Problem stabilnog sparivanja

Udovičić, Kristina

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:217:190366

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-29**

Repository / Repozitorij:

Repository of the Faculty of Science - University of Zagreb

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO–MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

Kristina Udovičić

PROBLEM STABILNOG SPARIVANJA

Diplomski rad

Voditelj rada:
doc. dr. sc. Goranka Nogo

Zagreb, rujan 2021.

Ovaj diplomski rad obranjen je dana ____________________ pred ispitnim povjerenstvom u sastavu:

1. ____________________________, predsjednik
2. ____________________________, član
3. ____________________________, član

Povjerenstvo je rad ocijenilo ocjenom _______________.

Potpisi članova povjerenstva:

1. ____________________________
2. ____________________________
3. ____________________________

Hvala profesoru Saši Singeru što me potaknuo na istraživanje ovog problema.
Hvala mojoj mentorici, profesorici Goranki Nogo, na susretljivosti, savjetima i strpljenju tijekom pisanja rada.
Hvala mojoj obitelji: roditeljima, sestrama i bratu, mnogobrojnoj rodbini i mom dragom, na ljubavi, podršci i zajedništvu.
Hvala mojim prijateljicama i prijateljima na svim trenucima radosti, ohrabrenja i veselja.
Rad posvećujem svojoj mami.
I na kraju, odnosno na početku: Omnia ad maiorem Dei gloriam.

Sadržaj

Uvod

U diplomskom radu obrađuje se problem stabilnog sparivanja.

Prvo poglavlje rada navodi kontekste službene i neslužbene pojave problema stabilnog sparivanja te opisuje generalizaciju problema stabilnog sparivanja obrađenu u radu: problem stabilnih brakova.

Drugim poglavljem definiraju se pojmovi relevantni za pojam sparivanja.

Analiziranjem Gale-Shapleyjevog algoritma u trećem poglavlju, dokazuju se egzistencija i mogućnost efektivne konstrukcije stabilnog sparivanja za svaki problem stabilnog sparivanja. Pokazuje se da je rezultat svih izvršavanja Gale-Shapleyjevog algoritma, za zadani problem stabilnog sparivanja, uvijek isto stabilno sparivanje te se razmatra generalizacija problema stabilnog sparivanja sa zabranjenim parovima.

Četvrto poglavlje demonstrira problem stabilnog sparivanja u praksi: svođenje opisanog problema na problem stabilnog sparivanja te implementaciju algoritma za rješavanje istog.

Poglavlje 1

Opis problema

Problem stabilnog sparivanja jedan je od najpoznatijih problema u području oblikovanja i analize algoritama. Radi se o pronalasku takozvanog stabilnog sparivanja dviju skupina, čiji članovi imaju rang-liste preferencija članova druge skupine. Sparivanje upućuje na definiranje uređenih parova čiji je prvi element član prve skupine, a drugi element član druge skupine. Pridjev stabilno označava zahtjev da nakon što su uređeni parovi konačno definirani, njihovo razdvajanje nije moguće.

Poznate su dvije neovisne pojave ovog problema. Kronološki kasnija i značajnija veže se uz dva matematičara, ujedno i ekonomista, Davida Galea i Lloyda S. Shapleyja koji su se 1962. pitali je li moguće dizajnirati proces upisa na fakultet, ili proces raspodjele poslova, takav da interes sudionika procesa jamči stabilan rezultat procesa.[1]

Deset godina ranije, kao odgovor na manipulacije procesom dodjele stažista bolnicama, nastaje National Resident Matching Program.

U potpoglavljima koja slijede opisani su koncepti vezani za problem stabilnog sparivanja i verzija problema stabilnog sparivanja koja se obrađuje u radu.

1.1 Gale i Shapley - proces upisa na fakultet

Proces upisa na fakultet promatra se kao problem stabilnog sparivanja zadan skupinama kandidata i fakulteta te rang-listama preferencija svih članova obiju skupina. Članovima prve skupine, kandidatima, mogu biti pridružene značajke vezane uz prosjek, uspjeh na maturi te uspjeh na prijemnom ispitu. Članovima druge skupine, fakultetima, pridružene su značajke koje definiraju broj slobodnih mjesta za upis i minimalne kriterije koje kandidati moraju ispuniti da bi dospjeli na rang-listu preferencija fakulteta.

[1] U stranoj literaturi, za takav se proces kaže da je *self-enforcing*.

Svaki kandidat razmatra ponuđene fakultete, izabire one koje bi želio upisati te ih rangira na svojoj rang-listi preferencija ovisno o svojim željama i mogućnostima. Svaki fakultet ovisno o svojim kriterijima rangira kandidate za upis i onemogućuje upis nekvalificiranima. Pojavom novih kvalificiranih kandidata zainteresiranih za upis, fakultet ih dodaje na svoju rang-listu preferencija.

Gale i Shapley osvrnuli su se na propuste procesa upisa na fakultet iz perspektive članova obiju skupina. U nastavku će biti jasno da osim što članovi skupina mogu biti zakinuti tijekom procesa, istim mogu i manipulirati.

Svaki fakultet ima određen broj q upisnih mjesta i cilj mu je primiti što kvalitetnije kandidate. Često je slučaj da broj kandidata zainteresiranih za upis fakulteta prelazi upisnu kvotu. Prva zapreka s kojom se fakulteti suočavaju je činjenica da ponuda upisa samo prvim q najbolje rangiranim kandidatima nije dobra taktika za popunjavanje svih upisnih mjesta jer nije sigurno da će sve ponude za upis biti prihvaćene. Dodatno, fakulteti su shvatili da moraju ponuditi više upisnih mjesta nego što ih zbilja imaju te da je potrebno taktički odabrati kandidate kojima će ponuditi upis. Za određivanje novog broja upisnih mjesta, ali i za odabir kandidata kojima će biti ponuđen upis, fakulteti se djelomično koriste strategijom nagađanja. Razlog je taj što, za svakog razmatranog kandidata, postoje pitanja na koja fakulteti nemaju odgovor:

- Je li kandidat istovremeno zainteresiran za upis na druge fakultete?
- Kako kandidat rangira fakultete za koje je zainteresiran?
- Koji će fakulteti također ponuditi upis kandidatu?

U ovoj situaciji, fakulteti se mogu samo nadati da će stvarne upisne kvote biti približno popunjene, kao i da će kvaliteta studenata biti približno zadovoljena.

S druge strane, i kandidati se susreću s poteškoćama tijekom procesa upisa. Primjerice, fakultet može od kandidata zahtijevati da prilikom prijave za upis priloži svoju rang-listu fakulteta. Kandidat bi tada mogao posumnjati da neće dobiti ponudu za upis na taj fakultet jer ga nije rangirao dovoljno visoko na svojoj rang-listi preferencija. S vremenom su kandidati shvatili da će vjerojatnije ostvariti upis željenog fakulteta ako ne budu iskreni prilikom rangiranja fakulteta te su počeli kalkulirati pri sastavljanju svojih rang-lista preferencija.

Još jedna nezgodna ideja za kandidate, osmišljena od strane fakulteta, su liste čekanja. Pomoću njih fakulteti informiraju zainteresirane kandidate, koje trenutno nisu primili, da postoji šansa dobivanja ponude za upis, u slučaju da više rangirani kandidati odustanu od upisa. Ovo prividno rješenje uzrokuje nove nezgodne situacije. Primjerice, kandidat može dobiti ponudu za upis od fakulteta koji je niže rangirao, a istovremeno se nalaziti na listi čekanja njemu prioritetnijeg fakulteta. Tada kandidat treba birati između dvije opcije: prihvatiti ponuđenu lošiju ponudu - te time osigurati upis - ili čekati nadajući se ponudi za upis od prioritetnijeg fakulteta - i pritom riskirati da ne bude upisan ni na jedan fakultet.

Ovakva provedba upisa uzrokovala je nezadovoljstvo upisanih kandidata i fakulteta, ali im je i omogućila da pokušaju poboljšati svoj rezultat - nauštrb drugih sudionika. Sljedeći primjer pokazuje neučinkovitost ovakvog procesa upisa i njegove posljedice.

Primjer 1.1.1. Završetkom procesa upisa, kandidat α upisao je fakultet A, a kandidat β upisao je fakultet B. Rang-liste preferencija kandidata α i fakulteta B ukazuju na mogućnost povoljnijeg upisa za oba člana: kandidat α rangirao je fakultet B više od fakulteta A na svojoj rang-listi preferencija, a fakultet B bi radije upisao kandidata α nego kandidata β. U ovoj situaciji, kandidat α i fakultet B spremni su raskinuti dogovor s dodijeljenom stranom, fakultetom A i kandidatom β, te ih pritom ništa ne sputava. Dakle, kandidat α će poslati zamolbu za upis fakultetu B, fakultet B će odgovoriti potvrdno i pritom ispisati nekog kandidata γ kojeg je rangirao niže od kandidata α na svojoj rang-listi preferencija. Nakon prihvaćene zamolbe za upis, kandidat α će se ispisati s fakulteta A. Tako će nastati novi upis: kandidat α će upisati fakultet B. Taj novi upis će uzrokovati dva ispisa kojima će fakultet A ostati bez jednog studenta, a kandidat β neće biti upisan ni na jedan fakultet.

Opisani događaj uzrokuje domino efekt: kandidat β šalje zamolbe za upis ostalim fakultetima sa svoje rang-liste preferencija, a fakultet A želi popuniti ispražnjeno mjesto i šalje ponude za upis neupisanim kandidatima sa svoje rang-liste preferencija. Razvrgavaju se prethodno postignuti upisi i dolazi do novih. Budući da se novi upis iz primjera nepovoljno odražava na jednog kandidata i jedan fakultet, te se svi ostali novi upisi uzrokovani njime nepovoljno odražavaju na jednog kandidata ili jedan fakultet, mnogi kandidati i fakulteti mogu dospjeti u ovaj začarani krug.

Novi upis također može postati novi ispis - kandidat i dalje može slati molbe za upis drugim fakultetima, kao i fakultet ponude drugim kandidatima. Zbog svega spomenutog, teško je procijeniti kada će novonastali proces upisa i ispisa završiti.

Gale i Shapley dokazali su da je moguće izvršiti upis na fakultete izbjegavanjem opisanih situacija tako da rezultatom budu zadovoljni i kandidati, i fakulteti. Zadovoljnim kandidatom, ili pak fakultetom, smatra se onaj koji neće odustati od dogovorenog upisa da bi postigao povoljniji.

Sljedeći primjer suprotan je prethodnom primjeru, a opisuje rezultat upisa na fakultete koji se želi postići završetkom procesa upisa.

Primjer 1.1.2. Završetkom procesa upisa, kandidat α upisao je fakultet A, a kandidat β upisao je fakultet B. Kandidat α rangirao je fakultet B više of fakulteta A na svojoj rang-listi preferencija pa šalje zamolbu za upis fakultetu B. Fakultet B nakon zaprimljene zamolbe uspoređuje kandidata α s upisanim kandidatima. Sve upisane kandidate rangirao je više od kandidata α na svojoj rang-listi preferencija zbog čega odbija zamolbu kandidata α. Dakle, iz vlastitog interesa fakultet B neće narušiti postignute dogovore.

Prethodni primjer mogao je opisivati situaciju u kojoj fakultet B šalje ponudu za upis kandidatu α jer je kandidat α na njegovoj rang-listi preferencija rangiran više od nekog upisanog kandidata γ. Kandidat α razmatra ponudu fakulteta B, ali ju odbija jer je upisani fakultet rangirao više od fakulteta B na svojoj rang-listi preferencija. Drugim riječima, opet interes sudionika, ovaj put kandidata, štiti postignute dogovore od narušavanja.

Sada se može preformulirati pitanje koje su postavili Gale i Shapley, a koje je spomenuto na početku poglavlja; ako su zadane rang-liste preferencija kandidata i fakulteta, je li moguće upisati kandidate na fakultete, tako da za svakog kandidata α i za svaki fakultet B koji kandidat α nije upisao, vrijedi barem jedna od sljedeće dvije tvrdnje:

- Fakultet B rangirao je svakog upisanog kandidata više od kandidata α na svojoj rang-listi preferencija.

- Kandidat α rangirao je upisani fakultet više od fakulteta B na svojoj rang-listi preferencija.

Kada se postigne navedeno, rezultat upisa na fakultet bit će stabilan jer će interes sudionika jamčiti da neće doći do raskida postignutih dogovora o upisima. Tada će se moći reći da je problem sparivanja zadan skupinama kandidata i fakulteta, i pripadnim rang-listama preferencija, uspješno riješen odnosno da je za njega pronađeno stabilno sparivanje.

1.2 National Resident Matching Program

National Resident Matching Program, u daljnjem tekstu NRMP, je postupak primanja diplomiranih studenata medicine na natječaje stažiranja u bolnicama. NRMP nastao je 1950-ih kao odgovor na nepravedne situacije u kojima bi se našli studenti medicine i bolnice tijekom, ali i nakon provedbe primanja stažista.

