

Sveučilište u Rijeci – Odjel za informatiku

Diplomski studij informatike - Informacijski i komunikacijski sustavi

Toni Čavar

# Aplikacija za raspoređivanje radnika u ugostiteljstvu

Diplomski rad

Mentor: doc. dr. sc. Marina Ivašić – Kos

Komentor: dr. sc. Martina Holenko Dlab

Rijeka, 2018

Rijeka, 12.3.2018.

## Zadatak za diplomski rad

Pristupnik: Toni Čavar

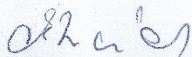
Naziv diplomskog rada: Aplikacija za raspoređivanje radnika u ugostiteljstvu

Naziv diplomskog rada na eng. jeziku: Application for staff scheduling in catering

Sadržaj zadatka: Jedan od problema u području operacijskih istraživanja je raspoređivanje radnika na poslove odnosno planiranje rada u smjenama. Zadatak diplomskog rada je izraditi matematički model problema raspoređivanja radnika u ugostiteljstvu. Na temelju matematičkog modela je potrebno izraditi aplikaciju koja će omogućiti izradu mjesečnih rasporeda za rad u smjenama, izračun troškova za plaće radnicima i modifikaciju planiranih radnih dana ili smjena za pojedinog radnika u skladu s potrebama u praksi.

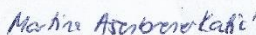
Mentor:

Doc. dr. sc. Marina Ivašić-Kos

  
\_\_\_\_\_

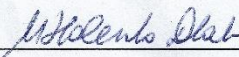
Voditelj za diplomske radove:

Dr. sc. Martina Ašenbrener Katić

  
\_\_\_\_\_

Komentor:

Dr. sc. Martina Holenko Dlab

  
\_\_\_\_\_

Zadatak preuzet: (05.04.2018.)

  
\_\_\_\_\_

(potpis pristupnika)

## Sažetak

Cilj diplomskog rada je prezentirati izrađenu aplikaciju za rješavanje problema raspoređivanja koji se javlja prilikom izrade mjesečnih rasporeda u ugostiteljskim objektima.

Teorijski dio rada se sastoji od opisa problema raspoređivanja, pojmova linearnog programiranja i algoritma grananja i ograđivanja. Analizirana su postojeća rješenja te je izrađen matematički model problema raspoređivanja smjena iz stvarnoga života.

Praktični dio rada se sastoji od izrade aplikacije u kojoj vlasnik može unijeti sve potrebne informacije o svom ugostiteljskom objektu i njegovim radnicima. Aplikacija, na bazi unesenih podataka, predloži mjesečni raspored s izračunatim troškom plaća. Vlasnik može detaljnije modificirati predloženi raspored (dodati i brisati smjene, izmijeniti trajanje smjena, zamijeniti smjene između radnika, itd.) te ga pospremiti u bazu podataka kako bi bio vidljiv svim radnicima tog ugostiteljskog objekta.

Za izradu aplikacije korišteni su Java i JavaScript programski jezici, MySql baza podataka te modificirana verzija programa Eclipse kao razvojno okruženje.

Ključne riječi: linearno programiranje, grananje i ograđivanje, problem raspoređivanja, izrada rasporeda smjena, Java, Spring okvir

# Sadržaj

1. Uvod.....	1
2. Problem raspoređivanja .....	2
2.1. Metode rješavanja problema raspoređivanja .....	3
2.2. Linearno programiranje .....	5
2.2.1. Standardni problem linearnog programiranja .....	5
2.3. Grananje i ograđivanje.....	6
2.3.1. Grananje .....	7
2.3.2. Ograđivanje .....	8
2.3.3. Procjenjivanje.....	9
2.4. Primjer cjelobrojnog programiranja .....	9
2.5. Pregled postojećih aplikacija .....	12
2.5.1. ABC Roster .....	12
2.5.2. 247 Shift .....	13
3. Izrada modela odabranog problema.....	14
3.1. Prikupljeni podaci i ograničenja .....	14
3.2. Postavljanje matematičkog modela .....	17
3.3. Korištene tehnologije.....	21
3.4. Izrada modela za aplikaciju .....	22
4. Izrada aplikacije za raspoređivanje radnika u ugostiteljstvu .....	22
4.1. DEV model baze podataka .....	23
4.2. Relacijski model baze podataka .....	23
4.3. Odabir programske podrške.....	24
4.4. Aplikacija.....	24
4.5. Pregled i usporedba rješenja .....	30
5. Zaključak.....	31
6. Literatura.....	32

7.	Popis slika .....	34
8.	Popis tablica .....	34

# 1. Uvod

Ovaj diplomski rad je posvećen rješavanju problema raspoređivanja radnika u ugostiteljstvu koristeći metodu linearnog programiranja i algoritma grananja i ograđivanja. Postoje mnoge vrste problema raspoređivanja te je za neke moguće dobiti optimalno rješenje koristeći olovku, papir i par skica problema. Kako svijet i tehnologija napreduju tako i problemi raspoređivanja postaju sve kompleksniji te njihovo rješenje zahtjeva složene matematičke postupke i primjenu računala. Problem raspoređivanja (eng. *assignment problem*) je vrlo značajan problem u području operacijskih istraživanja (eng. *operational research*). Iako takav problem spada u područje kombinatorne optimizacije može se smatrati kao specijalan slučaj općih zadataka linearnog programiranja [1]. Linearno programiranje (eng. *linear programming*, LP) je jedna od najstarijih metoda za rješavanje problema raspoređivanja te pripada području operacijskih istraživanja. Tehnika se koristi za utvrđivanje optimalne kombinacije ograničenih resursa u postizanju željenog cilja [2]. Bavi se problemima u kojima je cilj maksimizirati ili minimizirati linearnu funkciju uz dane uvjete. Fokus u ovome radu je na problemu raspoređivanja smjena (eng. *Work-Shift Scheduling Problem*). Problemi te vrste se najčešće rješavaju pomoću cjelobrojnog linearnog programiranja. Cjelobrojno linearno programiranje je specifična vrsta linearnog programiranja u kojemu neke ili sve varijable moraju imati cjelobrojne vrijednosti. Osim razlike u vrijednostima varijabli odlučivanja problemi cjelobrojnog programiranja, za razliku od klasičnih problema linearnog programiranja, se rješavaju posebnim algoritmima. Za rješavanje problema u ovome radu će se koristiti algoritam grananja i ograđivanja (eng. *Branch & Bound*).

Motivacija za ovakav rad proizašla je iz mnogih susretanja s problemom raspoređivanja smjena tokom rada kao konobar u jednom riječkom kafiću. Raspored smjena je, za svaki mjesec, bilo potrebno ručno izraditi u Excel programu. To je zahtijevalo jako puno vremena i strpljenja. Stalni radnici su često radili prekovremene sate ili su im bile dodijeljene smjene koje su puno jeftinije, ali s istom efikasnošću, mogli odraditi studenti. Nakon što je vlasniku predstavljen problem i mogućnost njegovog rješavanja s velikim entuzijazmom je pristao na davanje svih potrebnih informacija o plaćama i trenutno korištenog rasporeda smjena.

Poglavlje 2 sadrži teoriju problema raspoređivanja, opis nekih vrsta problema i mogućnosti rješavanja, standardni primjer linearnog programiranja, detaljan opis algoritma grananja i ograđivanja s popratnim primjerom te pregled aplikacija za izradu rasporeda smjena koje su najviše utjecale na izradu vlastite aplikacije. U 3. poglavlju je opisana izrada matematičkog

modela za odabrani problem raspoređivanja radnika u ugostiteljstvu. 4. poglavlje sastoji se od detaljnog opisa baze podataka i svih korištenih softverskih elemenata prilikom izrade aplikacije a 5. poglavlje donosi zaključke.

## 2. Problem raspoređivanja

Raspoređivanje je u širem smislu postupak izrade bilo kakvog rasporeda. Problem raspoređivanja se često pojavljuje u praksi i može se pojaviti u raznim oblicima. Proučavanje problema raspoređivanja je disciplina koja se razvija više od pola stoljeća. Kroz vrijeme, usporedo s razvojem tehnologije i sve kompleksnijih radnih okruženja (rad od kuće, određivanje vlastitih radnih sati, promjene radničkih prava, itd.), su problemi raspoređivanja postali sve složeniji te je bilo sve teže naći učinkovite algoritme za njihovo rješavanje. Kako za većinu problema nije više bilo moguće dobiti optimalno rješenje razvili su se heuristički algoritmi iz kojih su najčešće proizlazila dovoljno dobra rješenja.

U suštini kada se govori o problemu raspoređivanja onda se misli na alokaciju resursa na način da se optimizira funkcija cilja. Problemi raspoređivanja mogu se svrstati u kombinacijske probleme. Rješenje tih problema predstavlja raspored koji je izvediv, ali koji mora nastojati ispuniti niz dodatnih ograničenja. Bolje ili lošije ispunjavanje tih ograničenja definira kvalitetu samoga rješenja [3].

Postoje mnogi problemi koji se mogu klasificirati kao problemi raspoređivanja. U nastavku su navedeni samo neki od njih.

### **Raspoređivanje voznog reda javnog transporta**

Pod javni transport misli se na vozni red vlakova, aviona, autobusa, itd. Takvi problemi se rješavaju na isti način. Potrebno je pokriti sve dionice sa što manjim brojem transportnih sredstava (vlakovi, autobusi, avioni). Za svaku dionicu je potrebno osigurati dovoljan kapacitet za prijevoz putnika. Također je potrebno rasporediti rezervna transportna sredstva kako bi se, u slučaju kvarova, smanjilo vrijeme zastoja prometa.

