Chandler
Phoebe i Joey	Rachel i Joey	Monica i Joey
Rachel i Ross	Monica i Ross	Phoebe i Ross

U sparivanju 1 sve su žene uparene prema prvom odabiru, a svi muškarci dobivaju svoj posljednji odabir. Sparivanje 2 je suprotno, svi su muškarci upareni sa prvom ženom na svojoj rang-listi, a žene s posljednjim muškarcem sa svoje rang-liste. Kada svaka žena, odnosno svaki muškarac dobije svoj prvi izbor, tada ne mogu postojati nestabilnosti. U sparivanju 3 svatko je uparen sa svojim drugim izborom. Lako se pokaže da nema blokirajućeg para.

2. Gale – Shapley algoritam

Uvijek postoji stabilno sparivanje neovisno o rang-listi pojedine žene odnosno muškarca. Gale – Shapley algoritam uvijek pronalazi stabilno sparivanje.

Primjenom Gale – Shapley algoritma na problem stabilnog braka dobivamo dva suprotna pristupa problemu – „orijentiran prema ženama“ i „orijentiran prema muškarcima“. U ovome radu koristit će se algoritam orijentiran prema ženama, odnosno onaj u kojem žene biraju muškarce, redom sa svoje rang-liste preferencija.

2.1. Opis algoritma (primjena algoritma na problem stabilnog braka)

Svaka osoba je ili „slobodna“ ili „zaručena“. Na početku provođenja samog algoritma, sve osobe su slobodne. Osobe postaju zaručene kada slobodna žena zaprosi slobodnog muškarca te oni postaju par. Kada je muškarac prvi put zaručen, on ostaje zaručen do kraja algoritma, iako ne nužno za istu ženu. Ako zaručenog muškarca zaprosi žena koju preferira više nego ženu s kojom je trenutno u paru, tada raskida zaruke, žena s kojom je bio zaručen postaje ponovo slobodna, a muškarac ulazi u par s novom ženom (sada su oni zaručeni). Inače, ako tu, novu ženu preferira manje nad trenutnom, novu odbija i ostaje u paru s trenutnom. Svaka žena prosi muškarce sa svoje liste, redom, sve dok ne postane zaručena. Tek kada/ako žena ponovno postane slobodna, nastavlja sa prosidbom muškaraca sa liste, s prvim idućim muškarcem kojeg još nije zaprosila. Kada su svi zaručeni, spojeni u parove, algoritam završava i dobivamo jedno stabilno sparivanje.

2.2. Pseudokod implementacije algoritma

Slika 1 prikazuje pseudokod prethodno opisanog algoritma, Gale-Shapley algoritma orijentiranog prema ženama. U Teoremu 3 ćemo pokazati da, neovisno o redoslijedu odabira žena u liniji 2 ovog pseudokoda, uvijek dobivamo stabilno sparivanje. Za implementaciju analognog algoritma orijentiranog prema muškarcima potrebno je samo zamijeniti žene i muškarce, odnosno, sve dok postoji slobodan muškarac, prolazeći kroz njegovu rang listu preferencija, tražiti adekvatnu ženu.

```
GALE-SHAPLEY(men, women, rankings)
 1  assign each woman and man as free
 2  while some woman w is free
 3        let m be the first man on w's ranked list to whom she has not proposed
 4        if m is free
 5              w and m become engaged to each other (and not free)
 6        elseif m ranks w higher than the woman w' he is currently engaged to
 7              m breaks the engagement to w', who becomes free
 8              w and m become engaged to each other (and not free)
 9        else m rejects w, with w remaining free
10  return the stable matching consisting of the engaged pairs
```

Slika 1: Pseudokod

Provedimo sada algoritam gornjim pseudokodom na Primjeru 1. Na početku su sve osobe slobodne.