Poteškoće su nastale jer je broj diplomiranih studenata medicine bio manji od broja mjesta za stažiranja pa su bolnice, želeći se osigurati, objavljivale natječaje za stažiranja nekoliko godina prije no što su studenti diplomirali. To je uzrokovalo da studenti, iz straha da ne ostanu bez ikakvog stažiranja, prihvate stažiranje ranije nego što su odlučili u kojem se području žele usavršavati. Događalo se da mnogi studenti, bilo zbog pronalaska drugog stažiranja ili pak razočaranja, odustanu od dogovorenog stažiranja. Nakon njihovog odustajanja, bolnice često nisu uspijevale naći novog stažista za prazno mjesto.

Navedene probleme NRMP uspješno je riješio koristeći *Boston Pool algoritam*.

Pojava dodatnih zahtjeva, nastala porastom broja studentica, zahtijevala je redizajn algoritma. Naime, postalo je uobičajeno da par studenata želi stažirati u istoj regiji. Pritisak su stvarale i opravdane kritike da NRMP favorizira neke bolnice. Godine 1998. algoritam su redizajnirali Alvin E. Roth i Elliott Peranson, tako da zadovolji kompleksnije zahtjeve studenta, ali i da bude otporan na manipulacije - kako studenata, tako i bolnica.

1.3 Generalizacija - problem stabilnih brakova

Iz prethodnih potpoglavlja vidljiva je prirodna motivacija za stabilnim sparivanjem, kao i mogućnost različitih interpretacija skupina i članova kojima se zadaje problem stabilnog sparivanja. Ovisno o kontekstu problema, skupine ne moraju sadržavati jednak broj članova, iz čega odmah slijedi da neki članovi jedne skupine neće dobiti svoj par iz druge skupine. Zanimljivo je koliko odluka jednog člana, bila ona prihvaćanje ili odbijanje dobivene ponude, može utjecati na velik broj drugih članova obiju skupina.

Iako su problem stabilnog sparivanja isprva promatrali kroz primjere upisa na fakultet i raspodjele radnika po poslovima, Gale i Shapley su algoritam i matematičku pozadinu prilagodili takozvanom problemu stabilnih brakova.

Problem stabilnih brakova osnovna je verzija problema stabilnog sparivanja. Zadaje se disjunktnim skupinama koje se sastoje od istog broja članova, a svakom članu pojedine skupine pridružena je rang-lista preferencija svih članova druge skupine. Rang-listom preferencija svaki član jednoznačno određuje koji je član iz druge skupine njemu prvi najpoželjniji za sparivanje, koji je član iz druge skupine njemu drugi najpoželjniji za sparivanje, i tako redom do kraja. Točnije, ne postoje nepoželjni članovi - članovi s kojima se nikako ne želi biti u paru, čak i pod cijenu nesparivanja. Zatim, nije dopušteno da dva člana, ili više njih, iz druge skupine budu jednako poželjni za sparivanje. Zahtjevi članova, odnosno kriteriji koje članovi koriste prilikom rangiranja članova druge skupine, nisu relevantni u ovoj verziji problema stabilnog sparivanja.

Poglavlje 2

Vrste sparivanja

U poglavlju su definirani osnovni pojmovi problema stabilnog sparivanja zadanog dvama skupovima istog kardinaliteta n: skupom muškaraca M i skupom žena W. Za elemente skupa M koristit će se oznake: m, m', m'', m_i; za elemente skupa W: w, w', w'', w_i. U ovom poglavlju, prvi zadani skup je skup muškaraca M. Važnost redoslijeda zadavanja skupova i slučaj u kojem je skup žena W prvi zadani skup bit će objašnjeni u trećem poglavlju.

2.1 Savršeno sparivanje

Neka su $M = \{m_1, ..., m_n\}$ skup od n muškaraca i $W = \{w_1, ..., w_n\}$ skup od n žena, takvi da vrijedi $M \cap W = \emptyset$. Neka je $M \times W$ skup svih mogućih uređenih parova oblika (m, w), odnosno neka je $M \times W = \{(m, w) : m \in M, w \in W\}$.

Definicija 2.1.1. ***Sparivanje*** *S je skup uređenih parova iz $M \times W$, sa svojstvom da se svaki muškarac iz M i svaka žena iz W nalaze u najviše jednom uređenom paru skupa S.*

Definicija 2.1.2. ***Savršeno sparivanje*** *S' je sparivanje sa svojstvom da se svaki muškarac iz M i svaka žena iz W nalaze u točno jednom uređenom paru sparivanja S'.*

Primjer 2.1.3. Neka su $M = \{m_1, m_2, m_3, m_4\}$ skup muškaraca i $W = \{w_1, w_2, w_3, w_4\}$ skup žena. Neka su skupovi S_1 i S_2 zadani na sljedeći način:

$$S_1 = \{(m_1, w_2), (m_1, w_4), (m_3, w_3)\}, \tag{2.1}$$

$$S_2 = \{(m_2, w_2), (m_4, w_1)\}, \tag{2.2}$$

Skup S_1 nije sparivanje jer se muškarac m_1 nalazi u dva uređena para skupa S_1. Skup S_2 je sparivanje, ali nije savršeno sparivanje jer se muškarci m_1, m_3 i žene w_3, w_4 ne nalaze ni u jednom uređenom paru sparivanja S_2.

Primjer 2.1.4. Neka su $M = \{m_1, m_2, m_3, m_4\}$ skup muškaraca i $W = \{w_1, w_2, w_3, w_4\}$ skup žena. Neka su skupovi S_3 i S_4 zadani na sljedeći način:

$$S_3 = \{(m_1, w_2), (m_2, w_4), (m_3, w_3), (m_4, w_1)\}, \tag{2.3}$$

$$S_4 = \{(m_1, w_2), (m_2, w_1), (m_3, w_1), (m_4, w_3)\}. \tag{2.4}$$

Skup S_3 je savršeno sparivanje. Skup S_4 nije savršeno sparivanje. Štoviše, skup S_4 nije sparivanje jer se žena w_1 nalazi u dva uređena para skupa S_4.

2.2 Stabilno sparivanje

Svakom muškarcu $m \in M$, i svakoj ženi $w \in W$, pridružena je rang-lista preferencija svih žena, odnosno muškaraca. Iduće dvije analogne definicije objašnjavaju značenje preferiranja, odnosno rangiranja na rang-listi preferencija.

Definicija 2.2.1. *Kaže se da muškarac $m \in M$* ***preferira*** *ženu $w \in W$* ***više*** *od žene $w' \in W$, ako je žena w rangirana više od žene w' na njegovoj rang-listi preferencija.*

Definicija 2.2.2. *Kaže se da žena $w \in W$* ***preferira*** *muškarca $m \in M$* ***više*** *od muškarca $m' \in M$, ako je muškarac m rangiran više od muškarca m' na njenoj rang-listi preferencija.*

Rang-liste preferencija praktično je prikazati matrično, dvjema matricama preferencija. Matrica preferencija u $i-$tom retku sadržava rang-listu preferencija muškarca m_i, odnosno rang-listu preferencija žene w_i.

Primjer 2.2.3. Neka su $M = \{m_1, m_2, m_3\}$ skup muškaraca i $W = \{w_1, w_2, w_3\}$ skup žena. Matrice preferencija muškaraca m_i i žena w_i zadane su na sljedeći način:

$$\begin{array}{c} m_1 : \\ m_2 : \\ m_3 : \end{array} \begin{bmatrix} w_2 & w_3 & w_1 \\ w_3 & w_1 & w_2 \\ w_3 & w_2 & w_1 \end{bmatrix}, \quad \begin{array}{c} w_1 : \\ w_2 : \\ w_3 : \end{array} \begin{bmatrix} m_1 & m_2 & m_3 \\ m_2 & m_3 & m_1 \\ m_1 & m_3 & m_2 \end{bmatrix}. \tag{2.5}$$

Rang-lista preferencija muškarca m_3 je: w_3, w_2, w_1, što znači da je za muškarca m_3 žena w_3 prva najpoželjnija za sparivanje, žena w_2 druga najpoželjnija za sparivanje, a žena w_1 treća najpoželjnija za sparivanje.

Napomena 2.2.4. *Za uređeni par (m, w) nekog sparivanja S, kaže se da je muškarac m partner žene w u tom sparivanju. Analogno, kaže se da je žena w partnerica muškarca m u sparivanju S.*

Iduća definicija opisuje situaciju koja se nastoji izbjeći u savršenom sparivanju.

Definicija 2.2.5. *Neka je zadano savršeno sparivanje S. Neka su $(m, w), (m', w') \in S$ sa svojstvom da muškarac m preferira ženu w' više od svoje partnerice w i da žena w' preferira muškarca m više od svog partnera m'. Tada se kaže da je (m, w')* ***nestabilnost*** *u S.*

Uređeni par (m, w') iz prethodne definicije nije uređeni par sparivanja S, ali i muškarac m i žena w' preferiraju jedno drugo više od svojih partnera u sparivanju S. Cilj je da sparivanje ne sadrži nestabilnosti, odnosno da bude *stabilno*.

Definicija 2.2.6. *Za sparivanje S kaže se da je* ***stabilno*** *ako su ispunjena dva uvjeta:*

- *S je savršeno sparivanje.*
- *S ne sadržava nestabilnosti.*

Definicija 2.2.7. *Za sparivanje S kaže se da je* ***nestabilno****, ako nije stabilno.*

Primjer 2.2.8. Neka je $M = \{m_1, m_2, m_3, m_4\}$ skup muškaraca i $W = \{w_1, w_2, w_3, w_4\}$ skup žena te neka su matrice preferencija muškaraca m_i i žena w_i zadane na sljedeći način:

$$\begin{array}{c} m_1: \\ m_2: \\ m_3: \\ m_4: \end{array} \begin{bmatrix} w_1 & w_3 & w_2 & w_4 \\ w_3 & w_4 & w_1 & w_2 \\ w_4 & w_2 & w_3 & w_1 \\ w_4 & w_2 & w_1 & w_3 \end{bmatrix}, \quad \begin{array}{c} w_1: \\ w_2: \\ w_3: \\ w_4: \end{array} \begin{bmatrix} m_2 & m_4 & m_3 & m_1 \\ m_4 & m_3 & m_1 & m_2 \\ m_1 & m_4 & m_2 & m_3 \\ m_4 & m_1 & m_2 & m_3 \end{bmatrix}. \tag{2.6}$$

Neka su sparivanja S_5, S_6 i S_7 zadana na sljedeći način:

$$S_5 = \{(m_1, w_1), (m_2, w_3), (m_3, w_2), (m_4, w_4)\}, \tag{2.7}$$

$$S_6 = \{(m_1, w_3), (m_2, w_1), (m_3, w_4), (m_4, w_2)\}, \tag{2.8}$$

$$S_7 = \{(m_1, w_3), (m_2, w_1), (m_3, w_2), (m_4, w_4)\}. \tag{2.9}$$

Sparivanje S_5 je stabilno. Doista, svaki muškarac $m \in M$ i svaka žena $w \in W$ nalaze se u točno jednom uređenom paru sparivanja S_5 iz čega slijedi da je sparivanje S_5 savršeno sparivanje. Nadalje, S_5 ne sadržava nestabilnosti. Iz ovih dviju tvrdnji, po definiciji 2.2.6, slijedi da je sparivanje S_5 stabilno. Navedeno vrijedi i za sparivanje S_7.

Sparivanje S_6 je nestabilno. Sparivanje S_6 zadovoljava prvi uvjet stabilnosti: S_6 je savršeno sparivanje. Drugi uvjet stabilnosti nije ispunjen. Analizom prvog uređenog para $(m_1, w_3) \in S_6$ zaključuje se da muškarac m_1 preferira ženu w_1 više od svoje partnerice w_3, ali žena w_1 ne preferira muškarca m_1 više od svog partnera m_2. Nadalje, muškarac m_1 ne preferira žene w_2 i w_4 više od svoje partnerice w_3 pa muškarac m_1 neće biti u nestabilnosti. Istim postupkom zaključuje se da muškarac m_3 i žene w_1, w_2, w_3 neće biti u nestabilnosti.

Nakon analize uređenih parova sparivanja S_6 uočavaju se dvije nestabilnosti: (m_2, w_4) i (m_4, w_4). Muškarac m_2 preferira ženu w_4 više od svoje partnerice w_1, a žena w_4 preferira muškarca m_2 više od svog partnera m_3. Također, žena w_4 preferira muškarca m_4 više od svog partnera m_3. Dakle, sparivanje S_6 je nestabilno.

2.3 Optimalno sparivanje

U prethodnom primjeru za dane skupove M i W definirana su dva stabilna sparivanja. Budući da pojedini muškarac, odnosno pojedina žena, u različitim stabilnim sparivanjima ima različite partnerice, odnosno partnere, postavlja se pitanje u kakvom su odnosu sva stabilna sparivanja za zadana dva skupa i pripadne rang-liste preferencija.