### **Školski raspored sati**

U ovome problemu je potrebno grupi studenata, za nastavni dan, dodijeliti profesora i učionicu. Složenost ovoga problema ovisi od ustanove do ustanove. Ako ustanova ima jako velik broj studenata, profesora i učionica onda problem postaje jako kompleksan. Npr. na Fakultetu elektrotehnike i računarstva ima više od 300 zaposlenih profesora i asistenata, studenti su

raspoređeni u 60-tak nastavnih grupa i ima otprilike 60 učionica. Nastava se svakim radnim danom održava od 8:00 do 20:00, time dobivamo:

$$\begin{aligned} & 300 \text{ nastavnika} * 60 \text{ grupa studenata} * 60 \text{ učionica} * 5 \text{ dana u tjednu} * 12 \text{ sati} \\ & = 64.800.000 \text{ kombinacija} \end{aligned}$$

Dobivena vrijednost se odnosi na broj kombinacija za održavanje nastave iz samo jednog predmeta. Kako bi se smanjio broj kombinacija uvode se ograničenja vezana uz održavanje nastave. Nastava nekih predmeta se ne može održati u svim učionicama (npr. računalne vježbe mogu se održati samo u učionicama s računalima); prvo se mora održati teorijsko predavanje, a tek onda laboratorijske vježbe iz nekog predmeta; jedan nastavnik sudjeluje u nastavi u najviše nekoliko predmeta i sl.

### **Raspored poslova u tvornici**

Raspored poslova u tvornici najčešći je problem optimizacije. Problem se odnosi na raspoređivanje poslova na određenom stroju u određeno vrijeme kako bi se optimalno iskoristile sve sirovine i dobio konačan proizvod. Problemi ove vrste poprilično su jednostavni, ali se veoma lako zakompliciraju uvođenjem skladištenja sirovina, nabavka sirovina i isporuka gotovih proizvoda, suočavanje s kvarovima strojeva i sl.

### **Raspored radne snage u smjene**

Raspoređivanje radne snage može se primijeniti ili na raspoređivanje smjena u objektima uslužnih djelatnosti ili raspoređivanje posada u transportnom okruženju. Raspoređivanje smjena je na neki način slično raspoređivanju poslova po strojevima. Radnici se mogu smatrati ekvivalentnima s mašinama koji moraju obaviti određen zadatak (smjenu) u jediničnom vremenu [4].

## **2.1. Metode rješavanja problema raspoređivanja**

Postoje različite metode koje se koriste za rješavanje problema raspoređivanja. Ne postoji nijedna generalna metoda za sve probleme već je svaka od njih prilagođena pojedinim skupinama problema. U ovome poglavlju su predstavljene neke od metoda rješavanja problema raspoređivanja.

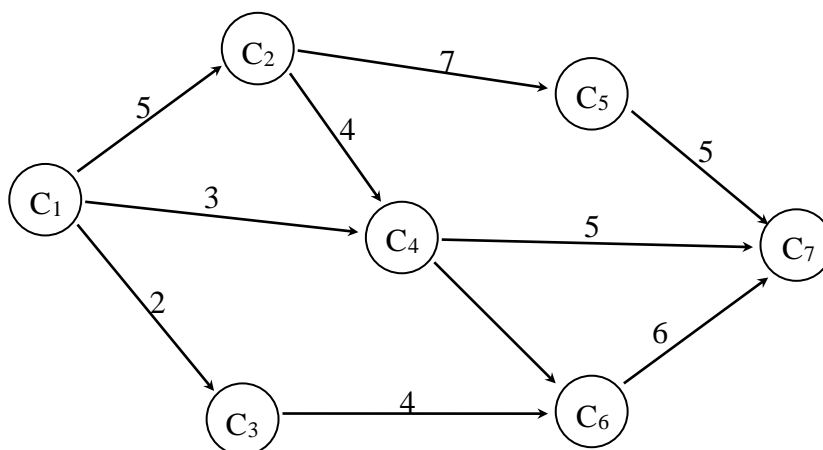


## Nasumične metode pretraživanja

Kod nasumičnih metoda pretražuje se prostor mogućih rješenja te se nastoji pronaći optimalno rješenje koje najbolje zadovoljava postavljene uvjete. Kvaliteta rješenja kod ovakvih metoda ovisi isključivo o iskustvu osobe. Metode se prvenstveno koriste za rješavanje problema manjeg opsega, dok kod većih rješavanje postaje veoma kompleksno i vremenski zahtjevno. Performanse rješenja ovise o omjeru između broja dobrih rješenja i ukupnog broja rješenja u prostoru koji se pretražuje, o početnoj točki za pretragu te o redoslijedu pretraživanja prostora [5].

## Tehnike mrežnog toka

Kod tehnike mrežnog toka definiran je usmjereni težinski graf. Opći problem mrežnoga toka jest pronalaženje minimalnog ili maksimalnog toka između dva zadana čvora [5]. Na Slici 1 prikazan je primjer mrežnog toka.



*Slika 1 Primjer mrežnog toka*

Primjer primjene ove tehnike je dan u znanstvenom radu matematičara Ostermann i De Werra [6]. U njihovom radu su uspješno riješili problem rasporeda sati pomoću tehnike mrežnog toka. Za svaki dan u mjesecu konstruiran je jedan mrežni tok gdje su čvorovi predavanja a veze njihovo trajanje. Sve mreže se na kraju spajaju u jednu veliku čije rješenje predstavlja raspored sati za mjesec dana.

## Cjelobrojno programiranje

Linearno programiranje je najčešće korištena metoda za rješavanje problema raspoređivanja. To je tehnika utvrđivanja optimalne kombinacije ograničenih resursa u postizanju željenog cilja. Kod problema je potrebno zadati linearnu funkciju cilja koju je potrebno minimizirati ili

maksimizirati i skup linearnih ograničenja koje je potrebno zadovoljiti. Cjelobrojno programiranje je posebna vrsta linearnog programiranja gdje neke ili sve varijable moraju imati cjelobrojne vrijednosti. Rješavanje problema cjelobrojnog programiranja izvodi se koristeći metodu grananja i ograđivanja (*eng. Branch & Bound*) [5]. Pomoću te metode se kroz iteracije i dodavanje novih ograničenja postepeno prisiljava varijable da poprime cjelobrojne vrijednosti.

## 2.2. Linearno programiranje

Problemi raspoređivanja u ovome radu će se riješiti najčešćom metodom rješavanja, to jest metodom linearnog programiranja. Linearno programiranje promatra probleme u kojima se linearna funkcija cilja mora optimizirati uz ograničenja dana u obliku linearnih jednadžbi i/ili nejednadžbi i uz nenegativne varijable odlučivanja. Tehniku je prvi razvio Leonid Kantorovich 1939. godine kako bi za vrijeme II. svjetskog rata olakšao planiranje troškova i zarade te time uspješno smanjio troškove vojske i povećao novčane gubitke neprijatelja. Nakon rata mnoge su industrije pronašle korist linearnog programiranja prilikom planiranja troškova i zarade [6].

### 2.2.1. Standardni problem linearnog programiranja

Standardni problem ima sljedeći oblik [5]:

$$\max z = \sum_{j=1}^n c_j * x_j$$

Uz ograničenja:

$$\sum_{j=1}^n a_{ij} * x_j \leq b_i; i = 1, \dots, m$$

$$x_j \geq 0; j = 1, \dots, n$$

Svi ostali oblici problema mogu se svesti na ovaj oblik, npr. :

1. Kod minimizacije maksimizira se funkcija cilja pomnožena s -1
2. Ograničenja vrste  $\geq$  se množe s -1 kako bi se dobila ograničenja vrste  $\leq$
3. Umjesto ograničenja vrste  $=$  uvode se dva nova ograničenja vrste  $\leq$  i  $\geq$  s istim vrijednostima

Najbitnije definicije problema linearnog programiranja:

1. Varijable  $x_1, \dots, x_n$  nazivaju se strukturnim varijablama
2. Rješenjem se naziva bilo koja uređena  $n$ -torka  $(x_1, \dots, x_n) \in \mathbb{R}^n$
3. Rješenje je moguće ili dopustivo ako zadovoljava sva postavljena ograničenja. Ako barem jedno ograničenje nije zadovoljeno, rješenje se naziva nemogućim
4. Optimalno rješenje je ono moguće rješenje za koje je vrijednost funkcije cilja maksimalna
5. Vršno moguće rješenje je rješenje koje ne leži na spojnici bilo koja druga dva moguća rješenja

### 2.3. Grananje i ograđivanje

Pošto problemi cjelobrojnog i binarnog programiranja imaju konačan broj mogućih rješenja prva ideja za rješavanje problema te vrste je ispitivanje svih mogućih rješenja. Iako ideja zvuči primamljivo i poprilično jednostavno, ona nije moguća zbog izrazito velikog broja mogućih rješenja i za maleni broj varijabli [5]. Kako bi se smanjio broj mogućih rješenja i usmjerilo pretraživanje prema optimalnome koristi se metoda grananja i ograđivanja koja se bazira na principu podijeli pa vladaj. Početno zadan problem se dijeli na manje dijelove ograničavanjem vrijednosti pojedinih varijabli. Podjelom na manje probleme se olakšava rješavanje te, po potrebi, odbacivanje nezadovoljavajućih vrijednosti funkcije cilja.