1. korak: Manda zaprosi Branka. Budući da je Branko slobodan, Manda i Branko postaju par i više nisu slobodni.
2. korak: Ema zaprosi Davida. Budući da je David slobodan, Ema i David postaju par i više nisu slobodni.
3. korak: Lucija zaprosi Branka. Iako je Branko zaručen za Mandu, više preferira Luciju pa prekida zaruke sa Mandom, koja postaje slobodna. Sada su Lucija i Branko par, a Lucija više nije slobodna.
4. korak: Ana zaprosi Branka. Budući da je Branko zaručen za Luciju, koju preferira nad Anom, odbija Aninu prosidbu i ostaje zaručen za Luciju. Ana ostaje slobodna.
5. korak: Ana zaprosi Hrvoja. Budući da je Hrvoje slobodan, Ana i Hrvoje postaju par i više nisu slobodni.
6. korak: Manda zaprosi Hrvoja. Iako je Hrvoje zaručen za Anu, više preferira Mandu pa prekida zaruke sa Anom, koja postaje slobodna. Sada su Manda i Hrvoje par i Manda više nije slobodna.
7. korak: Ana zaprosi Davida. Iako je David zaručen za Emu, više preferira Anu pa prekida zaruke sa Emom, koja postaje sobodna. Sada su Ana i David par i Ana više nije slobodna.
8. korak: Ema zaprosi Hrvoja. Budući da je Hrvoje zaručen za Mandu, koju preferira nad Emom, odbija Eminu prosidbu i ostaje zaručen za Mandu. Ema osraje slobodna.
9. korak: Ema zaprosi Oskara. Budući da je Oskar slobodan, Ema i Oskar postaju par i više nisu slobodni.

Sada više nema slobodne žene, dakle nema niti slobodnih muškaraca, što ćemo dokazati u Teoremu 1, algoritam završava. Dobili smo iste parove kao u Primjeru 1.

3. Teorijska analiza algoritma

Prvi teorem kojeg ćemo dokazati pokazuje da, ne samo da Gale – Shapley algoritam završava u konačno mnogo koraka, nego i uvijek vraća stabilno sparivanje, odnosno stabilno sparivanje uvijek postoji.

<u>Teorem 1:</u> Gale – Shapley algoritam uvijek završava u konačno mnogo koraka i vraća stabilno sparivanje.

<u>Dokaz:</u> Pokažimo prvo da while petlja u linijama 2 - 9 završava, odnosno da algoritam završava. Pretpostavimo suprotno, odnosno da while petlja nikada ne završi, tada postoji žena koja je ostala slobodna i nije zaručena. Ako je žena slobodna, to mora biti nakon što je zaprosila sve muškarce i bila odbijena. Da bi muškarac odbio prosidbu, mora biti zaručen. Dakle, svi muškarci su već zaručeni. Kada je muškarac zaručen, ostaje zaručen (ne nužno za istu ženu). S obzirom da je jednak broj žena i muškaraca, slijedi da je svaka žena zaručena, što dovodi do kontradikcije s pretpostavkom da postoji žena koja nije zaručena.

Nadalje, pokažimo da petlja uvijek završava u konačno mnogo koraka. Budući da svaka od *n* žena prolazi kroz svoju rang-listu *n* muškaraca, ne dolazeći nužno do kraja svoje liste, ukupan broj prolazaka kroz petlju je najviše n^2. Dakle, petlja uvijek završava u konačno mnogo koraka te dobivamo sparivanje.

Pokažimo još da ne postoje nestabilnosti. Uočimo prvo da kada je muškarac *m* zaručen za ženu *w*, linije koda u kojima se ponovo promatra muškarac *m* su 6 – 8. Dakle, kada je muškarac zaručen, ostaje zaručen, a ako prekine zaruke sa ženom *w*, to je radi žene koju preferira nad *w*. Pretpostavimo da je žena *w* u paru s muškarcem *m*, ali preferira muškarca *m'*. Pokazat ćemo da *w* i *m'* nisu nestabilnost, budući da *m'* ne preferira *w* nad svojom zaručnicom. Kako *w* preferira *m'* nad *m*, morala je prvo zaprositi muškarca *m'* prije *m* i *m'* ju je ili odbio ili prihvatio te kasnije prekinuo zaruke. Ako je *m'* odbio *w*, to je zato što je *m* već zaručen za ženu koju preferira nad *w*. Ako je *m'* prihvatio zaruke i kasnije ih prekinuo, znači da je u nekom trenutku bio zaručen za *w*, ali ga je kasnije zaprosila žena koju preferira nad *w* i zaručio se s njom. U oba slučaja, *m* je na kraju u paru sa ženom koju preferira nad *w*. Sada, iako *w* preferira *m'* nad svojim partnerom *m*, to nije slučaj kod *m'*, on preferira svoju ženu nad *w*. Dakle, sparivanje koje algoritam vraća nema nestabilnosti i ono je stabilno sparivanje.