Partnerica muškarca m_2 iz prethodnog primjera u stabilnom sparivanju S_5 je žena w_3, a u stabilnom sparivanju S_7 žena w_1. Muškarac m_2 preferira ženu w_3 više od žene w_1, što znači da je stabilno sparivanje S_5 povoljnije za njega. Može se reći da je muškarac m_2 u stabilnom sparivanju S_5 više profitirao.

Partner žene w_3 iz prethodnog primjera u stabilnom sparivanju S_5 je muškarac m_2, a u stabilnom sparivanju S_7 muškarac m_1. Žena w_3 preferira muškarca m_1 više od muškarca m_2, što znači da je stabilno sparivanje S_7 povoljnije za nju.

Zbog navedenog ni jedno od ovih stabilnih sparivanja nije moguće izdvojiti kao ”bolje”.

Ipak, stabilna sparivanja ne uspoređuju se određivanjem koje je od dva stabilna sparivanja ”bolje” nego se utvrđuje postoji li među svim stabilnim sparivanjima jedno stabilno sparivanje koje je ujedno i *optimalno*. To je stabilno sparivanje S takvo da svi muškarci i sve žene u svim drugim stabilnim sparivanjima ”prođu lošije” ili ”jednako dobro” kao u S.

Definicija 2.3.1. *Za stabilno sparivanje S kaže se da je **optimalno** ako za svaki uređeni par $(m, w) \in S$ vrijedi:*

- *U svakom drugom stabilnom sparivanju S', za uređeni par $(m, w') \in S'$ vrijedi jedna od dvije tvrdnje:*
  - *muškarac m preferira ženu w više od žene w',*
  - *žena w' je žena w.*
- *U svakom drugom stabilnom sparivanju S', za uređeni par $(m', w) \in S'$ vrijedi jedna od dvije tvrdnje:*
  - *žena w preferira muškarca m više od muškarca m',*
  - *muškarac m' je muškarac m.*

Za zadani problem stabilnog sparivanja, optimalno sparivanje ne mora postojati, no ako postoji, onda je ono jedinstveno.

Teorem 2.3.2. *Optimalno sparivanje je jedinstveno.*

Dokaz. Kontradikcijom. Neka su S i S' optimalna sparivanja. Budući da su S i S' različita sparivanja, partnerica nekog muškarca m u sparivanju S je žena w, a u sparivanju S' je žena w', $w \neq w'$. Muškarac m ne može jednako preferirati žene w i w'. Bez smanjenja općenitosti, neka muškarac m preferira ženu w više od žene w'. Dakle, sparivanje S' nije optimalno što je kontradikcija s pretpostavkom da su S i S' optimalna sparivanja. Dakle, optimalno sparivanje je jedinstveno. □

Napomena 2.3.3. *Optimalno sparivanje sigurno postoji u slučaju kada je svaki muškarac m prvi najpoželjniji za sparivanje ženi w koja je njemu prva najpoželjnija za sparivanje.*

Kada za problem stabilnog sparivanja ne vrijedi prethodna napomena, postojanje optimalnog sparivanja utvrđuje se usporedbom svih stabilnih sparivanja tako da se za svako stabilno sparivanje određuje koliko je povoljno za sve muškarce i sve žene. U sljedećem primjeru razmatra se problem stabilnog sparivanja iz primjera 2.2.3; matrice preferencija navode se ponovno radi preglednosti.

Primjer 2.3.4. Neka je problem stabilnog sparivanja zadan sljedećim matricama preferencija:

$$\begin{array}{c} m_1: \\ m_2: \\ m_3: \end{array} \begin{bmatrix} w_2 & w_3 & w_1 \\ w_3 & w_1 & w_2 \\ w_3 & w_2 & w_1 \end{bmatrix}, \quad \begin{array}{c} w_1: \\ w_2: \\ w_3: \end{array} \begin{bmatrix} m_1 & m_2 & m_3 \\ m_2 & m_3 & m_1 \\ m_1 & m_3 & m_2 \end{bmatrix}. \tag{2.10}$$

Budući da je $|M| = |W| = 3$, slijedi da je ukupan broj savršenih sparivanja za zadani problem sparivanja jednak $3! = 6$. Za svako od šest savršenih sparivanja potrebno je utvrditi je li stabilno, slično kao u primjeru 2.2.8. Pokaže se da su dva savršena sparivanja ujedno i stabilna[1]:

$$S_3 = \{(m_1, w_2), (m_2, w_1), (m_3, w_3)\}, \tag{2.11}$$

$$S_5 = \{(m_1, w_3), (m_2, w_1), (m_3, w_2)\}, \tag{2.12}$$

Promatranjem navedenih stabilnih sparivanja iz perspektive muškaraca i žena, zaključuje se sljedeće: za muškarce m_1 i m_3 povoljnije je sparivanje S_3, za žene w_2 i w_3 povoljnije je sparivanje S_5, dok su za muškarca m_2 i ženu w_1 sparivanja S_3 i S_5 jednako povoljna. Dakle, definicija 2.3.1 ne vrijedi ni za S_3 ni za S_5 što znači da za zadani problem ne postoji optimalno sparivanje.

[1]Ovisno o tehnici popisivanja savršenih sparivanja, oznake stabilnih sparivanja mogu varirati od navedenih, ali ukupan broj stabilnih sparivanja je svakako dva. Savršena sparivanja i pripadne nestabilnosti: $S_1, (m_1, w_3); S_2, (m_1, w_3)$ i $(m_3, w_3); S_4, (m_3, w_2)$, i $(m_3, w_3); S_6, (m_2, w_1)$.

Poglavlje 3

Gale-Shapleyjev algoritam

Kontekst ostaje isti kao u prethodnom poglavlju: problem stabilnog sparivanja zadan je skupom muškaraca M i skupom žena W, $|M| = |W| = n$ te je svakom muškarcu i svakoj ženi pridružena rang-lista preferencija.

Potrebno je odgovoriti na dva konkretna pitanja:

- Postoji li stabilno sparivanje za svaki problem stabilnog sparivanja?
- Ako postoji stabilno sparivanje, je li ga moguće efektivno konstruirati uzimajući pritom u obzir rang-liste preferencija?

U navedenim pitanjima naglasak je na rang-listama preferencija jer je cilj da konstruirano stabilno sparivanje poštuje rang-liste preferencija što je više moguće. Drugim riječima, idealno stabilno sparivanje bilo bi optimalno sparivanje. U nastavku će biti opisano kako je rezultat *Gale-Shapleyjevog algoritma* stabilno sparivanje, koje je optimalno samo za prvi zadani skup, odnosno za skup M. Treba primijetiti da je u tom slučaju ispunjen samo prvi dio definicije 2.3.1, zbog čega se ne može reći da je dobiveno stabilno sparivanje optimalno: za takvo stabilno sparivanje kaže se da je *optimalno za muškarce*[1] ili da je *optimalno za prvi skup*.

Potvrdni odgovori na oba postavljena pitanja dani su konstruktivnim dokazom: Gale-Shapleyjevim algoritmom. Budući da se radi o algoritmu, njegova korektnost argumentirana je i dokazana pripadnom matematičkom analizom. U nastavku rada umjesto punog naziva Gale-Shapleyjev algoritam koristit će se skraćenica G-S algoritam.[2]

[1] U stranoj literaturi koristi se termin *men-optimal*, odnosno *women-optimal* ako je skup W prvi zadani skup.

[2] Preuzeto iz [5].

3.1 Egzistencija stabilnog sparivanja

U ovom potpoglavlju dokazano je najvažnije svojstvo G-S algoritma: egzistencija stabilnog sparivanja za svaki problem stabilnog sparivanja.

Prije navođenja pseudokoda, potrebno je uvesti neformalne termine koji se pojavljuju u G-S algoritmu ili omogućavaju jednostavnije opisivanje problema stabilnog sparivanja u zadanom kontekstu.

Za muškarca m i ženu w koji čine uređeni par (m, w) nekog sparivanja S kaže se da su *zaručeni* ili da čine *zaručeni par*. Za muškarce i žene koji nisu zaručeni kaže se da su *slobodni*. Slobodan muškarac m može *prositi* ženu w, bez obzira na to je li žena w slobodna ili zaručena, a žena w odgovara na *prosidbu* muškarca m tako što ju *prihvaća* ili *odbija*, ovisno o pridruženoj rang-listi preferencija. Muškarac m koji prosi ženu w je *prosac* žene w. Slijedi pseudokod G-S algoritma.

Algorithm 1: Gale-Shapleyjev algoritam - pseudokod

```
inicijalizacija: svi muškarci m ∈ M i sve žene w ∈ W su slobodni;
while postoji muškarac m koji je slobodan i koji nije zaprosio svaku ženu do
    izaberi takvog muškarca m;
    neka je w najviše rangirana žena na rang-listi preferencija muškarca m koju m
      dosad nije zaprosio;
    if žena w je slobodna then
        muškarac m i žena w postaju zaručeni;
    else
        neka je m′ muškarac s kojim je trenutno zaručena žena w;
        if žena w preferira svog partnera m′ više od muškarca m then
            muškarac m ostaje slobodan;
        else
            muškarac m i žena w postaju zaručeni;
            muškarac m′ postaje slobodan;
        end
    end
end
return skup S uređenih parova muškaraca i žena
```

Korisno je istaknuti sljedeće:

– Inicijalno su svi muškarci m i sve žene w slobodni, u skladu s definiranjem problema stabilnog sparivanja.

– Svaka žena w uvijek prihvaća prvu prosidbu, bez obzira na rangiranje prvog prosca m na njenoj rang-listi preferencija. Razlog za bezuvjetno prihvaćanje prve prosidbe je sljedeći: postoji mogućnost da žena w sve sljedeće prosce preferira manje od prvog prosca m. Tako bi žena w odbijanjem prvog prosca m kasnije prihvatila prosidbu nekog muškarca m' kojeg preferira manje od prvog prosca m. Budući da je žena w imala priliku zaručiti se s muškarcem m, svojim prvim proscem, takva bi odluka za ženu w bila besmislena. Dodatno, žena w će svakako prihvatiti potencijalnu prosidbu muškarca m'' kojeg preferira više od prvog prosca m. Dakle, u interesu svake žene w je da prihvati prvu prosidbu, bez obzira na rangiranje prvog prosca m.

– Ako je muškarac m zaručen s nekom ženom w, on ne prosi nijednu drugu ženu w' jer preferira svoju partnericu w više od ostalih žena koje nije zaprosio. Nadalje, ako je muškarac m postao zaručen tijekom neke iteracije petlje, u nekoj od sljedećih iteracija petlje može ponovno postati slobodan.

– Odabir slobodnog muškarca m koji će prositi neku ženu w koju još nije zaprosio je proizvoljan u svakoj iteraciji petlje.

Iščitavanjem pseudokoda G-S algoritma [1], lako je uočiti da rezultat S zbilja jest neko sparivanje - što će radi cjelovitosti biti formalno iskazano i dokazano u idućoj lemi. Ipak, nije očito da je dobiveno sparivanje S stabilno, kao ni da je optimalno za muškarce.

Lema 3.1.1. *Rezultat G-S algoritma je sparivanje.*

Dokaz. Neka je skup S rezultat G-S algoritma, odnosno neka je S skup svih zaručenih parova tijekom i nakon izvršavanja G-S algoritma. Iz pseudokoda G-S algoritma slijede dvije tvrdnje: svaki muškarac m tijekom izvršavanja svakog koraka G-S algoritma može biti zaručen s najviše jednom ženom w'; svaka žena w, jednom zaručena, bez obzira na broj raskinutih zaruka, ima točno jednog partnera m' u svakom trenutku izvršavanja G-S algoritma. Dakle, tvrdnja leme vrijedi po definiciji 2.1.1. □

Napomena 3.1.2. *Skup S zaručenih parova je sparivanje u svakom trenutku izvršavanja G-S algoritma.*

Svaki muškarac m prosi žene redoslijedom koji ovisi o rang-listi preferencija koja mu je pridružena. Muškarac m prvo prosi prvu najpoželjniju ženu w za sparivanje, a zatim potencijalno prosi drugu najpoželjniju ženu w' za sparivanje i tako redom, a svaku ženu može zaprositi najviše jednom tijekom svakog izvršavanja G-S algoritma.

Svaka žena w prihvaća ili odbija prosidbu muškarca m ovisno o rang-listi preferencija koja joj je pridružena, s iznimkom za prvu prosidbu. Iduće dvije leme opisuju razlike između muškaraca i žena tijekom izvršavanja G-S algoritma, a navedene su se bez dokaza jer su očite iz pseudokoda G-S algoritma i načina zadavanja problema stabilnog sparivanja.