Postupak grananja i ograđivanja sastoji se od tri koraka:

1. Grananje (*eng. branching*)
2. Ograđivanje (*eng. bounding*)
3. Procjenjivanje (*eng. fathoming*)

Sva tri koraka su prikazana sljedećim primjerom u kojemu su sve varijable binarne:

$$\max z = 9x_1 + 5x_2 + 6x_3 + 4x_4$$

$$6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10$$

$$x_3 + x_4 \leq 1$$

$$-x_1 + x_3 \leq 0$$

$$-x_2 + x_4 \leq 0$$

### 2.3.1. Grananje

Kod binarnih varijabli najlakši način razdvajanja prostora rješenja je postavljanje vrijednosti varijable na 0 za prvi podskup i na 1 za drugi podskup. U problemima cjelobrojnog programiranja u kojemu varijable mogu poprimiti sve cjelobrojne vrijednosti grananje se obavlja tako da se, nakon odabira varijable, definiraju dvije cjelobrojne vrijednosti. U nekim problemima je moguće primijetiti koju je varijablu najbolje odabrati, ali u većini slučajeva je najbolje krenuti od varijable s najmanjim indeksom. Nakon postavljanja varijable  $x_1$  na 0 i 1 dobivaju se sljedeća dva potproblema:

Potproblem 1 ( $x_1 = 0$ ):

$$\max z = 5x_2 + 6x_3 + 4x_4$$

$$3x_2 + 5x_3 + 2x_4 \leq 10$$

$$x_3 + x_4 \leq 1$$

$$x_3 \leq 0$$

$$-x_2 + x_4 \leq 0$$

Potproblem 2 ( $x_1 = 1$ ):

$$\max z = 9 + 5x_2 + 6x_3 + 4x_4$$

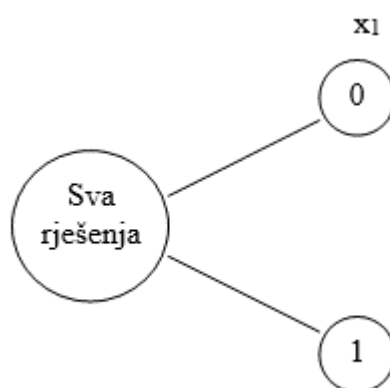
$$3x_2 + 5x_3 + 2x_4 \leq 4$$

$$x_3 + x_4 \leq 1$$

$$x_3 \leq 1$$

$$-x_2 + x_4 \leq 0$$

Na Slici 2 je prikazan način podjele prostora rješenja. Prikazano stablo naziva se stablo enumeracija (*eng. enumeration tree*).



Slika 2 Stablo enumeracija

### 2.3.2. Ograđivanje

Prije nego se, za svaki od prethodno definiranih potproblema, pronađe maksimalna vrijednost funkcije cilja potrebno je pronaći vrijednost funkcije cilja početnog problema. Željena vrijednost se najčešće dobiva rješavanjem relaksiranog problema. Relaksirani problem se dobiva micanjem jednog ili više ograničenja koja navise otežavaju rješavanje problema. Rješavanjem relaksiranog početnog problema dobije se vrijednost:

$$(x_1, x_2, x_3, x_4) = \left(\frac{5}{6}, 1, 0, 1\right); \quad z = 16\frac{1}{2}$$

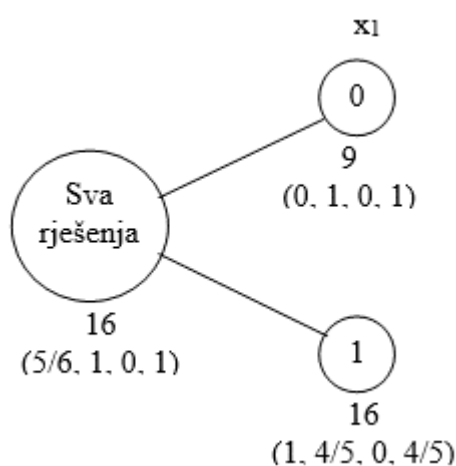
Dobivena vrijednost je maksimalna vrijednost koja se može dobiti ako ne postoji uvjet da su varijable cjelobrojne. Budući da su u ovome problemu sve varijable binarne, a samim tim i cjelobrojne, onda i funkcija cilja će zasigurno biti cjelobrojna vrijednost. Samim time kao ograničenje funkcije cilja za cijeli problem može se postaviti  $z \leq 16$ .

Rješavanjem relaksiranog problema za potprobleme dobiju se vrijednosti:

Potproblem 1:  $(x_1, x_2, x_3, x_4) = (0, 1, 0, 1); \quad z = 9$

Potproblem 2:  $(x_1, x_2, x_3, x_4) = \left(1, \frac{4}{5}, 0, \frac{4}{5}\right); \quad z = 16\frac{1}{5}$

Slika 3 prikazuje stablo rješenja nakon prvog koraka ograđivanja.



Slika 3 Stablo rješenja nakon prvog koraka ograđivanja

### 2.3.3. Procjenjivanje

U koraku procjenjivanja se odlučuje ako je moguće odbaciti neki od potproblema i time smanjiti prostor rješenja. Postoje tri načina na koji se može odbaciti potproblem iz daljnjeg razmatranja:

1. Rješenje potproblema 1 zadovoljava sva ograničenja početnog problema. Daljnjim grananjem ne može se dobiti bolja vrijednost funkcije cilja, pa daljnje grananje nema smisla. Pošto je to rješenje trenutno najbolje moguće potrebno ga je zabilježiti na sljedeći način:  $z^* = 9$
2. Ako je rješenje potproblema manje od trenutno najboljeg mogućeg onda se taj potproblem može odbaciti
3. Ako se, rješavanjem relaksiranog potproblema, dobije nemoguće rješenje onda se taj potproblem može odbaciti

### 2.4. Primjer cjelobrojnog programiranja

U ovome poglavlju će se do kraja riješiti problem iz prethodnog poglavlja. U koraku procjenjivanja se odbacio potproblem 1 te je samo potproblem 2 ostao aktivan za grananje. Sljedeća varijabla po kojoj se dijeli je  $x_2$ . Nakon podjele dobiju se sljedeća dva potproblema:

Potproblem 3 ( $x_1 = 0, x_2 = 0$ ):

$$\max z = 9 + 6x_3 + 4x_4$$

$$5x_3 + 2x_4 \leq 4$$

$$x_3 + x_4 \leq 1$$

$$x_3 \leq 1$$

$$x_4 \leq 0$$

Potproblem 4 ( $x_1 = 1, x_2 = 1$ ):

$$\max z = 14 + 6x_3 + 4x_4$$

$$5x_3 + 2x_4 \leq 1$$

$$x_3 + x_4 \leq 1$$

$$x_3 \leq 1$$

$$x_4 \leq 1$$

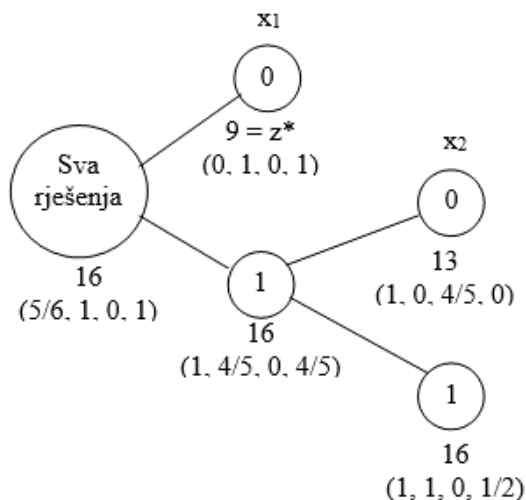
Nakon rješavanja relaksiranih problema dobiju se:

$$\text{Potproblem 3: } (x_1, x_2, x_3, x_4) = \left(1, 0, \frac{4}{5}, 0\right); \quad z = 13\frac{4}{5}$$

$$\text{Potproblem 4: } (x_1, x_2, x_3, x_4) = \left(1, 1, 0, \frac{1}{2}\right); \quad z = 16$$

Obje vrijednosti funkcije cilja su veća od trenutno najboljeg mogućeg rješenja te se niti jedan potproblem ne može odbaciti. Isto tako niti jedan od dobivenih rješenja ne daje novo moguće

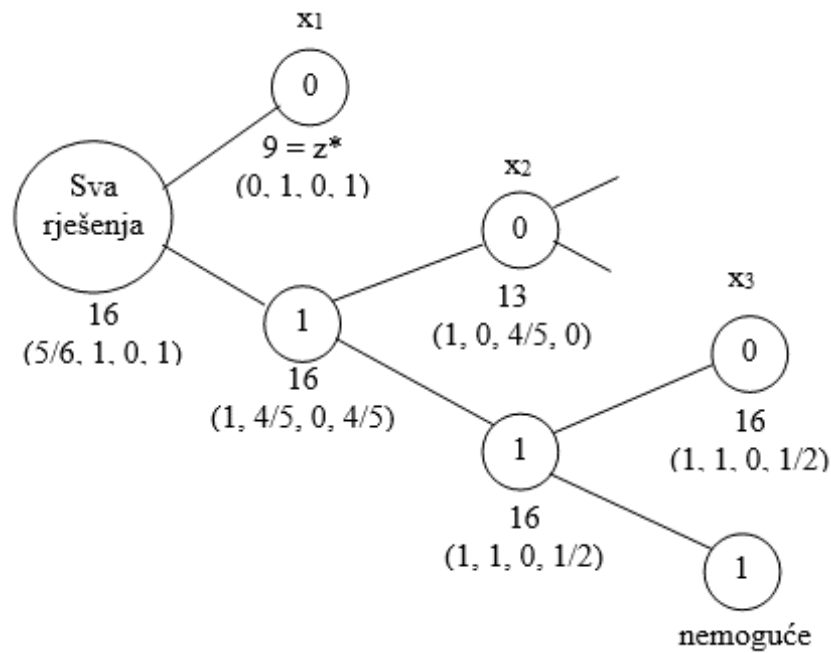
rješenje pošto nisu sve vrijednosti cjelobrojne. Na Slici 4 je prikazano stablo rješenja nakon drugog koraka.