<u>Korolar 2:</u> Sa zadanim rang listama za n žena i n muškaraca, Gale – Shapley algoritam se može implementirati sa vremenskom složenosti $O(n^2)$.

<u>Dokaz:</u> Kako je već dokazano u prethodnom teoremu, svaka od *n* žena prolazi kroz svoje rang-liste od *n* muškaraca. U najgorem slučaju, svaka žena morala bi zaprositi sve muškarce sa svoje liste i na kraju biti zaručena sa zadnjim muškarcem s liste. Dakle, while petlja će se za svaku od *n* žena, pokrenuti najviše *n* puta, što daje vremensku složenost $O(n*n)=O(n^2)$.

Iako se u liniji 2 pseudokoda izabire bilo koja slobodna žena, algoritam uvijek vraća isto stabilno sparivanje, što ćemo dokazati idućim teoremom. Štoviše, to stabilno sparivanje je optimalno za žene.

<u>Teorem 3:</u> Redoslijed odabira žena u liniji 2 Gale – Shapley algoritma ne utječe na sparivanje koje algoritam vraća. Ono je uvijek isto stabilno sparivanje u kojemu svaka žena ima za nju najpoželjnijeg muškarca.

<u>Dokaz:</u> Dokažimo najprije da stabilno sparivanje koje vraća algoritam je upravo ono u kojem je svaka žena u paru sa najbojim mogućim muškarcem s kojim može biti. Pretpostavimo suprotno, odnosno da je algoritam vratio stabilno sparivanje *S*, a postoji drugo stabilno sparivanje *S'*, takvo da je *S'* različito od *S*, u kojem žena *w* preferira svog partnera *m'* nad muškarcem *m* s kojim je u paru u *S*. Budući da *w* preferira *m'* nad *m*, morala je zaprositi muškarca *m'* prije nego što je zaprosila *m*. Tada postoji žena *w'* koju *m'* preferira nad *w* i *m'* je već u paru s *w'* kada ga *w* zaprosi ili *m'* prihvaća prosidbu od *w* koju kasnije prekida zbog prosidbe *w'*. Bez smanjenja općenitosti možemo pretpostaviti da je to prvi puta kada je neki muškarac odbio ženu koja već pripada nekom stabilnom sparivanju.

Tvdrimo da w' ne može imati partnera m'' u stabilnom sparivanju kojeg preferira nad m'. Kada bi postojao takav muškarac m'', tada bi w' morala zaprositi m'' i biti odbijena prije nego što zaprosi m'. Ako bi m' prihvatio w i kasnije prekinuo zaruke kako bi prihvatio w', tada, budući da je to prvo odbijanje u tom stabilnom sparivanju, dolazimo do kontradikcije s time da je m'' već odbio w'. S druge strane, ako bi m' već bio u paru s w' kada je w zaprosila, tada, ponovo, m'' nebi mogao prethodno odbiti w', čime smo dokazali tvrdnju.

Budući da w' ne preferira nikoga nad m', a nije s njime uparena u S' (jer je m' u paru s w u S'), w' preferira m' nad svojim partnerom u S'. Budući da w' preferira m' nad svojim partnerom u S' i m' preferira w' nad svojom partnericom u S', tada je par w' i m' nestabilnost u S'. Sada, S' ne može biti stabilno sparivanje zato što sadrži nestabilnost, čime dolazimo do kontradikcije s pretpostavkom da postoji drugo stabilno sparivanje u kojemu postoji žena koja ima boljeg partnera nego u sparivanju S kojeg je vratio algoritam.

Budući da je poredak žena bio proizvoljan, algoritam svaki put vraća isto stabilno sparivanje.

Korolar 4: Stabilno sparivanje nije jedinstveno, odnosno Gale-Shapley algoritam daje jedno moguće stabilno sparivanje.

Dokaz: Prethodni teorem tvrdi da, za dan skup rang-lista, algoritam vraća točno jedno sparivanje, za proizvoljan redoslijed žena, a u Primjeru 2 smo vidjeli da stabilno sparivanje ne mora biti jedinstveno, slijedi da algoritam može vratiti samo jedno od mogućih sparivanja.

Iako algoritam vraća najbolje moguće sparivanje za žene, u idućem korolaru ćemo pokazati da je to sparivanje najgore moguće za muškarce.

Korolar 5: Sparivanje kojeg vraća algoritam, je ono stabilno sparivanje u kojem muškarci dobivaju najgoru moguću partnericu od svih partnerica u ostalim stabilnim sparivanjima.