Lema 3.1.3. *Muškarac m može alternirati između stanja slobodan i zaručen. Svakom idućom prihvaćenom prosidbom, muškarac m postaje zaručen sa ženom w' koju preferira manje od žene w koju je prethodno zaprosio.*

Lema 3.1.4. *Žena w ostaje zaručena od trenutka prve prosidbe. Prihvaćanjem svake iduće prosidbe, žena w postaje zaručena s muškarcem m' kojeg preferira više od prethodnog prosca m.*

Sljedeća lema potrebna je za dokazivanje da je rezultat G-S algoritma savršeno sparivanje, što je jedan od nužna dva uvjeta za stabilno sparivanje, po definiciji 2.2.6.

Lema 3.1.5. *Neka tijekom izvršavanja G-S algoritma postoji slobodan muškarac m. Tada postoji žena w koju muškarac m nije zaprosio.*

Dokaz. Kontradikcijom. Neka u nekom trenutku izvršavanja G-S algoritma postoji slobodan muškarac m koji je zaprosio sve žene. Muškarac m nije mogao biti prvi prosac svakoj ženi, ali ako je zaprosio svaku ženu, tada je svaka žena u nekom trenutku, zaključno s pretpostavljenim, primila prvu prosidbu. Po lemi 3.1.4, svaka žena w ostaje zaručena od trenutka prve prosidbe iz čega slijedi da u pretpostavljenom trenutku postoji n zaručenih parova u skupu S. Budući da je muškarac m slobodan, vrijedi $|M \backslash \{m\}| = n - 1$, što je kontradikcija s tvrdnjom da je $|S| = n$. Dakle, tvrdnja leme je dokazana. □

Napomena 3.1.6. *U prethodnoj lemi tvrdnja ne glasi: postoji žena koju muškarac m još nije zaprosio, jer nije nužno da neki, ili pak svi, muškarci zaprose sve žene kako bi se dobilo stabilno sparivanje.*

Teorem 3.1.7. *Rezultat G-S algoritma je savršeno sparivanje.*

Dokaz. Neka je skup S rezultat G-S algoritma. Po lemi 3.1.1, skup S je sparivanje. Slijedi dokaz da je sparivanje S savršeno.
Kontradikcijom, za dva slučaja. Neka je sparivanje S rezultat G-S algoritma koje nije savršeno. Budući da je skup S sparivanje, nije moguće da neki muškarac m istovremeno bude zaručen s više žena, kao ni da neka žena w istovremeno bude zaručena s više muškaraca. Preostaje ispitati slučaj kada postoje neki muškarci, odnosno žene, koji su slobodni. Kako za muškarce i žene ne vrijede ista pravila tijekom izvođenja G-S algoritma, radi se o dva različita slučaja te će za svaki slučaj biti dokazano da tvrdnja vrijedi.
Za prvi slučaj: neka S nije savršeno sparivanje takvo da postoji barem jedan muškarac m koji se ne nalazi ni u jednom uređenom paru sparivanja S, odnosno koji je na kraju izvršavanja G-S algoritma slobodan. Budući da je G-S algoritam izvršen, slijedi da je muškarac m zaprosio sve žene - inače izvršavanje G-S algoritma ne bi bilo završeno, zbog

uvjeta petlje while. Ipak, po lemi 3.1.5, postoji žena w koju slobodan muškarac m nije zaprosio, što je kontradikcija s pretpostavkom o završetku izvršavanja G-S algoritma. Dakle, S je savršeno sparivanje.
Za drugi slučaj: neka S nije savršeno sparivanje takvo da postoji barem jedna žena w koja se ne nalazi ni u jednom uređenom paru sparivanja S, odnosno koja je na kraju izvođenja G-S algoritma slobodna. Budući da je G-S algoritam izvršen, po uvjetu petlje while slijedi da ne postoji slobodan muškarac koji je zaprosio sve žene. Kako je žena w slobodna, po lemi 3.1.4, slijedi da žena w nije primila prvu prosidbu. Budući da je S sparivanje i vrijedi $|M| = |W|$, mora postojati barem jedan slobodan muškarac m koji nije ni u jednom uređenom paru sparivanja S. Kako je m slobodan, a žena w nije primila prvu prosidbu, uvjet petlje while je ispunjen pa G-S algoritam sigurno nije izvršen do kraja, što je kontradikcija s pretpostavkom o završetku izvršavanja G-S algoritma. Dakle, S je savršeno sparivanje. □

Napomena 3.1.8. *Kako se prilikom svake iteracije petlje slobodan muškarac odabire proizvoljno, G-S algoritam će se različito izvršavati za isti ulaz.*[3] *Zbog toga se u daljnjoj analizi u obzir uzima izvršavanje G-S algoritma, a ne sâm G-S algoritam. Iz tog razloga, sljedeći teorem ne glasi: Rezultat G-S algoritma je stabilno sparivanje.*

Idućim teoremom dokazano je najvažnije svojstvo G-S algoritma.

Teorem 3.1.9. *Rezultat svakog izvršavanja G-S algoritma je stabilno sparivanje.*

Dokaz. Neka je skup S rezultat G-S algoritma. Po definiciji 2.2.6, treba pokazati da je S savršeno sparivanje i da ne sadrži nestabilnosti. Kako iz teorema 3.1.7 slijedi da je S savršeno sparivanje preostaje dokazati da S ne sadrži nestabilnosti.
Kontradikcijom. Neka je savršeno sparivanje S rezultat G-S algoritma i neka je (m, w) nestabilnost u S. Tada postoje dva uređena para (m, w'), (m', w) u S takvi da muškarac m preferira ženu w više od svoje partnerice w', a žena w preferira muškarca m više od svog partnera m'. Budući da je S rezultat na kraju izvršavanja G-S algoritma, zaključuje se da je muškarac m zaprosio ženu w' posljednju. S obzirom na nestabilnost (m, w), postavlja se pitanje: Je li muškarac m zaprosio ženu w prije žene w'? Ako je odgovor negativan, tada muškarac m preferira svoju partnericu w' više od žene w, što je kontradikcija s pretpostavkom da je (m, w) nestabilnost u S. Ako je pak odgovor potvrdan, tada je žena w odbila prosidbu muškarca m u korist nekog muškarca m''. Budući da je na kraju izvršavanja G-S algoritma žena w zaručena s muškarcem m', moguće su dvije opcije: prva je da vrijedi $m'' = m'$, a druga, po lemi 3.1.4, da žena w preferira muškarca m' više od muškarca m''. Oba slučaja vode do kontradikcije s pretpostavkom da žena w preferira muškarca m više od muškarca m'. Dakle, S je stabilno sparivanje čime je dokazana tvrdnja teorema. □

[3] U stranoj literaturi koristi se termin neodređen (engl. *under-specified).*

3.2 Optimalnost za prvi skup

Tijekom dokazivanja drugog važnog svojstva G-S algoritma: rezultat svakog izvršavanja G-S algoritma je stabilno sparivanje optimalno za muškarce, odgovorit će se i na pitanje koje se prirodno javlja iz napomene 3.1.8 i teorema 3.1.9:

– Je li rezultat svih izvršavanja G-S algoritma uvijek isto stabilno sparivanje?

Budući da se izvršavanja G-S algoritma razlikuju za isti ulaz, za očekivati je da će se razlikovati i dobiveni rezultati: odabir slobodnog muškarca m umjesto slobodnog muškarca m' imat će za posljedicu različite prošnje i odbijanje istih, a onda i različite zaručene parove tijekom različitih izvršavanja G-S algoritma. U nastavku je objašnjeno kako unatoč razlikama sva izvršavanja G-S algoritma daju za rezultat isto stabilno sparivanje.

Jedan od poznatijih i jednostavnijih načina za dokazivanje spomenutog je jedinstvena karakterizacija rezultata jednog izvršavanja G-S algoritma, nakon čega se dokazuje da su svi rezultati dobiveni ostalim izvršavanjima G-S algoritma jednaki karakteriziranom rezultatu. Za karakterizaciju je potrebno uvesti nekoliko novih pojmova i skup S^*.

Definicija 3.2.1. *Kaže se da je žena w **valjana partnerica** muškarca m ako postoji stabilno sparivanje koje sadrži uređeni par (m, w).*

Definicija 3.2.2. *Kaže se da je žena w **najbolja valjana partnerica** muškarca m, ako je w valjana partnerica muškarca m i ako nijedna žena w' koju muškarac m preferira više od žene w nije njegova valjana partnerica.*

Skup S^* definira se na sljedeći način: $S^* = \{(m, b(m)) : m \in M\}$, gdje je $b(m)$ najbolja valjana partnerica muškarca m. Idući teorem povezuje rezultat G-S algoritma, stabilna sparivanja i skup S^*. Također, teoremom se razjašnjavaju tri značajne tvrdnje:

– Prvo, skup S^* je stabilno sparivanje. S obzirom na način zadavanja skupa S^* nije očito ni da je S^* sparivanje. Kao opreku tvrdnji da je S^* sparivanje, razumno je postaviti pitanje: Zašto problem stabilnog sparivanja ne bi bio zadan tako da dva muškarca m, m' imaju istu najbolju valjanu partnericu w?

– Drugo, rezultat svakog izvršavanja G-S algoritma je takav da za svakog muškarca m vrijedi da ne postoji neko drugo stabilno sparivanje u kojem bi muškarac m mogao biti zaručen sa ženom w' koju preferira više od svoje najbolje valjane partnerice w.

– Treće, redoslijed prosidbi tijekom izvršavanja G-S algoritma ne utječe na konačan rezultat, čime se potvrdno odgovara na zadnje postavljeno pitanje: rezultat svih izvršavanja G-S algoritma je isto stabilno sparivanje.

Teorem 3.2.3. *Rezultat svakog izvršavanja G-S algoritma je skup S^*.*

Dokaz. Kontradikcijom. Neka je $\mathcal{E}$ neko izvršavanje G-S algoritma takvo da njegov rezultat nije skup S^*. Tada je rezultat izvršavanja $\mathcal{E}$ stabilno sparivanje S u kojem postoji barem jedan muškarac m koji nije zaručen sa svojom najboljom partnericom w. Tada je muškarac m zaručen sa ženom w', koja je njegova valjana partnerica po definiciji 3.2.1. Treba razmotriti prvi trenutak, u izvršavanju $\mathcal{E}$, u kojem je nekog muškarca m' odbila njegova najbolja valjana partnerica $b(m')$.
Neka je prosidbu muškarca m odbila žena w'', takva da je ona *prva* njegova valjana partnerica koja je odbila njegovu prosidbu. Budući da muškarac m prosi žene s obzirom na svoju rang-listu preferencija, slijedi da je žena w'' najbolja valjana partnerica muškarca m, to jest vrijedi: $w'' = b(m) = w$.
Žena w odbila je prosidbu muškarca m iz jednog od sljedeća dva razloga: ili je žena w bila zaručena s muškarcem m' kojeg preferira više od muškarca m, ili je žena w prihvatila prosidbu muškarca m te ga kasnije odbila u korist prosidbe muškarca m''. Irelevantno iz kojeg od navedena dva razloga, u razmatranom trenutku tijekom izvršavanja $\mathcal{E}$ žena w je odbila muškarca m i zaručena je s muškarcem m'.
Kako je žena w valjana partnerica muškarca m, po definiciji 3.2.1, postoji stabilno sparivanje S' koje sadrži zaručeni par (m, w); potrebno je odrediti partnericu muškarca m' u tom sparivanju. Neka je muškarac m' u sparivanju S' zaručen sa ženom w'; vrijedi $w' \neq w$ budući da je $(m, w) \in S'$. S obzirom na to da je u prethodno promatranom izvršavanju G-S algoritma, izvršavanju $\mathcal{E}$, prvo odbijanje nekog muškarca m'' od strane njegove valjane partnerice $b(m'')$ bilo upravo odbijanje muškarca m od žene w, znači da do tog trenutka muškarca m' nije odbila nijedna njegova valjana partnerica. Kako je u tom trenutku m' zaručen sa ženom w, slijedi da muškarac m' preferira ženu w više od svoje konačne partnerice w'. Iz prethodnog dijela dokaza vrijedi da žena w preferira muškarca m' više od muškarca m. Dakle, nestabilnost $(m', w) \in S'$, što je kontradikcija sa stabilnošću sparivanja S', što je kontradiktorno početnoj pretpostavci. Time je dokazana tvrdnja teorema. □

Napomena 3.2.4. *U prethodnom dokazu, trenutci koji su od značaja za muškarca m, a koji su prethodili istaknutom trenutku izvršavanja $\mathcal{E}$, su trenutci u kojima su prosidbu muškarca m prihvatile ili odbile žene koje nisu njegove valjane partnerice, trenutak u kojem je muškarac m zaprosio svoju najbolju valjanu partnericu $b(m)$ te potencijalno trenutak u kojem je $b(m)$ prihvatila prosidbu muškarca m.*

Napomena 3.2.5. *Zaključak iz prethodnog dokaza: $w'' = b(m) = w$, vrijedi zbog sljedećeg. Muškarac m prosi žene s obzirom na svoju rang-listu preferencija. Najbolju valjanu partnericu zaprosit će prije bilo koje druge valjane partnerice jer preferira svoju najbolju valjanu partnericu $b(m)$ više od ostalih valjanih partnerica, po definiciji 3.2.2. Dakle, prva valjana partnerica koja odbija prosidbu muškarca m mora biti njegova najbolja valjana partnerica $b(m)$ jer je ona prva valjana partnerica koju će muškarac m zaprositi.*

Za skup žena, to jest za drugi zadani skup problema stabilnog sparivanja, vrijedi sljedeće kada se radi o rezultatu G-S algoritma: nijedna žena w ne može proći gore u nekom drugom stabilnom sparivanju no što je prošla u stabilnom sparivanju S^*. Odnosno, ne postoji neko stabilno sparivanje u kojem bi žena w bila zaručena s muškarcem m' kojeg preferira manje od muškarca m koji je njen partner u sparivanju S^*. Sukladno tome, definiraju se sljedeća dva pojma i dokazuje teorem.