Slika 4 Stablo rješenja nakon drugog koraka

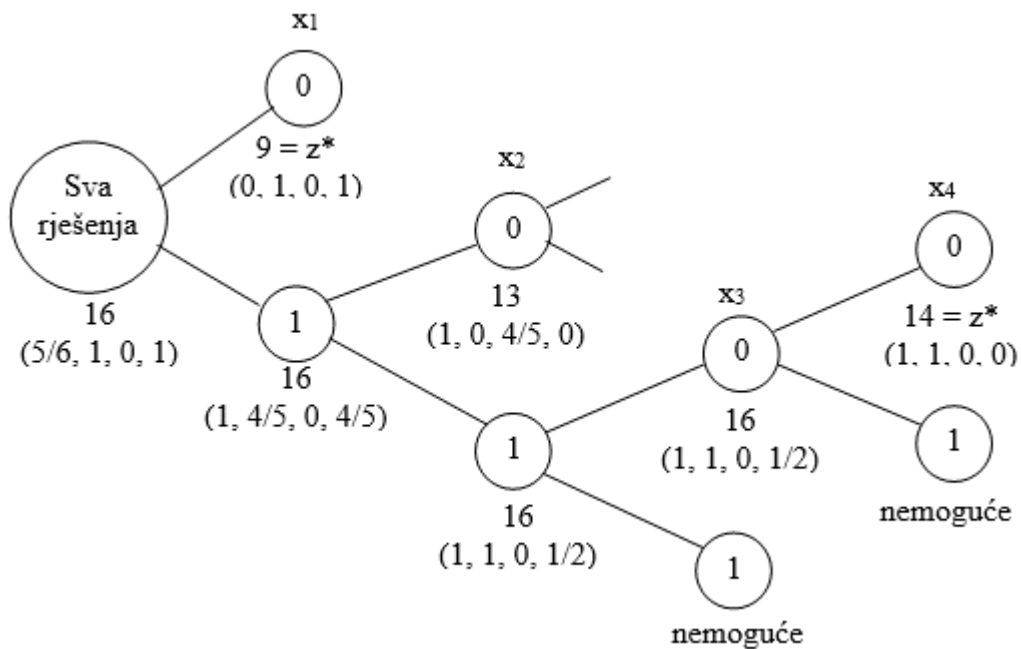
Nakon drugog koraka oba potproblema su ostala aktivna te je sada potrebno odlučiti s kojim od njih je bolje nastaviti. U ovome primjeru potproblem 4 ima veću vrijednost funkcije cilja te se grananje nastavlja njime.

U trećem koraku, nakon grananja potproblema 4 i rješavanja relaksiranih problema, potproblem 6 daje nemoguće rješenje te ga može odbaciti dok potproblem 5 daje rješenje koje nije manje od trenutno najboljeg te ostaje aktivan. Na Slici 5 je prikazano stablo rješenja nakon trećeg koraka.



Slika 5 Stablo rješenja nakon trećeg koraka

U četvrtom koraku se grananjem potproblema 5 dobiva nova dva. Potproblem 8 ima nemoguće rješenje te se odbacuje. Potproblem 7 ima moguće rješenje te su sve vrijednosti varijabli cjelobrojne. Pošto rješenje potproblema 7 ima vrijednost 14, koja je veća od trenutnog najboljeg mogućeg, ona postaje nova trenutna najbolja. Na Slici 6 je prikazano stablo nakon četvrtog koraka rješavanja problema.



Slika 6 Stablo rješenja nakon četvrtog koraka



Na kraju četvrtog koraka ostao je samo potproblem 3 koji ima vrijednost rješenja 13. Pošto je ta vrijednost manja od trenutno najbolje taj potproblem se odbacuje te time završava postupak grananja i ograđivanja. Kao završno najbolje cjelobrojno rješenje dobiveno je:

$$(x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 0); z = 14;$$

## 2.5. Pregled postojećih aplikacija

Prije početka izrade aplikacije bilo je potrebno proučiti već postojeće programe za izradu rasporeda, pronaći njihove nedostatke i pokušati implementirati većinu tokom izrade. Proučeni su mnogi programi, ali ABC Roster i 247 Shift su najviše utjecali na mogućnosti koje se moraju implementirati.

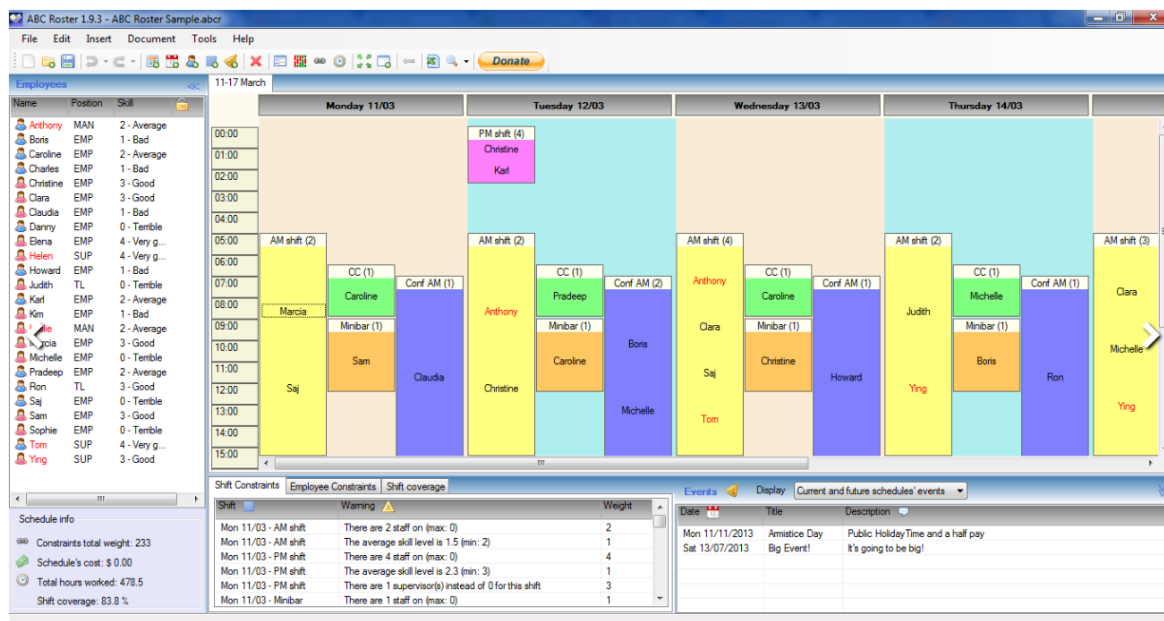
### 2.5.1. ABC Roster

ABC Roster [8] besplatna je aplikacija za izradu rasporeda smjena. U početku je bila zamišljena kao aplikacija za kafiće i restorane, ali se proširila na razne ustanove kojima je potreban takav program. Aplikacija nudi mogućnost unosa detaljnih podataka o radnicima te automatske izrade mjesečnih rasporeda. Prije izrade rasporeda moguće je detaljno unijeti mnoštvo detaljnih ograničenja kao:

- Maksimalan broj smjena po danu
- Broj slobodnih dana za željeni mjesec
- Minimalan i maksimalan broj sati za odraditi
- Vrste smjena koje radnik ne želi raditi (noćne, jutarnje, itd.)
- Imena radnika s kojima radnik ne želi biti u smjeni

Nakon izrade raspored je moguće izvesti u Excel formatu i poslati elektroničkom poštom odabranim radnicima. Iako ima puno dobrih funkcionalnosti nedostaje joj nekolicina ključnih. Glavni nedostaci su:

- Nemogućnost izmjene rasporeda nakon automatske izrade
- Zastarjeli izgled sučelja
- Aplikacija ne pohranjuje i ne koristi podatke iz baze podataka na internetu već se oni nalaze lokalno na računalu. To onemogućuje dostupnost podataka s različitih mjesta.
- Jedan korisnik je vezan na samo jednu ustanovu



Slika 7 Sučelje aplikacije ABC Roster

## 2.5.2. 247 Shift

247 Shift [9] je aplikacija koja je besplatna za do 10 korisnika. Služi za izradu detaljnih rasporeda koje je moguće izmijeniti po želji, pospremiti i pristupiti s bilo koje lokacije. Predviđena je za veće tvrtke koje imaju više odjela te je moguće za svaki odjel izraditi zaseban raspored. Odjeli tvrtke ne moraju biti na istoj lokaciji već je moguće imati globalnu raspršenost te pomoću Google Karte vidjeti i izraditi rasporede za svaku lokaciju. Dostupna je i kao mobilna aplikacija pomoću koje se lako mogu pregledati svi rasporedi, slati poruke drugim radnicima i zahtjeve za slobodne dane. Aplikacija nudi jake analitičke alate pomoću kojih je moguće pratiti efikasnost svakog radnika, odjela i cijele tvrtke. Svaki raspored je moguće izvesti u Excel datoteku. Iako je aplikacija veoma dobro osmišljena i nudi mnoštvo mogućnosti, ima i par ključnih nedostataka. Jedan od ključnih jest nepostojanje izračuna troškova. Nije moguće unijeti satnice za radnike kako bi se nakon izrade rasporeda dobila okvirna cifra mjesečnog troška. Nakon izrade rasporeda nije moguće nadodati nove smjene ili obrisati postojeće već je samo zamjena između dvije moguća.