Dokaz: Neka je S sparivanje koje vraća algoritam. Pretpostavimo da postoji drugo stabilno sparivanje S' i muškarac m koji preferira svoju partnericu w iz S nad partnericom w' it S'. Neka je m' u paru s w u S'. Po teoremu 3, m je najbolji mogući partner kojeg w može imati u svim stabilnijm sparivanjima, što povlači da w preferira m nad m'. Budući da m preferira w nad w', par w i m je nestabilnost u S', što je kontradikcija s time da je S' stabilno sparivanje.

4. Implementacija algoritma u C++ -u

Analogno gornjem pseudokodu, implementiran je Gale-Shapley algoritam u C++ -u uz korištenje strukture *preference*.

```cpp
struct preference{
    int n;
    set<int> freew; //skup slobodnih žena
    vector<vector<int>> women; //vektor vektora preferencije žena
    vector<vector<int>> men; //vektor vektora preferencije muškaraca
    vector<int> wnext; //vektor u kojem spremamo kojeg muškarca koja žena treba idućeg zaprositi (odn. na kojoj poziciji u matrici je stala)
    vector<int> engaged_women; // parovi (žena, muškarac) koji su trenutno zaručeni
    vector<int> engaged_men; // parovi (muškarac, žena) koji su trenutno zaručeni

    preference(int _n, const vector<vector<int>>& _women, const vector<vector<int>>& _men)
        : n(_n), freew(),  women(_women), men(_men), wnext(_n, 0), engaged_women(_n, -1), engaged_men(_n, -1) {
        for (int i = 0; i < n; ++i) { //assign each woman and man as free
            freew.insert(i);
        }
    }
};
```

Slika 2: struktura Preference

```cpp
void GaleShapley(preference &P){
    while(!P.freew.empty()){ //while some woman w is free
        int w = *P.freew.begin();
        int m = P.women[w][P.wnext[w]++]; // let m be the first man on w’s ranked list to whom she has not proposed

        if(P.engaged_men[m] == -1){ //m is free
            P.engaged_women[w] =  m; //w and m become engaged to each other (and not free)
            P.engaged_men[m] =  w;
            P.freew.erase(w);
        } else {
            int w0 = P.engaged_men[m];
            if(P.men[m][w] < P.men[m][w0]) { // elseif m ranks w higher than the woman w0 he is currently engaged to
                P.freew.insert(w0); //m breaks the engagement to w 0 , who becomes free
                P.freew.erase(w);
                P.engaged_women[w] =  m; //w and m become engaged to each other (and not free)
                P.engaged_men[m] =  w;
            }
            // else m rejects w, with w remaining free
        }
    }
}
```

Slika 3: funkcija Gale-Shapley

Funkcija na Slici 3 kao argument prima tip podatka *preference*. Struktura *preference* pamti broj *n* koji predstavlja broj žena, odnosno muškaraca, koje ćemo spajati u parove, dakle, ukupno 2n ljudi. Svaka osoba jednoznačno je određena brojem od 0 do n-1 te se pod tom oznakom nalazi i na listama preferencija osoba suprotnog spola. Skup *set<int> freew* sadrži oznake žena koje su trenutno slobodne. Na početku samog algoritma, taj skup sadrži sve brojeve od 0 do n-1, što se postavlja u konstruktoru. Zatim, *vector<vector<int>> women* i *vector<vector<int>> men* sadrže liste preferencija žena, odnosno muškaraca. Na primjer, u *vector<vector<int>> women* na i-tom mjestu ($i \in \{0, ..., n-1\}$) se nalazi vektor preferencija žene sa oznakom *i*, dakle *women[i][0]* bi sadržavalo oznaku najpoželjnijeg muškarca žene *i*, *women[i][n-1]* bi sadržavalo oznaku najnepoželjnijeg muškarca te žene, dok bi *women[i][j]* sadržavalo oznaku muškarca koji se nalazi na j+1. mjestu ($j \in \{1, ..., n\}$) liste preferencija žene *i*. *vector<int> wnext* je vektor koji pamti na kojoj poziciji (indeksu) na svojoj listi preferencija je pojedina žena stala, odnosno ako je *wnext[i] = j* žena sa oznakom i, ako je slobodna te treba zaprositi idućeg muškarca, to će biti muškarac koji je na j+1. mjestu na njenoj rang-listi, odnosno muškarca sa oznakom *women[i][j]*. Posljednje, struktura pamti sparivanje pomoću dva vektora *vector<int> engaged_women* i *vector<int> engaged_men* koji sadrže iste podatke, samo sa suprotnim oznakama. Na primjer *engaged_women[i]* sadrži oznaku muškarca s kojim je žena *i* zaručena, recimo da je to muškarac sa oznakom *j*, tada je *engaged_men[j] = i*. U konstruktoru, budući da su sve osobe na početku slobodne i nitko nije zaručen, na sva mjesta u vektoru postavlja se vrijednost -1.