Definicija 3.2.6. *Kaže se da je muškarac m **valjani partner** žene w ako postoji stabilno sparivanje koje sadrži uređeni par (m, w).*

Definicija 3.2.7. *Kaže se da je muškarac m **najgori valjani partner** žene w, ako je m valjani partner žene w i ako nijedan muškarac m' kojeg žena w preferira manje od muškarca m nije njen valjani partner.*

Teorem 3.2.8. *U stabilnom sparivanju S^*, svaka žena w je zaručena sa svojim najgorim valjanim partnerom.*

Dokaz. Kontradikcijom. Neka je S^* stabilno sparivanje takvo da za uređeni par $(m, w) \in S^*$ vrijedi da muškarac m nije najgori valjani partner žene w. Tada po definicijama 3.2.6 i 3.2.7, postoji stabilno sparivanje S' u kojem je žena w zaručena s muškarcem m' kojeg preferira manje od muškarca m. Muškarac m je u S' zaručen s nekom ženom $w' \neq w$. Kako je $(m, w) \in S^*$, žena w je najbolja valjana partnerica muškarca m iz čega slijedi da muškarac m preferira ženu w više od žene w', svoje partnerice u S'. Sada za stabilno sparivanje S' vrijedi: $(m, w'), (m', w) \in S'$ su takvi da muškarac m preferira ženu w više od svoje partnerice w' i žena w preferira muškarca m više od svog partnera m'. Dakle, (m, w) je nestabilnost u S' što je kontradikcija s pretpostavkom da je S' stabilno sparivanje. □

Napomena 3.2.9. *U slučaju kada je problem stabilnog sparivanja definiran skupom žena W i skupom muškaraca M, odnosno u slučaju kada je prvi zadani skup skup žena, a drugi zadani skup skup muškaraca, za sve navedeno u poglavlju vrijedi obrnuta situacija te bi rezultat G-S algoritma bilo stabilno sparivanje S_w^* optimalno za žene.*

Skup S^* nije nužno jedino stabilno sparivanje zadanog problema stabilnog sparivanja. Ipak, S^* jest jedino stabilno sparivanje koje će G-S algoritam konstruirati prilikom svakom izvršavanja, prema teoremu 3.2.3. Slijedi primjer.

Primjer 3.2.10. Neka je problem stabilnog sparivanja zadan na sljedeći način: skupom muškaraca $M = \{m_1, m_2, m_3\}$, skupom žena $W = \{w_1, w_2, w_3\}$ i matricama preferencija muškaraca m_i i žena w_i:

$$\begin{array}{c} m_1: \\ m_2: \\ m_3: \end{array} \begin{bmatrix} w_2 & w_1 & w_3 \\ w_3 & w_2 & w_1 \\ w_1 & w_2 & w_3 \end{bmatrix}, \quad \begin{array}{c} w_1: \\ w_2: \\ w_3: \end{array} \begin{bmatrix} m_2 & m_1 & m_3 \\ m_3 & m_1 & m_2 \\ m_1 & m_2 & m_3 \end{bmatrix}. \tag{3.1}$$

Lako se uoči da je rezultat svakog izvršavanja G-S algoritma stabilno sparivanje, optimalno za skup M, $S^* = \{(m_1, w_2), (m_2, w_3), (m_3, w_1)\}$.
Sukladno napomeni 3.2.9, kada bi prvi zadani skup problema stabilnog sparivanja, uz iste matrice preferencija, bio skup žena W, rezultat izvršavanja G-S algoritma bilo bi stabilno sparivanje optimalno za skup W, $S_w^* = \{(w_1, m_2), (w_2, m_3), (w_3, m_1)\}$.
Ostala stabilna sparivanja za zadani problem stabilnog sparivanja - bez obzira na prvi zadani skup - po teoremu 2.3.2 nisu optimalna ni za muškarce ni za žene. Jedno takvo stabilno sparivanje je skup $S = \{(m_1, w_1), (m_2, w_3), (m_3, w_2)\}$. Provjera da je S stabilno sparivanje provodi se analogno kao u primjeru 2.2.8.

3.3 Vremenska složenost

Prilikom određivanja vremenske složenosti algoritma - vremena potrebnog za izvršavanje algoritma - u obzir se uzimaju veličina ulaza i relevantne operacije. Čitanjem pseudokoda G-S algoritma [1], uočavaju se tri potencijalno relevantne operacije: prosidba, prihvaćanje prosidbe s pripadnim posljedicama te odbijanje prosidbe s pripadnim posljedicama. Operacija prosidbe u pseudokodu uključuje odabir slobodnog muškarca m te pronalazak žene w koju muškarac m nije zaprosio, a koju preferira više od svih ostalih žena koje nije zaprosio.

Iz navedenih operacija slijedi da je, za određivanje vremenske složenosti, potrebno tijekom izvršavanja G-S algoritma brojiti jednu od triju mjera: parove (m, w), gdje je m muškarac koji je zaprosio ženu w, zaručene parove (m', w') ili slobodne muškarce. Usporedbom navedenih mjera uočava se da je jedina moguća *mjera napretka*[4] par (m, w), gdje je m muškarac koji je zaprosio ženu w. Naime, to je jedina mjera koja tijekom izvršavanja G-S algoritma konstantno raste: svakom iteracijom petlje povećava se točno za jedan.

Napomena 3.3.1. *Za mjeru napretka par (m, w), gdje je m muškarac koji je zaprosio ženu w, nije važno kako je žena w odgovorila na prosidbu muškarca m.*

G-S algoritam pronalazi stabilno sparivanje u vremenu $O(n^2)$, dakle izvršava se u polinomnom vremenu. Slijedi pripadni teorem.

Teorem 3.3.2. *G-S algoritam završava nakon najviše n^2 iteracija petlje while.*

Dokaz. Izvršavanje G-S algoritma ovisi o uvjetu petlje while. Operacija prosidbe izvršava se prilikom svake iteracije petlje while pri čemu se broj parova (m, w), gdje je m muškarac koji je zaprosio ženu w, povećava za jedan. Za svaki zadani ulaz, broj muškaraca iznosi

[4] Pronalazak mjere napretka (engl. *measure of progress*) je strategija za određivanje gornje granice izvršavanja algoritma.

n, kao i broj žena, iz čega slijedi da mogući broj parova (m, w), gdje je m muškarac koji je zaprosio ženu w, iznosi najviše n^2 jer je moguće da tijekom izvršavanja G-S algoritma svaki od n muškaraca zaprosi svaku od n žena najviše jednom. Kako je n^2 ujedno i najveći mogući broj iteracija petlje while, tvrdnja teorema je dokazana. □

Napomena 3.3.3. *Za probleme stabilnog sparivanja zadane rang-listama preferencija takvim da za svakog muškarca vrijedi da njegova prva najpoželjnija žena za sparivanje nije prva najpoželjnija žena za sparivanje ni za jednog drugog muškarca, prilikom svakog izvršavanja G-S algoritma broj zaručenih parova strogo raste nakon svake iteracije. U primjeru 3.2.10 zadan je jedan takav problem stabilnog sparivanja.*

Za svaki problem stabilnog sparivanja, ulaz G-S algoritma čine dvije matrice preferencija: prva matrica preferencija sadrži n rang-listi preferencija muškaraca, a druga matrica preferencija sadrži n rang-listi preferencija žena. Dakle, ukupan broj ulaznih podataka iznosi $2n^2$. Nadalje, za svaki problem stabilnog sparivanja *prostor pretraživanja*[5] iznosi $n!$, što znači da za svaki problem stabilnog sparivanja postoji $n!$ savršenih sparivanja među kojima G-S algoritam pronalazi ono koje je optimalno za prvi skup.

3.4 Nestabilnosti

Zanimljivo je razmotriti sljedeću generalizaciju problema stabilnog sparivanja: problem stabilnog sparivanja zadan skupom muškaraca M, skupom žena W, skupom *zabranjenih parova* $F \subseteq M \times W$ te pripadnim rang-listama preferencija, pri čemu je kardinalitet skupova M i W isti. Rang-liste preferencija nisu potpune jer ne sadržavaju zabranjene partnere, odnosno zabranjene partnerice, s iznimkom za slučaj $F = \emptyset$. Formalno, rang-lista preferencija pridružena muškarcu m sadrži sve žene w tako da vrijedi $(m, w) \notin F$ i rang-lista preferencija pridružena ženi w' sadrži sve muškarce m' tako da vrijedi $(m', w') \notin F$.

Uz definiciju pojma nestabilnosti 2.2.5 korištenu u dosadašnjem dijelu rada, potrebno je definirati dodatne tri vrste nestabilnosti:

- Postoji slobodan muškarac koji je poželjniji od partnera i nije zabranjen.
- Postoji slobodna žena koja je poželjnija od partnerice i nije zabranjena.
- Postoje slobodan muškarac i slobodna žena koji međusobno nisu zabranjeni.

[5] engl. *search space*

Slijede formalne definicije navedenih nestabilnosti. Radi potpunosti, navodi se i već poznata nestabilnost, ali u kontekstu zadanog problema stabilnog sparivanja. U svrhu jednostavnijeg raspoznavanja, nestabilnostima se dodjeljuje vrsta.[6]

Definicija 3.4.1. *Neka je zadano sparivanje S. Neka su $(m, w), (m', w') \in S$ sa svojstvom da muškarac m preferira ženu w' više od svoje partnerice w i da žena w' preferira muškarca m više od svog partnera m'. Tada se kaže da je (m, w')* ***nestabilnost prve vrste*** *u S.*

Definicija 3.4.2. *Neka je zadano sparivanje S. Neka je $(m, w) \in S$ i neka je muškarac m' slobodan. Kaže se da je (m', w)* ***nestabilnost druge vrste*** *u S ako žena w preferira muškarca m' više od svog partnera m i ako vrijedi $(m', w) \notin F$.*

Definicija 3.4.3. *Neka je zadano sparivanje S. Neka je $(m, w) \in S$ i neka je žena w' slobodna. Kaže se da je (m, w')* ***nestabilnost treće vrste*** *u S ako muškarac m preferira ženu w' više od svoje partnerice w i ako vrijedi $(m, w') \notin F$.*

Definicija 3.4.4. *Neka je zadano sparivanje S. Neka su muškarac m i žena w slobodni. Kaže se da je (m, w)* ***nestabilnost četvrte vrste*** *u S ako vrijedi $(m, w) \notin F$.*

Napomena 3.4.5. *Za ovako zadan problem stabilnog sparivanja, stabilno sparivanje S ne mora biti savršeno, ali ne smije sadržavati niti jednu od četiriju navedenih nestabilnosti. Dakle, prvi uvjet definicije 2.2.6 ne mora biti zadovoljen.*

Iduće četiri leme i pripadni dokazi da sparivanje S dobiveno svakim izvršavanjem G-S algoritma, za problem stabilnog sparivanja opisan u ovom potpoglavlju, ne sadrži niti jednu od navedenih nestabilnosti objedinjeni su teoremom 3.4.10.

Lema 3.4.6. *Rezultat svakog izvršavanja G-S algoritma ne sadrži nestabilnost prve vrste.*

Dokaz. Slijedi iz teorema 3.1.9. □

Lema 3.4.7. *Rezultat svakog izvršavanja G-S algoritma ne sadrži nestabilnost druge vrste.*

Dokaz. Kontradikcijom. Neka je sparivanje S rezultat nekog izvršavanja G-S algoritma koje sadrži nestabilnost druge vrste $(m', w) \notin F$. Tada postoje $(m, w) \in S$ i slobodan muškarac m' kojeg žena w preferira više od svog partnera m. Budući da je izvršavanje G-S algoritma završeno, slijedi da uvjet petlje while nije ispunjen, odnosno slijedi da je muškarac m' zaprosio sve žene, uključujući ženu w. Kako je $(m, w) \in S$, slijedi da je žena w odbila muškarca m' u korist nekog muškarca m'', što je u kontradikciji s pretpostavkom da je na kraju izvršavanja G-S algoritma žena w zaručena s muškarcem m kojeg preferira manje od muškarca m', bez obzira vrijedi li $m'' = m$ ili $m'' \neq m$, čime je dokaz gotov. □

[6] U stranoj literaturi nestabilnosti se navode isključivo opisno.