247 Shift

Welcome, Sarah Kim | Quicklinks | Help

Dashboard

View Schedule

Manage Schedule

Leave Request

Reports

Dashboard

Alerts

Request pending your approval [1]

Your request pending for approval [1]

Schedule

Show My Schedule

Show Department Schedule

03 Jan 2014 - 09 Jan 2014

Show More

Staff [Dept./Designation.]	Fri 03	Sat 04	Sun 05	Mon 06	Tue 07	Wed 08	Thu 09
Joshua Sinclair [CS./CSR.]	CBay 09:00am-04:00pm	CBay 09:00am-04:00pm	CBay 09:00am-04:00pm	CBay 09:00am-04:00pm			
Pauline Loo [CS./CSR.]	CBay 09:00am-04:00pm						

First

Prev

Page 1 of 1

Next

Last

Total : 2 Staff

Slika 8 Sučelje aplikacije 247 Shift

### 3. Izrada modela odabranog problema

Za lakše shvaćanje problema i izradu aplikacije prikupljeni su stvarni podaci iz jednog poznatog kafića u Rijeci. Na sastanku s vlasnikom je dogovoreno da će ime kafića i njegovih djelatnika ostati anonimno. Vlasnik je također napomenuo da su mjesečni troškovi za plaće svaku puta viši nego planirani radi krivo raspoređenih smjena i nepotrebnih prekovremenih sati.

#### 3.1. Prikupljeni podaci i ograničenja

Ugostiteljski objekt ima sveukupno dva stalna radnika i četiri studenta. Satnica za studente iznosi 20 kn po satu dok stalni zaposlenici imaju fiksnu plaću od 4000 kn i 5000 kn. Svaki stalni radnik morao je odraditi točno 40 radnih sati kroz tjedan, odnosno 160 sati mjesečno dok svaki student nije imao nikakvih ograničenja po tom pitanju. Stalni zaposlenici moraju odraditi smjenu od 8 sati dnevno dok svaki student mora odraditi minimalno 4 sata. Ponedjeljkom, srijedom i četvrtkom ugostiteljski objekt je imao smanjeni obujam posla dok utorkom, petkom i subotom povećani obujam što je zahtijevalo veći broj radnika kroz dvije smjene. Stalnim zaposlenicima je satnica 50% veća ukoliko rade prekovremeno, a ako rade noću onda je satnica 30% veća. Rad noću se računa od 22 h nadalje. Onaj student koji kroz tjedan odradi najmanje sati, taj student odrađuje smjenu nedjeljom.

Broj potrebnih radnika po danu kroz jedan tjedan prikazan je tablicom 1.

Tablica 1 Potrebni radnici po danu

Radni dan	Broj potrebnih radnika
Ponedjeljak	2
Utorak	3-4
Srijeda	2
Četvrtak	2
Petak	3-4
Subota	3-4
Nedjelja	1

U Tablici 2 je prikazan stvarni raspored radnika za 2. mjesec u 2017. godini.

Tablica 2 Raspored konobara kroz 2. mjesec u 2017. godini

<b>SRI</b>	<i>osoba</i>	Radnik 1	Student 1		
01.02.	<i>vrijeme</i>	8-16	16-24		
<b>ČET</b>	<i>osoba</i>	Radnik 2	Student 2		
02.02.	<i>vrijeme</i>	8-16	16-24		
<b>PET</b>	<i>osoba</i>	Radnik 1	Radnik 2	Student 3	Student 2
03.02.	<i>vrijeme</i>	8-16	16-02	17-02	21-02
<b>SUB</b>	<i>osoba</i>		Radnik 1	Student 2	Student 3
04.02.	<i>vrijeme</i>		17-02	20-02	21-02
<b>NED</b>	<i>osoba</i>		Student 1		
05.02.	<i>vrijeme</i>		17-23		
<b>PON</b>	<i>osoba</i>	Radnik 1	Radnik 2		
06.02.	<i>vrijeme</i>	8-16	16-24		
<b>UTO</b>	<i>osoba</i>	Radnik 2	Radnik 1	Student 3	Student 2
07.02.	<i>vrijeme</i>	8-16	16-02	20-02	21-02
<b>SRI</b>	<i>osoba</i>	Radnik 2	Student 1		
08.02.	<i>vrijeme</i>	8-16	16-24		
<b>ČET</b>	<i>osoba</i>	Radnik 1	Radnik 2		
09.02.	<i>vrijeme</i>	8-16	16-24		

<b>PET</b>	<i>osoba</i>	Radnik 1	Radnik 2	Student 2	Student 1
10.02.	<i>vrijeme</i>	8-16	16-02	20-02	21-02
<b>SUB</b>	<i>osoba</i>		Radnik 1	Student 2	Student 4
11.02.	<i>vrijeme</i>		17-02	20-02	21-02
<b>NED</b>	<i>osoba</i>		Student 4		
12.02.	<i>vrijeme</i>		17-23		
<b>PON</b>	<i>osoba</i>	Radnik 1	Radnik 2		
13.02.	<i>vrijeme</i>	8-16	16-24		
<b>UTO</b>	<i>osoba</i>	Student 2	Radnik 2	Radnik 1	Student 3
14.02.	<i>vrijeme</i>	8-16	16-02	19-02	20-02
<b>SRI</b>	<i>osoba</i>	Student 1	Radnik 1		
15.02.	<i>vrijeme</i>	8-16	16-24		
<b>ČET</b>	<i>osoba</i>	Radnik 2	Radnik 1		
16.02.	<i>vrijeme</i>	8-16	16-24		
<b>PET</b>	<i>osoba</i>	Student 1	Radnik 1	Radnik 2	Student 3
17.02.	<i>vrijeme</i>	8-16	16-02	19-02	20-02
<b>SUB</b>	<i>osoba</i>		Student 3	Radnik 2	Student 2
18.02.	<i>vrijeme</i>		17-02	20-02	21-02
<b>NED</b>	<i>osoba</i>		Student 2		
19.02.	<i>vrijeme</i>		17-23		
<b>PON</b>	<i>osoba</i>	Radnik 1	Radnik 2		
20.02.	<i>vrijeme</i>	8-16	16-24		
<b>UTO</b>	<i>osoba</i>	Student 2	Radnik 1	Student 4	Student 1
21.02.	<i>vrijeme</i>	8-16	16-02	19-02	20-02
<b>SRI</b>	<i>osoba</i>	Student 2	Radnik 1		
22.02.	<i>vrijeme</i>	8-16	16-24		
<b>ČET</b>	<i>osoba</i>	Radnik 1	Student 3		
23.02.	<i>vrijeme</i>	8-16	16-24		
<b>PET</b>	<i>osoba</i>	Student 2	Student 3	Radnik 2	Student 4
24.02.	<i>vrijeme</i>	8-16	16-02	19-02	20-02
<b>SUB</b>	<i>osoba</i>		Radnik 2	Student 4	Student 3

25.02.	<i>vrijeme</i>		17-02	20-02	21-02
<b>NED</b>	<i>osoba</i>	Student 3	Student 2	Radnik 2	Radnik 1
26.02.	<i>vrijeme</i>	10-23	10-23	12-23	13-23
<b>PON</b>	<i>osoba</i>	Radnik 1	Radnik 2		
27.02.	<i>vrijeme</i>	8-16	16-24		
<b>UTO</b>	<i>osoba</i>	Radnik 1	Radnik 2	Student 4	Student 2
28.02.	<i>vrijeme</i>	8-16	16-02	19-02	20-02

### 3.2. Postavljanje matematičkog modela

Rješavanje problema započeto je određivanjem varijabli odlučivanja nakon čega je definirana funkcija cilja, te ograničenja na funkciju cilja. Prvi korak izrade modela je bilo pojednostavljenje radnih dana u tjednu tako da ih se grupiralo u dvije skupine:

1. Dani kada su potrebna samo dva radnika koji su podijeljeni u dvije smjene (ponedjeljak, srijeda, četvrtak)
2. Dani kada su potrebna tri ili više radnika koji su podijeljeni u pet smjena (utorak, petak, subota)

Tablice 3 i 4 prikazuju raspored smjena za dvije grupe radnih dana te broj potrebnih radnika za svaku smjenu.

*Tablica 3 Smjene za dane kada nije potrebno više od dva radnika*

Radno vrijeme	Broj potrebnih radnika
8-16	1
16-24	1

Tablica 4 Smjene za dane kada je potrebno tri ili više radnika

Radno vrijeme	Broj potrebnih radnika
8-16	1
16-20	1
20-22	2
22-24	4
24-02	3

Kako bi se lakše raspoznali radnici u rasporedu svaka varijabla će označavati jednog radnika.

Popis varijabli:

$R_{111}, R_{112}, R_{121}, R_{122}, R_{211}, R_{212}, R_{221}, R_{222}, S_{111}, S_{112}, S_{121}, S_{122}, S_{123}, S_{124}, S_{211}, S_{212}, S_{221}, S_{222}, S_{223}, S_{224},$

$S_{311}, S_{312}, S_{321}, S_{322}, S_{323}, S_{324}, S_{411}, S_{412}, S_{421}, S_{422}, S_{423}, S_{424}$

Slova R označavaju stalne radnike dok slova S označavaju studente. Indeksi označavaju redni broj radnika, grupu radnog dana i smjenu kada radnik mora biti prisutan na poslu.