Slika 3 prikazuje Gale-Shapley algoritam implementiran u C++ -u. Uz pojedine linije koda kao komentar dodane su linije iz pseudokoda. Sve dok skup slobodnih žena *freew* nije prazan, uzimamo prvu slobodnu ženu *w* iz skupa (bez smanjenja općenitosti možemo uzeti prvu ženu iz skupa budući da smo dokazali da savršeno sparivanje dobiveno algoritmom ne ovisi o odabiru žena iz tog skupa). Zatim uzimamo idućeg muškarca *m* sa liste preferencija odabrane žene i odmah vrijednost u vektoru *wnext* na mjestu žene *w* povećavamo za 1, odnosno bilježimo kojeg muškarca će žena *w* morati iduće zaprositi. Ako je *P.engaged_men[m] == -1*, muškarac *m* još nije zaručen te žena *w* i muškarac *m* postaju par, što moramo evidentirati u vektorima *engaged_women* i *engaged_men* i obrisati ženu iz skupa slobodnih žena, budući da sada ona više nije slobodna. Inače, muškarac *m* je zaručen, pa u varijablu *w0* spremamo oznaku njegove trenutne žene. Sada, ako se žena *w* nalazi na manjem indeksu od žene *w0*, znači da ženu *w* preferira nad ženom *w0*, žena *w0* postaje slobodna, dok *w* više nije slobodna te *w* i *m* postaju par i upisujemo to u vektore sparivanja. Kada bi žena *w0* bila više na rang listi od žene *w*, tada nebi trebali raditi nikakve promjene u našim podacima budući da bi muškarac *m* odbio zaruke.

```cpp
vector<vector<int>> Permutation (int n){
    vector<vector<int>> all_permutations;

    vector<int> elements(n);
    for (int i = 0; i < n; ++i) {
        elements[i] = i;
    }

    for (int i = 0; i < n; ++i) {

        unsigned seed = chrono::system_clock::now().time_since_epoch().count();
        shuffle(elements.begin(), elements.end(), default_random_engine(seed));

        all_permutations.push_back(elements);
    }

    return all_permutations;
}
```

Slika 4: Funkcija za konstrukciju vektora preferencija

Za konstrukciju vektora preferencija muškaraca i žena pozivana je funkcija Permutation koja prima broj *n* te prvo konstruira vektor *vector<int> elements(n)*, veličine n, brojeva od 0 do n-1. Zatim, u idućoj for petlji konstruira n permutacija brojeva iz vektora *elements* pomoću funkcije shuffle koja koristi „sjeme“ *chrono::system_clock::now().time_since_epoch().count()* koje daje trenutno vrijeme u trajanju od epohe. Tada svako izvršavanje programa ima različito sjeme što rezultira različitim permutacijama. Svih n dobivenih permutacija zapisujemo u *vector<vector<int>> all_permutations* koji funkcija vraća u glavni program te se kasnije koriste u Gale-Shapley algoritmu.

5. Empirijska analiza algoritma

Empirijska analiza provedena je na prijenosnom računalu sa sljedećim specifikacijama:

Procesor: AMD Ryzen 7 7730U with Radeon Graphics 2.00 GHz

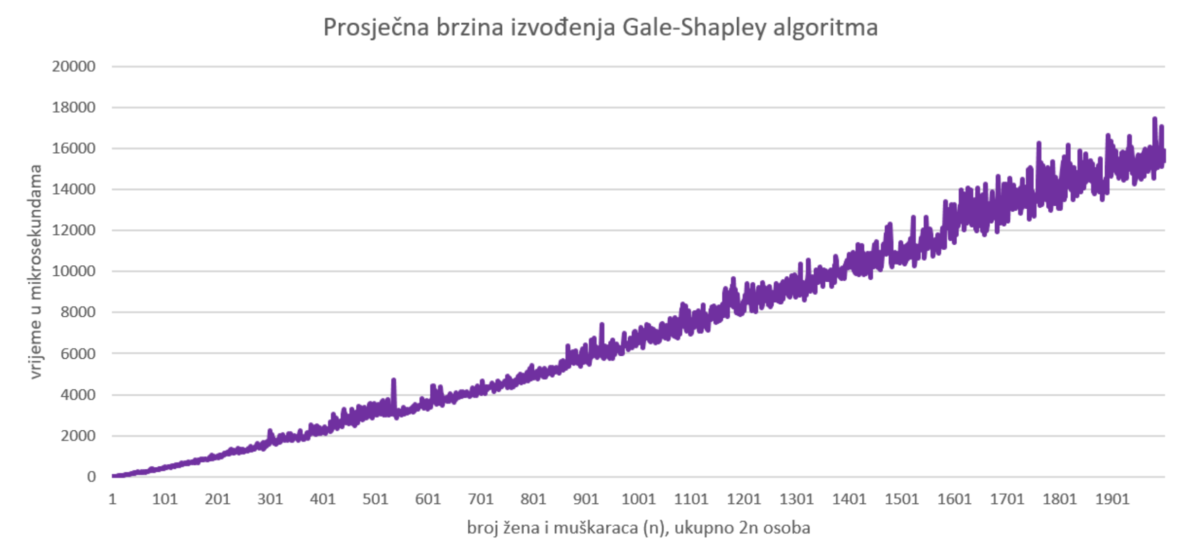
Instalirani RAM: 16,0 GB (13,8 GB iskoristivo)

Vrsta sustava: 64-bitni operacijski sustav, procesor x64

Operacijski sustav: Windows 11 Pro, verzija: 22H2, međuverzija operacijskog sustava: 22621.2861

Kompajler: gcc (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0

Algoritam je pozivan za n ∈ {1, ... 2000}, za svaki n po 10 puta. Mjerenje vremena izvršavanja Gale-Shapley algoritma odnosno funkcije sa Slike 3 provedeno je pomoću biblioteke *chrono* bilježenjem trenutnog vremena prije izvršavanja funkcije i nakon izvršavanja funkcije te oduzimanjem te dvije vrijednosti. Vrijeme je mjereno u mikrosekundama. Zatim se računalo prosječno vrijeme za svaki n od 10 ponavljanja te se vrijednost zapisivala u CSV datoteku. Dobivene vrijednosti prikazane su na Grafu 1 ispod.



Graf 1: Prosječna brzina izvođenja Gale-Shapley algoritma

Iz grafa možemo zaključiti da je u slučaju mjerenja prosječnog vremena izvođenja vremenska složenost skoro linearna. Što bi značilo da, najčešće, svaka žena ima različitog muškarca prvog na svojoj listi preferencija te se tada algoritam za svaku od n žena iterira samo jednom, budući da će svaka žena zaprositi smo prvog muškarca sa svoje liste, koji će prosidbu prihvatiti, a do druge prosidbe neće ni doći, čime dobivamo vremensku složenost O(n*1) = O(n). Razlog tome su slučajno generirani vektori preferencija koji rijetko kada imaju istu vrijednost na prvom mjestu.

6. Zaključak

Problem stabilnog braka efikasno rješava Gale-Shapley algoritam koji je u ovome radu postavljen na način orijentiran prema ženama. Dakle, žene prose muškarce, redom po svojoj listi preferencija, dok svi ne budu upareni i sparivanje bude stabilno. Iako je vremenska složenost $O(n^2)$ u najgorem slučaju, kada svaka žena mora zaprositi i posljednjeg muškarca sa svoje liste, empirijskom analizom je pokazano da je ta složenost u prosjeku O(n) budući da se nasumičnim generiranjem preferencija dobivaju liste koje najčešće nemaju iste oznake na prvom mjestu kada ne dolazi niti do druge prosidbe te dobivamo stabilno sparivanje samo jednim prolaskom kroz algoritam.

7. Literatura

- T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, Introduction to Algorithms, Fourth edition, MIT Press, 2022.
- https://en.cppreference.com/w/cpp/chrono
- https://cplusplus.com/reference/algorithm/shuffle/
- https://web.math.pmf.unizg.hr/nastava/oaa/seminari/2014_15/Horvat_Stabilno-sparivanje.pdf