Lema 3.4.8. *Rezultat svakog izvršavanja G-S algoritma ne sadrži nestabilnost treće vrste.*

Dokaz. Kontradikcijom. Neka je sparivanje S rezultat nekog izvršavanja G-S algoritma koje sadrži nestabilnost treće vrste $(m, w') \notin F$. Tada postoje $(m, w) \in S$ i slobodna žena w' koju muškarac m preferira više od svoje partnerice w. Muškarac m je zaprosio ženu w' prije svoje partnerice w. Po lemi 3.1.4, žena w' nakon prve prosidbe više nikada neće biti slobodna što je kontradikcija s pretpostavkom, bez obzira je li muškarac m prvi prosac žene w' ili jedan od narednih. Tvrdnja leme je dokazana. □

Lema 3.4.9. *Rezultat svakog izvršavanja G-S algoritma ne sadrži nestabilnost četvrte vrste.*

Dokaz. Kontradikcijom. Neka je sparivanje S rezultat nekog izvršavanja G-S algoritma koje sadrži nestabilnost četvrte vrste $(m, w) \notin F$. Tada su muškarac m i žena w slobodni. Kako je izvršavanje G-S algoritma završeno, slijedi da je muškarac m zaprosio sve žene, uključujući ženu w, koja po lemi 3.1.4 ne može biti slobodna što je kontradikcija s pretpostavkom, čime je tvrdnja leme dokazana. □

Teorem 3.4.10. *Neka je problem stabilnog sparivanja zadan skupom muškaraca M, skupom žena W, pri čemu je $|M| = |W|$, skupom zabranjenih parova $F \subseteq M \times W$ te pripadnim rang-listama preferencija. Rezultat svakog izvršavanja G-S algoritma je stabilno sparivanje S.*

Dokaz. Slijedi iz napomene 3.4.5 i prethodnih četiriju lema: 3.4.6, 3.4.7, 3.4.8, 3.4.9. □

Dakle, za problem stabilnog sparivanja koji uključuje zabranjene parove, G-S algoritam bez ikakvih preinaka pronalazi stabilno sparivanje - koje ne mora biti savršeno.

3.5 Primjer preinake

U ovom potpoglavlju, koristeći narednu generalizaciju problema stabilnog sparivanja, pokazat će se prilagodljivost G-S algoritma dodatnim zahtjevima kojima problem stabilnog sparivanja može biti zadan.

Problem stabilnog sparivanja zadan je skupom bolnica H, skupom studenata medicine N i pripadnim rang-listama preferencija čija se veličina međusobno može razlikovati. Uz pripadnu rang-listu preferencija, svakoj bolnici h pridružen je broj mjesta za stažiranje q. Pretpostavka je da ukupan broj studenata premašuje ukupan broj mjesta za stažiranje zbog čega neki studenti neće dobiti mjesto za stažiranje pa stabilno sparivanje u ovom slučaju sigurno neće biti savršeno. Dakle, radi se o osnovnom problemu stabilnog sparivanja uz dodatne zahtjeve da bolnice u pravilu žele dobiti više od jednog stažista te da postoji višak studenata. Slijedi definicija stabilnog sparivanja za ovaj problem stabilnog sparivanja.

Definicija 3.5.1. *Za sparivanje S kaže se da je stabilno ako vrijedi:*

- *Ne postoje $(h, s), (h', s') \in S$ tako da bolnica h preferira studenta s' više od studenta s i da student s' preferira bolnicu h više od bolnice h'.*
- *Ne postoji par $(h, s) \in S$ i student s' koji nema mjesto za stažiranje tako da bolnica h preferira studenta s' više od studenta s.*[7]

U nastavku slijedi pseudokod G-S algoritma s preinakama koji pronalazi stabilno sparivanje za problem stabilnog sparivanja definiran u ovom potpoglavlju.[8]

Algorithm 2: Gale-Shapleyjev algoritam s preinakama

```
inicijalizacija: sva mjesta za stažiranje svih bolnica h ∈ H su slobodna i nijedan
  student s ∈ N nema mjesto za stažiranje;
while postoji bolnica h koja nije popunila sva mjesta za stažiranje i koja nije
  ponudila svim studentima sva svoja mjesta za stažiranje do
    izaberi takvu bolnicu h;
    neka je p_h slobodno mjesto za stažiranje bolnice h;
    neka je s najviše rangiran student na rang-listi preferencija bolnice h kojem h
      dosad nije ponudila mjesto za stažiranje p_h i koji nema neko drugo mjesto za
      stažiranje u bolnici h;
    if student s nema mjesto za stažiranje then
        student s prihvaća mjesto za stažiranje p_h u bolnici h;
        broj slobodnih mjesta za stažiranje bolnice h smanjuje se za jedan;
    else
        neka je h' bolnica kod koje je student s prihvatio mjesto za stažiranje p_h';
        if student s preferira bolnicu h' više od bolnice h then
            student s ostaje na mjestu za stažiranje p_h' bolnice h';
            broj slobodnih mjesta za stažiranje bolnice h ostaje isti;
        else
            student s prihvaća mjesto za stažiranje p_h u bolnici h;
            broj slobodnih mjesta za stažiranje bolnice h smanjuje se za jedan;
            broj slobodnih mjesta za stažiranje bolnice h' povećava se za jedan;
        end
    end
end
return skup S uređenih parova bolnica i studenata
```

[7] Odnosno, kaže se da je sparivanje S stabilno ako ne sadrži nestabilnost prve vrste i nestabilnost treće vrste (definicije 3.4.1, 3.4.3); uz promjenu korištene terminologije.

[8] Dokazi se provode na sličan način kao u prethodnim dijelovima poglavlja.

Poglavlje 4

Praktična primjena

Cilj poglavlja je pokazati primjenu problema stabilnog sparivanja na primjeru problema u kojem skupovi i rang-liste preferencija nisu eksplicitno zadani.

4.1 Opis problema

Jedna tvrtka posjeduje n brodova i pruža usluge u n luka. Za neki period, svakom brodu dodjeljuje se raspored posjeta lukama u danima perioda. Za broj dana perioda, u oznaci m, vrijedi $m > n$. Tijekom perioda svaki brod s posjećuje svaku luku p na točno jedan dan; ako u nekom danu brod s ne posjećuje nijednu luku, tada plovi. Svi rasporedi brodova, zadani za neki period, ispunjavaju uvjet:

$$\text{Nikoja dva broda ne smiju se nalaziti u istoj luci istog dana.} \tag{4.1}$$

Tvrtka želi obaviti održavanje na svim brodovima tijekom nekog perioda koristeći pri tom skraćeni raspored za svaki brod: brod s pristat će u neku luku p te neće posjetiti preostale luke sa svog rasporeda, ako takve postoje. Dakle, skraćeni raspored broda s sastoji se od prvotnog rasporeda broda s zaključno s nekim danom u kojem brod s posjećuje luku p; luka p je odredište broda s u kojoj on ostaje ostatak perioda.

Treba pokazati da za svaki zadani skup rasporeda uvijek postoji skup skraćenih rasporeda tako da navedeni uvjet ostane ispunjen te konstruirati algoritam koji pronalazi skraćene rasporede.

Napomena 4.1.1. *Svaki skraćeni raspored mora završiti lukom - ne plovidbom - i sve odredišne luke skraćenih rasporeda moraju biti različite. Svaki prvotni raspored je duljine m. Svaki skraćeni raspored je duljine najviše $m - 1$, inače nije skraćen.*

Primjer 4.1.2. Neka se period sastoji od četiri dana i neka su za brodove s_1, s_2 i s_3 zadani sljedeći rasporedi za posjete lukama p_1, p_2 i p_3:

- za brod s_1: p_1, plovidba, p_2, p_3
- za brod s_2: p_2, p_3, plovidba, p_1
- za brod s_3: plovidba, p_1, p_3, p_2.

Skraćeni rasporedi brodova su:

- za brod s_1: p_1, plovidba, p_2
- za brod s_2: p_2, p_3
- za brod s_3: plovidba, p_1.

Prvog dana brod s_1 posjećuje luku p_1, brod s_2 posjećuje luku p_2, a brod s_3 plovi. Drugog dana brod s_1 plovi, brod s_2 posjećuje luku p_3 koja je njegovo odredište, a brod s_3 posjećuje luku p_1 koja je njegovo odredište. Brodovi s_2 i s_3 ostaju u svojim odredištima ostatak perioda, to jest trećeg i četvrtog dana. Trećeg dana brod s_1 posjećuje luku p_2 koja je njegovo odredište i ostaje u njoj do kraja perioda, to jest četvrti dan.
Sljedeći rasporedi nisu primjer traženog skraćivanja rasporeda:

- za brod s_1: p_1
- za brod s_2: p_2
- za brod s_3: plovidba, p_1, p_3.

Odredišne luke brodova su različite, no brod s_3 drugog dana posjećuje luku p_1 koja je odredište broda s_1 od prvog dana. Dakle, drugog dana brodovi s_1 i s_3 nalaze se u luci p_1 pa traženi uvjet 4.1 nije ispunjen.

4.2 Analiza problema, egzistencija rješenja

Sadržavanje svih luka u rasporedu svakog broda i postojanje n-članih skupova brodova i luka, $S = \{s_1, s_2, ..., s_n\}$ i $P = \{p_1, p_2, ..., p_n\}$, naznake su za mogućnost svođenja opisanog problema na problem stabilnog sparivanja. Potrebno je odabrati koji će skup biti prvi, odnosno drugi, te kreirati rang-liste preferencija; oboje je potrebno učiniti u skladu s ciljem problema: skraćivanje rasporeda, odnosno pronalazak odredišta za svaki brod tako da zadani uvjet 4.1 ostane ispunjen.

U trećem poglavlju dokazano je da G-S algoritam, za svaki problem stabilnog sparivanja[1], daje rezultat kojim najviše profitira prvi skup (teorem 3.2.3), dok je drugi skup zakinut najviše moguće (teorem 3.2.8). Navedeno su smjernice za određivanje redoslijeda skupova i kreiranje pripadnih rang-listi preferencija.

S obzirom na način zadavanja rasporeda, za prvi zadani skup odabire se skup brodova, a za drugi skup skup luka. Kreirane rang-liste preferencija trebaju omogućiti skraćivanje rasporeda brodova: prednost pri odabiru odredišta pojedinog broda imaju luke koje bi brod, po svom rasporedu, posjetio ranije.

Rang-lista preferencija svakog broda kreira se iz njegova rasporeda zanemarivanjem plovidbe čime redoslijed posjeta lukama ostaje isti. Rang-liste preferencija luka trebaju biti takve da svaka luka ima što manje šanse postati odredište onim brodovima koji bi ju kasnije posjetili - svaka luka treba najviše preferirati brod koji bi ju, po zadanim rasporedima brodova, posjetio posljednji. Dakle, rang-lista preferencija luke kreira se prema načelu: što je dan posjete broda luci kasniji, to luka više preferira brod.

Primjer 4.2.1. Rasporedi brodova zadani u primjeru 4.1.2 mogu se prikazati matrično, uz zamjenu oznake za plovidbu nulom radi preglednosti:

$$\begin{array}{c} s_1: \\ s_2: \\ s_3: \end{array} \begin{bmatrix} p_1 & 0 & p_2 & p_3 \\ p_2 & p_3 & 0 & p_1 \\ 0 & p_1 & p_3 & p_2 \end{bmatrix}. \tag{4.2}$$

Skupovima brodova i luka, $S = \{s_1, s_2, s_3\}$ i $P = \{p_1, p_2, p_3\}$, pridodaju se matrice preferencija brodova i luka kreirane iz matrice rasporeda brodova 4.2:

$$\begin{array}{c} s_1: \\ s_2: \\ s_3: \end{array} \begin{bmatrix} p_1 & p_2 & p_3 \\ p_2 & p_3 & p_1 \\ p_1 & p_3 & p_2 \end{bmatrix}, \quad \begin{array}{c} p_1: \\ p_2: \\ p_3: \end{array} \begin{bmatrix} s_2 & s_3 & s_1 \\ s_3 & s_1 & s_2 \\ s_1 & s_3 & s_2 \end{bmatrix}. \tag{4.3}$$

Rang-liste preferencija brodova odgovaraju prvotnim rasporedima brodova - uz izbacivanje oznaka plovidbe, odnosno nula. Rang-liste preferencija luka sadržavaju brodove u obrnutom redoslijedu njihova posjećivanja. Primjerice, rang-lista preferencija luke p_2: na prvom mjestu nalazi se brod s_3 koji bi luku p_2 trebao posjetiti četvrtog dana, na drugom mjestu nalazi se brod s_1 koji bi luku p_2 trebao posjetiti trećeg dana, a na trećem mjestu nalazi se brod s_2 koju bi luku p_2 trebao posjetiti prvog dana.