$R_{111}$  – prvi radnik koji može raditi u ponedjeljak, srijedu ili četvrtak u prvoj smjeni (od 8 h do 16 h)

$R_{112}$  - prvi radnik koji može raditi u ponedjeljak, srijedu ili četvrtak u drugoj smjeni (od 16 h do 24 h)

$R_{211}$  – drugi radnik koji može raditi u utorak, petak ili subotu u prvoj smjeni (od 8 h do 16 h)

$R_{212}$  - drugi radnik koji može raditi u utorak, petak ili subotu u drugoj smjeni (od 16 h do 24 h)

$R_{121}$  - prvi radnik koji može raditi u ponedjeljak, srijedu ili četvrtak u prvoj smjeni (od 8 h do 16 h)

$R_{122}$  - prvi radnik koji može raditi u ponedjeljak, srijedu ili četvrtak u drugoj smjeni (od 16 h do 24 h)

$R_{221}$  - drugi radnik koji može raditi u utorak, petak ili subotu u prvoj smjeni (od 8 h do 16 h)

$R_{222}$  - drugi radnik koji može raditi u utorak, petak ili subotu u drugoj smjeni (od 16 h do 24 h)

$S_{111}$  - prvi student koji može raditi u ponedjeljak, srijedu ili četvrtak u prvoj smjeni (od 8 h do 16 h)

$S_{112}$  - prvi student koji može raditi u ponedjeljak, srijedu ili četvrtak u drugoj smjeni (od 16 h do 24 h)

$S_{121}$  - prvi student koji može raditi u utorak, petak ili subotu u prvoj smjeni (od 8 h do 16 h)

$S_{122}$  - prvi student koji može raditi u utorak, petak ili subotu u drugoj smjeni (od 16 h do 24 h)

$S_{123}$  - prvi student koji može raditi u utorak, petak ili subotu u trećoj smjeni (od 20 h do 02 h)

$S_{124}$  - prvi student koji može raditi u utorak, petak ili subotu u četvrtoj smjeni (od 22 h do 02 h)

$S_{211}$  - drugi student koji može raditi u ponedjeljak, srijedu ili četvrtak u prvoj smjeni (od 8 h do 16 h)

$S_{212}$  - drugi student koji može raditi u ponedjeljak, srijedu ili četvrtak u drugoj smjeni (od 16 h do 24 h)

$S_{221}$  - drugi student koji može raditi u utorak, petak ili subotu u prvoj smjeni (od 8 h do 16 h)

- S<sub>222</sub> - drugi student koji može raditi u utorak, petak ili subotu u drugoj smjeni (od 16 h do 24 h)
- S<sub>223</sub> - drugi student koji može raditi u utorak, petak ili subotu u trećoj smjeni (od 20 h do 02 h)
- S<sub>224</sub> - drugi student koji može raditi u utorak, petak ili subotu u četvrtoj smjeni (od 22 h do 02 h)
- S<sub>311</sub> - treći student koji može raditi u ponedjeljak, srijedu ili četvrtak u prvoj smjeni (od 8 h do 16 h)
- S<sub>312</sub> - treći student koji može raditi u ponedjeljak, srijedu ili četvrtak u drugoj smjeni (od 16 h do 24 h)
- S<sub>321</sub> - treći student koji može raditi u utorak, petak ili subotu u prvoj smjeni (od 8 h do 16 h)
- S<sub>322</sub> - treći student koji može raditi u utorak, petak ili subotu u drugoj smjeni (od 16 h do 24 h)
- S<sub>323</sub> - treći student koji može raditi u utorak, petak ili subotu u trećoj smjeni (od 20 h do 02 h)
- S<sub>324</sub> - treći student koji može raditi u utorak, petak ili subotu u četvrtoj smjeni (od 22 h do 02 h)
- S<sub>411</sub> - četvrti student koji može raditi u ponedjeljak, srijedu ili četvrtak u prvoj smjeni (od 8 h do 16 h)
- S<sub>412</sub> - četvrti student koji može raditi u ponedjeljak, srijedu ili četvrtak u drugoj smjeni (od 16 h do 24 h)
- S<sub>421</sub> - četvrti student koji može raditi u utorak, petak ili subotu u prvoj smjeni (od 8 h do 16 h)
- S<sub>422</sub> - četvrti student koji može raditi u utorak, petak ili subotu u drugoj smjeni (od 16 h do 24 h)
- S<sub>423</sub> - četvrti student koji može raditi u utorak, petak ili subotu u trećoj smjeni (od 20 h do 02 h)
- S<sub>424</sub> - četvrti student koji može raditi u utorak, petak ili subotu u četvrtoj smjeni (od 22 h do 02 h)

Za formuliranje funkcije cilja potrebno je izračunati cijenu svake smjene za svakog radnika na sljedeći način:

1. Ako smjena ne spada pod noćni rad, onda se trajanje smjene pomnoži sa satnicom radnika
2. Ako smjena spada pod noćni rad, onda se trajanje smjene pomnoži sa satnicom radnika uvećanom za 30%

Nakon što je postavljen početni matematički model izračunato je rješenje te je ono iskorišteno kako bi se detaljnije optimizirao model. Analizom dobivenih rezultata primijećeno je da se najviše isplati ako student radi minimalno 4, a maksimalno 10 sati.

Zadnji korak je bilo uvođenje rada nedjeljom. Nedjeljno radno vrijeme je od 17 do 23 sata te tu smjenu uvijek odrađuje student. Cijena te smjene iznosi 126 kn te taj iznos je potrebno pomnožiti s brojem nedjelja u željenom mjesecu.



Konačna ograničenja i funkcija cilja su sljedeća:

$$R_{111} + R_{211} + S_{111} + S_{211} + S_{311} + S_{411} = 1$$

$$R_{112} + R_{212} + S_{112} + S_{212} + S_{312} + S_{412} = 1$$

$$R_{121} + R_{221} + S_{121} + S_{221} + S_{321} + S_{421} = 1$$

$$R_{122} + R_{222} + S_{122} + S_{222} + S_{322} + S_{422} = 1$$

$$R_{122} + R_{222} + S_{122} + S_{222} + S_{322} + S_{422} + S_{123} + S_{223} + S_{323} + S_{423} = 2$$

$$R_{122} + R_{222} + S_{122} + S_{222} + S_{322} + S_{422} + S_{123} + S_{223} + S_{323} + S_{423} + S_{124} + S_{224} + S_{324} + S_{424} = 4$$

$$S_{122} + S_{222} + S_{322} + S_{422} + S_{123} + S_{223} + S_{323} + S_{423} + S_{124} + S_{224} + S_{324} + S_{424} = 3$$

$$R_{111} + R_{112} = 1$$

$$R_{121} + R_{122} = 1$$

$$R_{211} + R_{212} = 1$$

$$R_{221} + R_{222} = 1$$

$$S_{111} + S_{112} \leq 1;$$

$$S_{211} + S_{212} \leq 1;$$

$$S_{311} + S_{312} \leq 1;$$

$$S_{411} + S_{412} \leq 1;$$

$$S_{121} + S_{122} + S_{123} + S_{124} \leq 1;$$

$$S_{221} + S_{222} + S_{223} + S_{224} \leq 1;$$

$$S_{321} + S_{322} + S_{323} + S_{324} \leq 1;$$

$$S_{421} + S_{422} + S_{423} + S_{424} \leq 1;$$

$$R_{ijk} \leq 1 \quad \forall \quad i = 1, 2 \quad j = 1, 2 \quad k = 1, 2$$

$$R_{ijk} \geq 0 \quad \forall \quad i = 1, 2 \quad j = 1, 2 \quad k = 1, 2$$

$$S_{ijk} \leq 1 \quad \forall \quad i = 1, 2, 3, 4 \quad j = 1, 2 \quad k = 1, 2, 3, 4$$

$$S_{ijk} \geq 0 \quad \forall \quad i = 1, 2, 3, 4 \quad j = 1, 2 \quad k = 1, 2, 3, 4$$

$$R_{ijk} \geq 0 \quad \forall i = 1, 2 \quad j = 1, 2 \quad k = 1, 2$$

$$\begin{aligned} \min Z = & 268,75 * (R_{212} + R_{222}) + 250 * (R_{211} + R_{221}) + 215 * (R_{112} + R_{122}) + 200 * \\ & (R_{111} + R_{121}) + 224 * (S_{122} + S_{222} + S_{322} + S_{422}) + 172 * (S_{112} + S_{212} + S_{312} + S_{412}) + \\ & 160 * (S_{111} + S_{211} + S_{311} + S_{411} + S_{121} + S_{221} + S_{321} + S_{421}) + 144 * (S_{123} + S_{223} + \\ & S_{323} + S_{423}) + 104 * (S_{124} + S_{224} + S_{324} + S_{424}) \end{aligned}$$

### 3.3. Korištene tehnologije

Postoje mnogi programi koji omogućavaju rješavanje problema cjelobrojnog programiranja. Najpoznatiji program je LP Solve [10] te će isti biti korišten u ovome radu.

LP Solve je besplatan program otvorenoga koda. Uspješno rješava standardne linearne programe koristeći simpleks metodu, ali i omogućuje rješavanje cjelobrojnih linearnih programa pomoću postupka grananja i ograđivanja. LP Solve ima jednostavno grafičko sučelje pomoću kojega je s lakoćom moguće zapisati model linearnog programa.