Određivanjem prvog i drugog skupa te kreiranjem pripadnih rang-listi preferencija, zadan je osnovni problem stabilnog sparivanja koji se rješava G-S algoritmom. Rješenje, to jest stabilno sparivanje S', čini n uređenih parova (s, p) gdje je s brod, a p luka koja je odredište broda s.

[1]Odnosi se na osnovni problem stabilnog sparivanja.

Napomena 4.2.2. *Određivanjem odredišta p broda s određeno je skraćivanje prvotnog rasporeda broda s, to jest skraćeni raspored broda s.*

Uvjet 4.1 iz opisanog problema moguće je promatrati kao nestabilnost svojstvenu ovom problemu stabilnog sparivanja. Slijedi definiranje *uvjetne nestabilnosti*[2] i stabilnog sparivanja u kontekstu ovog problema stabilnog sparivanja.

Definicija 4.2.3. *Neka je zadano sparivanje S'. Neka su $(s, p), (s', p') \in S'$ sa svojstvom da za skraćene rasporede brodova vrijedi: brod s je prije pristanka u svoje odredište p posjetio luku p' koja je odredište broda s' u koje je brod s' pristao ranije ili istog dana. Tada se kaže da je (s, p') **uvjetna nestabilnost** u S'.*

Napomena 4.2.4. *Za postojanje uvjetne nestabilnosti nužno je postojanje nestabilnosti (definicija 2.2.5[3]). Nastavno na definiciju 4.2.3: brod s pokušat će pristati u luku p' prije nego pristane u svoje odredište p ako luku p' preferira više od svog odredišta p. Nadalje, ako je brod s' već pristao u svoje odredište p', pristanak broda s u luku p' bit će moguć samo ako luka p' preferira brod s više od broda s'.*

Zbog prethodne napomene, stabilno sparivanje ovog problema stabilnog sparivanja definira se bez korištenja pojma uvjetne nestabilnosti. Budući da se radi o osnovnom problemu stabilnog sparivanja, vrijedi definicija 2.2.6, koja se ponovno navodi radi cjelovitosti analize problema.

Definicija 4.2.5. *Za sparivanje S' kaže se da je stabilno ako je savršeno i ako ne sadrži nestabilnosti.*

Sada tvrdnja da je rezultat G-S algoritma, za problem stabilnog sparivanja kreiran iz problema opisanog na početku poglavlja, stabilno sparivanje slijedi iz teorema 3.1.9.[4]

Teorem 4.2.6. *Neka je problem stabilnog sparivanja zadan skupom brodova S, skupom luka P, pri čemu je $|S| = |P|$ i rang-listama preferencija kreiranim iz zadanih rasporeda brodova. Rezultat svakog izvršavanja G-S algoritma je stabilno sparivanje S'.*

Dokaz. Slijedi iz teorema 3.1.9. □

[2]Slično su u potpoglavljima 3.4 i 3.5 definirane nestabilnosti potrebne za analizu opisanih problema stabilnog sparivanja.

[3]Za prilagodbu definicije problemu stabilnog sparivanja opisanom u ovom poglavlju potrebno je promijeniti korištenu terminologiju.

[4]Promatra se pseudokod G-S algoritma [1] na skupovima brodova i luka te se umjesto izraza ”zaprositi” koristi izraz ”tražiti dozvolu za pristanak”, a umjesto izraza ”prihvatiti prosidbu” koristi se izraz ”dodijeliti dozvolu za pristanak”.

Napomena 4.2.7. *Ne postoji dodatni uvjet koji rang-liste preferencija ispunjavaju, unatoč tome što rasporedi brodova ispunjavaju uvjet 4.1. Rang-liste preferencija, bilo brodova bilo luka, mogu se u potpunosti, djelomično ili nimalo podudarati - razlog je broj dana perioda m za koji vrijedi m > n, gdje je n broj brodova, odnosno luka.*

Primjer 4.2.8. Nastavno na primjer problema, odnosno problem stabilnog sparivanja, opisan u primjerima 4.1.2 i 4.2.1: u sparivanju $S'' = \{(s_1, p_1), (s_2, p_2), (s_3, p_3)\}$, uređeni par (s_3, p_1) je nestabilnost, što znači da S'' nije stabilno sparivanje. Odnosno, po napomeni 4.2.2, sparivanjem S'' nisu određeni skraćeni rasporedi brodova. Korištenjem G-S algoritma, dobiva se stabilno sparivanje $S' = \{(s_1, p_2), (s_2, p_3), (s_3, p_1)\}$, kojim su određeni skraćeni rasporedi brodova navedeni u primjeru 4.1.2.

Za potpuno rješenje prvog dijela opisanog problema, potrebno je dokazati da su rezultatom G-S algoritma svi određeni rasporedi brodova zbilja skraćeni - nije očito da stabilno sparivanje S' određuje rasporede brodova takve da nijedan brod sigurno za svoje odredište ne dobije luku koju bi, po prvotnom rasporedu, trebao posjetiti m-tog dana. Slijedi dokaz da je opisani problem uvijek svediv na problem stabilnog sparivanja rješiv G-S algoritmom.[5]

Teorem 4.2.9. *Svi rasporedi brodova određeni rezultatom svakog izvršavanja G-S algoritma su skraćeni.*

Dokaz. Promatrajući odnos traženja dozvola za pristanak u luku i rasporede brodova tijekom bilo kojeg izvršavanja G-S algoritma, zaključuje se sljedeće: prije nego se svaki brod nađe u situaciji da treba tražiti dozvolu za pristanak u luku koja se nalazi u m-tom danu njegova rasporeda, već se od svih n luka tražila dozvola za pristanak. Naime, u m-tom danu svakog broda može biti navedena najviše jedna luka, što znači da je svaki brod prije traženja dozvole za pristanak od luke iz m-tog dana tražio dozvolu za pristanak od najmanje $n-1$ luke. Kako se u m-tom danu, po uvjetu 4.1, sve luke moraju razlikovati, vrijedi da je od svih n luka zatražena dozvola za pristanak prije nego brodovi zatraže dozvolu za pristanak od luke iz m-tog dana svog rasporeda. Po lemi 3.1.4[6] vrijedi: dozvola za pristanak luke p ostaje dodijeljena od trenutka prvog traženja dozvole za pristanak. Dakle, izvršavanje G-S algoritma bit će završeno pa nijedan brod neće tražiti dozvolu za pristanak od luke iz m-tog dana. Time je dokazano da će duljina svih skraćenih rasporeda iznositi najviše $m-1$. □

Napomena 4.2.10. *U primjeru 4.1.2 duljina skraćenog rasporeda broda s_3 iznosi $m-1$. Kada bi prvotni raspored broda s_3 bio takav da brod s_3 prvog dana posjećuje luku p_3, a da trećeg dana plovi, tada bi duljina svih skraćenih rasporeda iznosila 1. Dakle, duljine skraćenih rasporeda brodova ovise o prvotnim rasporedima brodova, a sigurno nijedan skraćeni raspored neće biti duljine m.*

[5] Vrijedi fusnota 4.

[6] Uz promjenu korištene terminologije.

4.3 Implementacija algoritma

Drugi dio rješavanja problema odnosio se na konstruiranje algoritma za pronalazak skraćenih rasporeda. *Ship-Port algoritam*[7] sastoji se od tri dijela:

- kreiranja rang-listi preferencija brodova i luka iz rasporeda brodova
- određivanja odredišta brodova; pomoću G-S algoritma[8] (pseudokod [1])
- skraćivanja rasporeda brodova.

Slijedi pseudokod Ship-Port algoritma.

Algorithm 3: Ship-Port algoritam - pseudokod

---

inicijalizacija: zadani rasporedi brodova ispunjavaju uvjet 4.1 i nijedan brod $s \in S$ nema određen skraćeni raspored[a];
iz n zadanih rasporeda brodova kreiraj n rang-listi preferencija brodova zanemarivanjem oznaka plovidbe;
iz n zadanih rasporeda brodova kreiraj n rang-listi preferencija luka prema načelu: što je dan posjete luci kasniji, to luka više preferira brod;
primjeni G-S algoritam na kreiranim rang-listama preferencija brodova i luka;
kreiraj skraćene rasporede brodova korištenjem rezultata G-S algoritma;
return skraćene rasporede brodova;

---

[a]Uvjet da nijedan brod nema određen skraćeni raspored vrijedi po napomeni 4.1.1.

Prije opisa implementacije S-P algoritma, slijedi kratko objašnjenje Pythonove kolekcije rječnik (engl. *dictionary*). Rječnik je promjenjiva neuređena kolekcija. Svaki element rječnika sastoji se od ključa (engl. *key*) i vrijednosti (engl. *value*) koja je dodijeljena ključu. Ova kolekcija podobna je za opis problema stabilnog sparivanja jer svaki ključ u rječniku, za razliku od vrijednosti, mora biti jedinstven; prilikom zadavanja problema stabilnog sparivanja elementi skupova ne smiju se ponavljati, dok se rang-liste preferencija elemenata skupa smiju podudarati. Ključevi predstavljaju elemente skupova, a dodijeljene vrijednosti pripadne rang-liste preferencija. Nadalje, neuređenost rječnika ne predstavlja prepreku jer se elementi problema stabilnog sparivanja zadaju dvama skupovima elemenata - redoslijed elemenata nije relevantan za zadavanje problema stabilnog sparivanja.

---

[7]Ime algoritma inspirirano je skupovima brodova (engl. *ships*) i luka (engl. *ports*) te G-S algoritmom.

[8]Iako je G-S algoritam opisan terminologijom koja je svojstvena skupovima muškaraca i žena, jasno je da će rezultat za bilo koja dva skupa i pripadne rang-liste preferencija biti stabilno sparivanje.

Rješenje problema implementirano je u dvjema datotekama: *ship_port_algorithm.py* sadrži prvi i treći dio S-P algoritma te poziva drugi dio, izvršavanje G-S algoritma na kreiranim rang-listama preferencija, implementiran u datoteci *gale_shapley_algorithm.py*.

Rječnikom *original_ships_schedules* opisan je problem zadan prvotnim rasporedima brodova: ključevi su vrijednosti podatkovnog tipa *string* i predstavljaju brodove, a vrijednost svakog ključa je lista koja predstavlja prvotni raspored broda. Dani plovidbe označeni su dvjema povlakama radi čitljivosti koda.[9] Za opis pojedinih dijelova implementacije rješenja koristit će se problem opisan u primjeru 4.1.2. Iduća slika prikazuje zadavanje problema korištenjem kolekcije rječnik.

```
original_ships_schedules: Dict[str, List[str]] = {
    's1': ['p1', '--', 'p2', 'p3'],
    's2': ['p2', 'p3', '--', 'p1'],
    's3': ['--', 'p1', 'p3', 'p2']
}
```

Slika 4.1: Zadavanje prvotnih rasporeda brodova za problem iz primjera 4.1.2

Za izvršavanje S-P algoritma potrebno je pozvati metodu *sp_algorithm* koja prima parametar *original_schedules*; rječnik *original_ships_schedules* je argument prilikom pozvanja metode. Za kreiranje rang-listi preferencija luka potrebno je koristiti ulazni rječnik i rječnik koji sadrži skraćene rasporede brodova. Zbog toga nakon pozivanja metode prvo slijedi kopiranje rječnika *original_schedules* u rječnik *ships_preferences*. Vrijednosti rječnika nije moguće dodijeliti drugom rječniku klasičnim pridruživanjem; rječnik *original_schedules* potrebno je kopirati korištenjem modula *copy* i metode *deepcopy*. Vrijednosti kopiranog rječnika *ships_preferences* pretvaraju se u rang-liste preferencija pripadnih ključeva, odnosno brodova, uklanjanjem oznaka plovidbe. Budući da su vrijednosti ključeva liste, uređene kolekcije, nakon uklanjanja pojedinog elementa redoslijed preostalih ostaje nepromijenjen čime se poštuje koncept kreiranja rang-listi preferencija brodova naveden u prethodnom potpoglavlju: brod preferira luke u redoslijedu kojim su navedene u njegovom prvotnom rasporedu.

Za kreiranje rang-listi preferencija luka potrebno je dohvatiti njihove nazive; dohvaća se vrijednost ključa s nultog indeksa iz rječnika *ships_preferences*. Nakon inicijalizacije rječnika *ports_preferences*, svakom se ključu, koji predstavlja luku, za vrijednost dodjeljuje prazna lista. Zatim se korištenjem triju petlji for kreiraju rang-liste preferencija luka prema načelu navedenom u pseudokodu S-P algoritma [3].

[9] U prethodnom potpoglavlju za oznaku dana plovidbe korištene su nule.

Prije izvršavanja triju ugniježđenih petlji for, potrebno je dohvatiti prvotni raspored jednog od brodova jer je gornja ograda druge petlje for broj m, to jest duljina prvotnih rasporeda brodova. Rang-liste preferencija luka kreiraju se jedna po jedna; rang-lista preferencija luke *port* dopunjuje se novim brodom *ship* ako je ispunjen zadani uvjet, odnosno ako je po prvotnom rasporedu brod *ship* na dan i trebao posjetiti luku *port*. Pripadni kôd prikazan je na slici 4.2.

```
original_schedule = original_schedules[one_ship]
m = len(original_schedule)
shortened_schedule = ships_preferences[one_ship]
n = len(shortened_schedule)
for port in ports_preferences.keys():
    for i in reversed(range(m)):
        if len(ports_preferences[port]) == n:
            break
        for ship in ships_preferences.keys():
            if original_schedules[ship][i] == port:
                ports_preferences[port].append(ship)
                break
```

Slika 4.2: Kreiranje rang-listi preferencija luka

Sada rječnik *ships_preferences* opisuje matricu preferencija prvog skupa, to jest skupa brodova, a rječnik *ports_preferences* opisuje matricu preferencija drugog skupa, to jest skupa luka. Time je obavljen prvi dio S-P algoritma: kreiranje rang-listi preferencija brodova i luka.

Slijedi izvršavanje drugog dijela S-P algoritma: određivanje odredišta brodova. Terminologija korištena prilikom implementacije G-S algoritma odgovara trećem poglavlju. Metoda *gs_algorithm* prima dva parametra, rječnike *preferences_men* i *preferences_woman*; pozvana s argumentima *ships_preferences* i *ports_preferences* vraća stabilno sparivanje *matching* čijim su elementima određeni brodovi i njihova odredišta.

Inicijaliziraju se kolekcije: prazni rječnik *matching*, lista *free_men* sa svim muškarcima i rječnik *proposals* u kojem svaki par ključ-vrijednost predstavlja muškarca i indeks žene koju će muškarac potencijalno sljedeću zaprositi. Sve vrijednosti rječnika *proposals* inicijalizirane su na nulu - predstavljaju prve žene s rang-listi preferencija muškaraca. Pseudokod G-S algoritma [1] u inicijalizaciji navodi da su sve žene slobodne; taj zahtjev nije potrebno inicijalizirati u kodu. Status žene doznaje se iz rječnika *matching* u kojem se žena nalazi samo ako je prihvatila prosidbu. Po lemi 3.1.4, jednom kada se nađe u rječniku *matching*, žena ostaje u njemu do kraja izvršavanja G-S algoritma.

Brojač *iterations_counter* inicijaliziran je na jedan; nije nužan, ali je koristan prilikom testiranja i analiziranja izvršavanja G-S algoritma. Petlja while izvršava se dok postoji barem jedan slobodan muškarac, odnosno dok lista *free_men* ne postane prazna, i dok je ispunjen drugi uvjet. Drugi dio uvjeta iz pseudokoda [1], da muškarac osim što je slobodan nije zaprosio svaku ženu, nije potrebno implementirati - zbog leme 3.1.5. Ipak, radi potpunosti, *bool* vrijednošću varijable *not_propose_to_all_woman* provjerava se jesu li svi muškarci sadržani u listi *free_men* zaprosili sve žene; pri čemu se koristi metoda *all*. Kôd petlje while prikazan je na slici 4.3, pojašnjenje slijedi u nastavku.

```
while free_men and not_propose_to_all_women:
    potential_fiance = random.choice(free_men)
    woman_index = proposals[potential_fiance]
    proposals[potential_fiance] += 1
    potential_fiance_preferences = preferences_men[potential_fiance]
    woman_to_propose = potential_fiance_preferences[woman_index]

    if woman_to_propose not in matching.values():
        matching[potential_fiance] = woman_to_propose
        free_men.remove(potential_fiance)
    else:
        current_fiance = list(matching.keys())[list(
            matching.values()).index(woman_to_propose)]
        woman_to_propose_pref = preferences_women[woman_to_propose]
        current_fiance_index = woman_to_propose_pref.index(current_fiance)
        potential_fiance_index = woman_to_propose_pref.index(potential_fiance)

        if current_fiance_index < potential_fiance_index:
            pass
        else:
            matching.pop(current_fiance)
            matching[potential_fiance] = woman_to_propose
            free_men.remove(potential_fiance)
            free_men.append(current_fiance)

    iterations_counter += 1
    not_propose_to_all_women = all(proposals[men] <
                                   out_of_range_index for men in free_men)
```

Slika 4.3: Petlja while G-S algoritma

Nakon dohvaćanja proizvoljno odabranog muškarca *potential_fiance* korištenjem modula *random*, indeks žene koju će muškarac zaprositi sprema se u varijablu *woman_index*, a indeks potencijalne buduće prosidbe dohvaćenog muškarca *proposals[potential_fiance]* inkrementira se za jedan.

Slijedi dohvaćanje rang-liste preferencija proizvoljno odabranog muškarca u varijabli *potential_fiance_preferences* i žene koju on prosi u varijabli *woman_to_propose*. U nastavku se određuje ishod prosidbe. Ako se žena *woman_to_propose* ne nalazi u vrijednostima rječnika *matching*, znači da je ovo prva prosidba koju žena prima i prihvaća ju, sukladno drugoj tvrdnji 3.1 navedenoj nakon pseudokoda G-S algoritma. Tada se ključ i vrijednost, odnosno proizvoljno odabran muškarac i žena koju je zaprosio, dodaju u rječnik *matching* te se proizvoljno odabran muškarac uklanja iz liste *free_men*.

Inače, ako se žena *woman_to_propose* nalazi u vrijednostima rječnika *matching*, dohvaća se sljedeće: njen trenutni zaručnik *current_fiance*, njena rang-lista preferencija *woman_to_propose_pref* s koje se dohvaćaju indeks trenutnog zaručnika *current_fiance_index* i indeks potencijalnog zaručnika *potential_fiance_index*. Rang-liste preferencija opisane su listama tako da se muškarci, odnosno žene, s manjim indeksom više preferiraju. Dakle, ako je vrijednost varijable *current_fiance_index* strogo manja od vrijednosti varijable *potential_fiance_index*, žena *woman_to_propose* preferira trenutnog zaručnika više od prosca, odnosno potencijalnog zaručnika, te se situacija ne mijenja - rječnik *matching* ostaje isti kao i u prethodnoj iteraciji petlje while.

Inače, ako žena *woman_to_propose* preferira prosca više od svog zaručnika, prosidba se prihvaća: iz rječnika *matching* uklanja se element jedinstveno određen ključem *current_fiance* te se dodaje element s ključem *potential_fiance* i vrijednošću *woman_to_propose*; iz liste *free_men* uklanja se muškarac *potential_fiance* i dodaje se muškarac *current_fiance*.[10] Na kraju, brojač iteracija inkrementira se za jedan i provjerava se vrijedi li za sve slobodne muškarce uvjet da nisu zaprosili sve žene.

Iako u ovom slučaju standardna terminologija korištena za opisivanje G-S algoritma ne odgovara kontekstu promatranog problema, jasno je da će za argumente *ships_preferences* i *ports_preferences* rezultat, odnosno rječnik *matching*, biti stabilno sparivanje.

Treći dio S-P algoritma odnosi se na kreiranje skraćenih rasporeda brodova ovisno o rječniku *matching* čije se vrijednosti nakon pozivanja metode *gs_algorithm* dodjeljuju istoimenom rječniku. Nakon inicijalizacije praznog rječnika *shortened_schedules*, za svaki ključ *ship* u rječniku *matching* dohvaća se indeks vrijednosti, to jest luke *port*, *port_index* iz prvotnog rasporeda broda *ship*. U rječnik *shortened_schedule* dodaje se ključ *ship* s pridruženom vrijednošću: skraćenom vrijednošću ključa *ship* iz rječnika *original_schedules* zaključno s odredišnom lukom *port*. Pripadni kôd prikazan je na slici 4.4.

[10] Vrijedi lema 3.1.3.

Metoda *sp_algorithm* vraća rječnik *shortened_shedules* čime je izvršavanje S-P algoritma završeno.

```
    for ship in matching:
        port = matching[ship]
        port_index = original_schedules[ship].index(port)
        shortened_schedule = original_schedules[ship][:port_index + 1]
        shortened_schedules[ship] = shortened_schedule
```

Slika 4.4: Kreiranje skraćenih rasporeda brodova

Implementacija[11] je zaključena ispisom prvotnih i skraćenih rasporeda brodova; metodom *comparison*. Radi jednostavnije usporedbe prvotnih i skraćenih rasporeda, rječnici su sortirani prije ispisa. Na slici 4.5 prikazan je ispis prvotnih i skraćenih rasporeda brodova za primjer 4.1.2; odredišne luke brodova su različite te se nikoja dva broda ne nalaze u istoj luci istog dana.

```
- original schedules:
    ship s1: ['p1', '--', 'p2', 'p3']
    ship s2: ['p2', 'p3', '--', 'p1']
    ship s3: ['--', 'p1', 'p3', 'p2']

- shortened schedules:
    ship s1: ['p1', '--', 'p2']
    ship s2: ['p2', 'p3']
    ship s3: ['--', 'p1']
```

Slika 4.5: Zadani problem iz primjera 4.1.2 i pripadno rješenje

[11]Link na potpunu implementaciju: `https://github.com/KristinaUdovicic/master_thesis`. Ship-Port algoritam implementiran je u programskom jeziku Python, verzija 3.8 te je korišteno integrirano razvojno okruženje PyCharm Community. Slike iz potpoglavlja prikazuju sadržaj datoteka iz direktorija *algorithm*: *ship_port_algorithm.py*, *gale_shapley_algorithm.py* i *sp_example_1.txt*.

Gale-Shapleyjev algoritam implementiran je zasebno radi važnosti te omogućavanja samostalnog izvršavanja i analiziranja. Problemi su zadani u *.py* datotekama i linije poziva metoda su zakomentirane, osim za prvi problem u datoteci *ship_port_algorithm.py*. Priložene su *.txt* datoteke s pripadnim ispisima. Datoteke direktorija *algorithm_detailed* sadržavaju dodatne komentare i linije kôda za ispis koje opisuju izvršavanje algoritama - s ciljem da uz dobivanje rješenja problema bude jasan i postupak dobivanja istog.

Bibliografija

[1] D. Austin, *The Stable Marriage Problem and School Choice*, American Mathematical Society, Feature Column (March 2015.), http://www.ams.org/samplings/feature-column/fc-2015-03.

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

[3] D. Gusfield i R. W. Irving, *The stable marriage problem: structure and algorithms*, MIT press, 1989.

[4] R. W. Irving, P. Leather i D. Gusfield, *An efficient algorithm for the "optimal" stable marriage*, Journal of the ACM (JACM) **34** (1987), br. 3, 532–543.

[5] J. Kleinberg i É. Tardos, *Introduction to Algorithms*, Cornell University Course Notes, COMS **482** (2003).

[6] D. G. McVitie i L. B. Wilson, *The Stable Marriage Problem*, Commun. ACM **14** (1971), br. 7, 486–490, ISSN 0001-0782, https://doi.org/10.1145/362619.362631.

[7] Royal Swedish Academy of Sciences, *The Prize in Economic sciences 2012., Stable matching: Theory, evidence, and practical design*, (2012), https://www.nobelprize.org/uploads/2018/06/popular-economicsciences2012.pdf.

[8] K. Wayne, *Stable matching*, Copyright © 2005 Pearson-Addison Wesley (2018), https://www.cs.princeton.edu/~wayne/kleinberg-tardos/pdf/01StableMatching.pdf.

Sažetak

U diplomskom radu detaljno je analiziran problem stabilnog sparivanja koji su 1962. riješili David Gale i Lloyd S. Shapley: problem stabilnih brakova. Rješenje problema, Gale-Shapleyjev algoritam, poznat i kao algoritam *odgođenog prihvaćanja* (engl. *deferred acceptance*), pokazao se uspješnim i za rješavanje drugih generalizacija problema stabilnog sparivanja, uz potrebne modifikacije.

Diplomski rad zamišljen je kao temelj za razumijevanje problema stabilnog sparivanja i kao motivacija za njegovo daljnje proučavanje.

Summary

The master thesis analyzes in detail the stable matching problem, which was solved by David Gale and Lloyd S. Shapley in 1962: the stable marriage problem. The solution to the problem, the Gale-Shapley algorithm, also known as the *deferred acceptance* algorithm, has proven to be successful for solving other generalizations of the problem, with the necessary modifications.

The master thesis was conceived as a basis for understanding the stable matching problem and as a motivation for its further study.

Životopis

Rođena sam 30. 12. 1994., u Splitu. Osnovnu školu pohađala sam u Studencima. Srednju školu, IV. gimnaziju "Marko Marulić", pohađala sam u Splitu. Na Prirodoslovno-matematičkom fakultetu u Splitu, 2017. godine završavam sveučilišni preddiplomski studij Informatika, nakon čega upisujem sveučilišni diplomski studij Računarstvo i matematika, na Prirodoslovno-matematičkom fakultetu u Zagrebu.