Slika 9 prikazuje izgled sučelja programa te dio napisanog modela linearnog programiranja.

```

1 /* Objective function */
2 min: 268.75*R212+268.75*R222+250*R211+250*R221+215*R112+215*R122+200*R111+200*R121+224*S122+224*S222+224*S322+224*S422;
3
4 /* Variable bounds */
5 R111+R211+S111+S211+S311+S411=1;
6 R112+R212+S112+S212+S312+S412=1;
7 R121+R221+S121+S221+S321+S421=1;
8 R122+R222+S122+S222+S322+S422=1;
9 R122+R222+S122+S222+S322+S422+S123+S223+S323+S423=2;
10 R122+R222+S122+S222+S322+S422+S123+S223+S323+S423+S124+S224+S324+S424=4;
11 S122+S222+S322+S422+S123+S223+S323+S423+S124+S224+S324+S424=3;
12 R111+R112=1;
13 R121+R122=1;
14 R211+R212=1;
15 R221+R222=1;
16 S111+S112<=1;
17 S211+S212<=1;
18 S311+S312<=1;
19 S411+S412<=1;
20 S121+S122+S123+S124<=1;
21 S221+S222+S223+S224<=1;
22 S321+S322+S323+S324<=1;
23 S421+S422+S423+S424<=1;
24 S111<=1;
25 S112<=1;
26 S121<=1;
27 S122<=1;
28 S123<=1;
29 S124<=1;
30 S211<=1;
31 S212<=1;
32 S221<=1;
33 S222<=1;
34 S223<=1;
35 S224<=1;

```

Slika 9 Izgled sučelja programa LP Solve

### 3.4. Izrada modela za aplikaciju

Kako bi aplikacija mogla uspješno odrediti optimalan raspored potrebno je dodatno proširiti matematički model iz prethodnog primjera. Varijable su u prethodnom modelu bile označene sa slovima R i S. U aplikaciji je potrebno dodatno specificirati ime varijable kako bi se točno znalo na kojeg radnika se odnosi. Pošto je korisničko ime jedinstveno za svakog korisnika ono će se koristiti umjesto slova R i S. Varijable su imenovane sa sljedeći način:

[korisničko ime]\_[y]\_[z] – Varijabla radnika s korisničkim imenom za dan y za smjenu z.

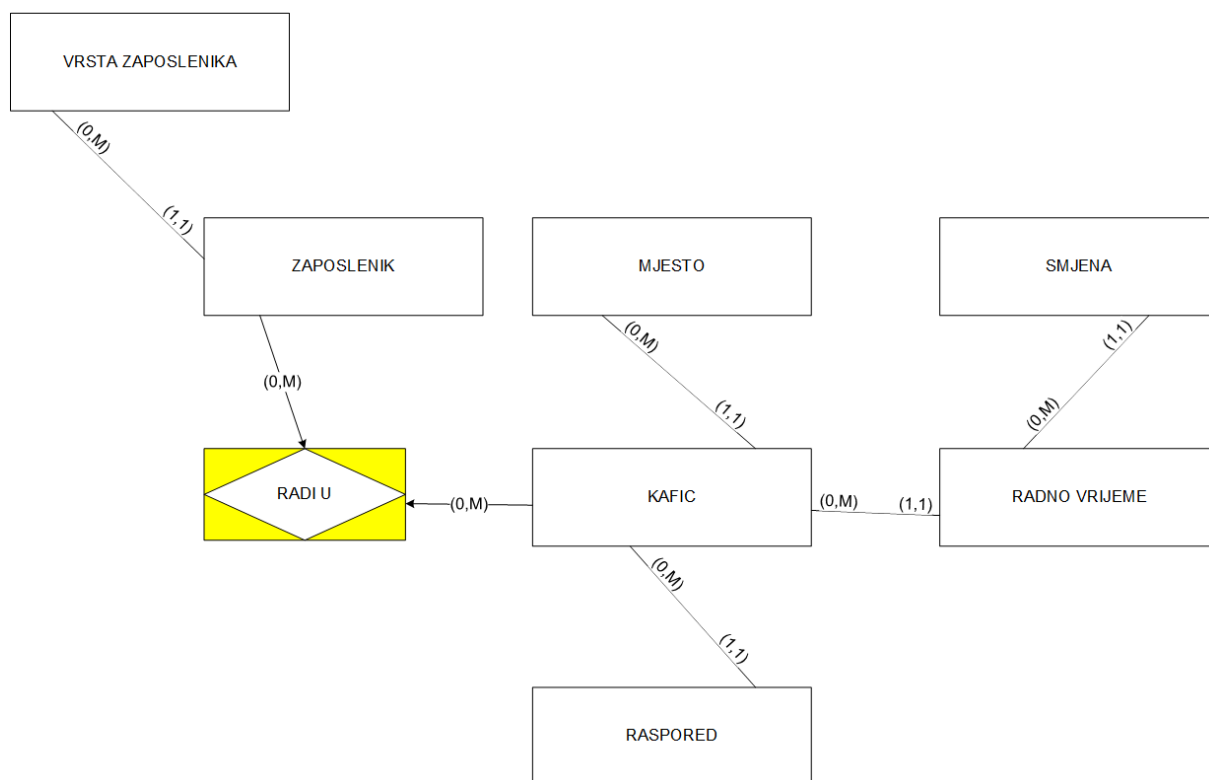
U prethodnom modelu su se dani dijelili na dvije vrste, manje i više frekventni. Za lakše raspoznavanje dana, parametar „y“ će označavati dan u željenom mjesecu za izračun.

Parametar „z“ označava redni broj smjene za dan „y“. Kada radnik za jedan dan „y“ može doći samo u jednu smjenu, onda je parametar „z“ označena kao broj smjene. Kada radnik može doći u više smjena, npr. radnik je došao u drugu smjenu i ostao do pete smjene, onda je varijabla „z“ zapisana kao „a:b“ di je „a“ početna smjena, a „b“ završna smjena za taj dan.

## 4. Izrada aplikacije za raspoređivanje radnika u ugostiteljstvu

Izrada aplikacije je započeta izradom modela baze podataka. Nakon izrade modela bilo je potrebno napraviti bazu podataka te ju pospremiti na poslužitelja kako bi joj se mogli pristupiti od bilo kuda. Odabran je poslužitelj GearHost [11] koji nudi besplatno posluživanje (*eng. hosting*) mnogih vrsta baza podataka do veličine od 5 MB. Nakon što je uspješno omogućeno posluživanje baze podataka započeta je izrada aplikacije koristeći Spring okvir [12]. Napravljen je algoritam koji, pomoću potrebnih podataka o ugostiteljskom objektu i njegovim radnicima, izradi model cjelobrojnog linearnog programiranja te ga pomoću SSC (*eng. Software for the Calculation of the Simplex*) Java biblioteke [13] riješi koristeći algoritam Grananja i Ograđivanja. Nakon što je stvoren raspored za željeni mjesec, i odobren od strane vlasnika, posprema se u bazu podataka te je dostupan svim radnicima ugostiteljskog objekta.

## 4.1. DEV model baze podataka



Slika 10 DEV model baze podataka

## 4.2. Relacijski model baze podataka

**VRSTA ZAPOSLENIKA** ( **id\_vrsta\_zap**, naziv\_vrste\_zap)

**ZAPOSLENIK** ( **id\_zap**, ime\_zap, prezime\_zap, spol\_zap, godina\_rodenja\_zap, kontakt\_zap, mail\_zap, id\_vrsta\_zap)

**KAFIC** ( **id\_kafic**, naziv\_kafic, ulica\_kafic, postotak\_nocni\_rad, postotak\_prekovremeni, postotak\_blagdan, id\_mjesto)

**RADI U** ( **id\_zap**, **id\_kafic**, satnica, placen\_nocni\_rad, placeni\_prekovremeni, placeni\_blagdani, min\_sati\_tjedno)

**MJESTO** ( **id\_mjesto**, postanski\_broj, naziv\_mjesto)

**RADNO VRIJEME** ( **id\_rad\_vrijeme**, radni\_dan, vrijeme\_otvaranja, vrijeme\_zatvaranja, id\_kafic)

**SMJENA** ( **id\_smjena**, od\_sati, do\_sati, id\_rad\_vrijeme)

**RASPORED** ( **id\_raspored**, mjesec, godina, kalendar\_podaci, id\_kafic)

### 4.3. Odabir programske podrške

Kao srž cijele aplikacije odabran je Spring okvir [12]. Spring je okvir otvorenoga koda koji je savršen za izradu Java web aplikacija. Sličan je Java Enterprise Edition, ali podržava puno više mogućnosti. Samim time je u zadnjih tri godine postao najbolja zamjena za Java Enterprise Edition.

Prvu verziju okvira napisao je Rod Johnson 2002. godine. 2004. godine je objavljena prva verzija okvira te je od tada prikupio mnoštvo prestižnih nagrada kao što je JAX (Java API for XML) nagradu za inovaciju.

Za pohranu podataka o ugostiteljskim objektima izrađena je MySQL baza podataka. Za pristup i pospremanje podataka u bazu Spring okvir sadrži komponentu zvanu „*Spring Data*“ [14] koja, uz nekoliko podešenih parametara, s lakoćom se povezuje s bazom podataka i odrađuje sav potreban posao.

Za izradu rasporeda korištena je SSC (*eng. Software for the Calculation of the Simplex*) Java biblioteka [13]. Napisana od strane studenta Stefana Scariolia kao diplomski rad na sveučilištu u Rimu. Biblioteka podržava rješavanje problema cjelobrojnog linearnog programiranja koristeći prethodno opisani algoritam grananja i ograđivanja.

Najbitniji element aplikacije je bio prikaz izrađenih rasporeda i mogućnost detaljne izmjene smjena. Full Calendar [15] okvir nudi mnoštvo mogućnosti kako bi se s lakoćom prikazao kalendar smjena, dopustile izmjene na način da korisnik na intuitivan način dodaje, briše, produžuje ili smanjuje željene smjene. Full Calendar je okvir otvorenoga koda te je s time njegovo korištenje potpuno besplatno.

Za integrirano razvojno okruženje (*engl. IDE – Integrated development environment*) odabran je Spring Tool Suite [16]. Razvojni alat se bazira na Eclipse razvojnom programu. Pomoću tog alata puno je lakše implementirati, pokrenuti i objaviti aplikacije koje koriste Spring okvir.

### 4.4. Aplikacija

Aplikacija omogućuje svakom poslodavcu za svaki kafić, čiji je vlasnik, unos svih potrebnih podataka o kafiću, zaposlenicima, radnim vremenima i smjenama. Nakon unosa svih potrebnih podataka vlasnik je u mogućnosti izraditi početni raspored smjena za željeni mjesec, izmijeniti ga po želji te ga pospremiti u bazu podataka. Slika 11 prikazuje popis vlastitih kafića s mogućnošću izmjene, brisanja te pregleda zaposlenih radnika.

Job Scheduler	Home	About	Contact	Welcome vlasnik	Logout
City List	My Bar List	Scheduler			

My Bars List							Add Bar		
Name	Address	Postcode	City	Percent Night Shift	Percent Overtime	Percent on Holiday	Edit	Delete	
Kafic 1	E. Randića 4	51000	RIJEKA	30.0	50.0	100.0			Employees
Kafic 2	Ulica Kafica 2	10000	ZAGREB	30.0	50.0	50.0			Employees

Slika 11 Popis vlastitih kafića

Slika 12 prikazuje popis svih radnika, neke od njihovih osobnih podataka, satnicu te dodatne bonuse na satnicu u određenim uvjetima.

Job Scheduler	Home	About	Contact	Welcome vlasnik	Logout
City List	My Bar List	Scheduler			

My Employee List													Add Employee	
First Name	Last Name	Gender	Date of Birth	Contact	E-Mail	Username	Role	Min. Hours Weekly	Night Shifts Payed Extra	Overtime Payed Extra	Holidays Payed Extra	Hourly Rate	Edit	Delete
Toni	Čavar	M	05.05.1992	0918949219	toni.cavar45@gmail.com	cavar45	admin	40.0 h	NO	YES	YES	50.0 kn		
Vlasnik	Vlasnik	M	05.05.1992	051674205	test@test.com	vlasnik	vlasnik	40.0 h	YES	NO	NO	50.0 kn		
Zap 2	Zap	F	21.05.1990	0918949219	zap2@gmail.com	zap2	zaposlenik	40.0 h	YES	YES	NO	25.0 kn		
Zap 4	Zap	F	14.06.1995	0918949219	zap4@gmail.com	zap4	zaposlenik	40.0 h	YES	YES	NO	31.25 kn		
Zap 3	Zap	F	14.05.1980	0918949219	zap3@gmail.com	zap3	student	0.0 h	YES	NO	NO	20.0 kn		
Zap 5	Zap	F	26.08.1992	0918949219	zap5@gmail.com	zap5	student	0.0 h	YES	NO	NO	20.0 kn		
Zap 6	Zap	M	14.08.1991	0918949219	zap6@gmail.com	zap6	student	0.0 h	YES	NO	NO	20.0 kn		

Slika 12 Popis zaposlenika kafića

Sljedeće slike prikazuju početni formular za izradu rasporeda. Na Slici 13 prikazana je stranica na kojoj vlasnik kafića može vidjeti sve zaposlenike kafića, njihove satnice, minimalne tjedne sate te postavke o dodatnim bonusima na plaću. Prilikom generiranja rasporeda uzima se u obzir da stalni radnici moraju odraditi 8 sati dnevno dok ne ispune potrebne uvjete. Za studente se ne uzimaju u obzir tjedni mjesečni sati, već se za njih u dodatnim postavkama unose minimalni i maksimalni dnevni sati. U dodatnim postavkama se još može označiti ako vlasnik konobari te da li su dopušteni prekovremeni.

Worker List

First Name	Last Name	Role	Min. Hours Weekly	Night Shifts Payed Extra	Overtime Payed Extra	Holidays Payed Extra	Hourly Rate
Toni	Čavar	admin	40.0 h	NO	YES	YES	50.0 kn
Vlasnik	Vlasnik	vlasnik	40.0 h	YES	NO	NO	50.0 kn
Zap 2	Zap	zaposlenik	40.0 h	YES	YES	NO	25.0 kn
Zap 4	Zap	zaposlenik	40.0 h	YES	YES	NO	31.25 kn
Zap 3	Zap	student	0.0 h	YES	NO	NO	20.0 kn
Zap 5	Zap	student	0.0 h	YES	NO	NO	20.0 kn
Zap 6	Zap	student	0.0 h	YES	NO	NO	20.0 kn

Additional Settings

Min. Hours Daily For Students :

Max. Hours Daily For Students :

☐

Owner Works as a Waiter

☐

Allow Overtime

Choose Month :

January ▼

Create

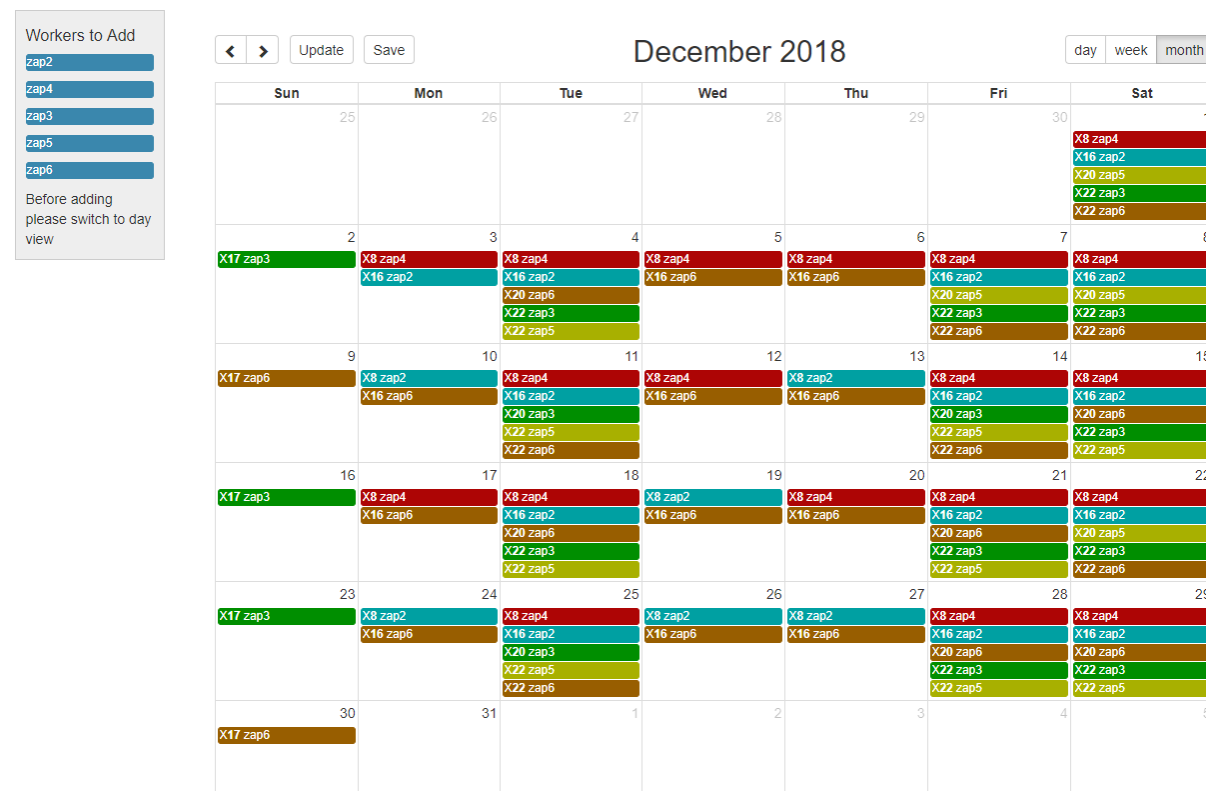
Slika 13 Početni formular za izradu rasporeda

Slika 14 i Slika 15 prikazuju dobivene rezultate nakon automatskog generiranja rasporeda. Tablica na Slici 14 ima dvije svrhe. Prva svrha je prikaz imena i prezimena radnika te njihovo ime unutar kalendara. Druga svrha je prikaz mjesečnih sati za svakog radnika i ukupan trošak plaća za njih.

Results			
First Name	Last Name	Calendar Name	Hours This Month
Zap 2	Zap	zap2	160 h
Zap 4	Zap	zap4	160 h
Zap 3	Zap	zap3	68 h
Zap 5	Zap	zap5	70 h
Zap 6	Zap	zap6	180 h
TOTAL COST		16683.0 KN	

Slika 14 Legenda radnika u kalendaru, broj mjesečnih sati i mjesečni trošak radnika

Slika 15 prikazuje početni izrađeni kalendar. Vlasnik može taj kalendar urediti kako god želi. Može povećati ili smanjiti smjenu radniku, nadodati novoga radnika u neku smjenu, obrisati smjenu radniku, itd. Nakon željenih izmjena pritiskom na gumb „*Update*“ Tablica rezultata i cijeli kalendar se ažuriraju s novim podacima. Kada je vlasnik zadovoljan izrađenim rasporedom, pritiskom na gumb „*Save*“, kompletan raspored se pohranjuje u bazu podataka.



*Slika 15 Generirani početni kalendar rasporeda*

Nakon što je raspored uspješno pospremljen u bazu svi zaposlenici imaju mogućnost pregleda, ali ne i izmjene, željenog rasporeda.



Slika 16 prikazuje popis svih postojećih rasporeda za određeni kafić. Klikom na gumb „View Schedule“ korisnik dobiva kalendarski prikaz rasporeda na pregled bez mogućnosti izmjene.

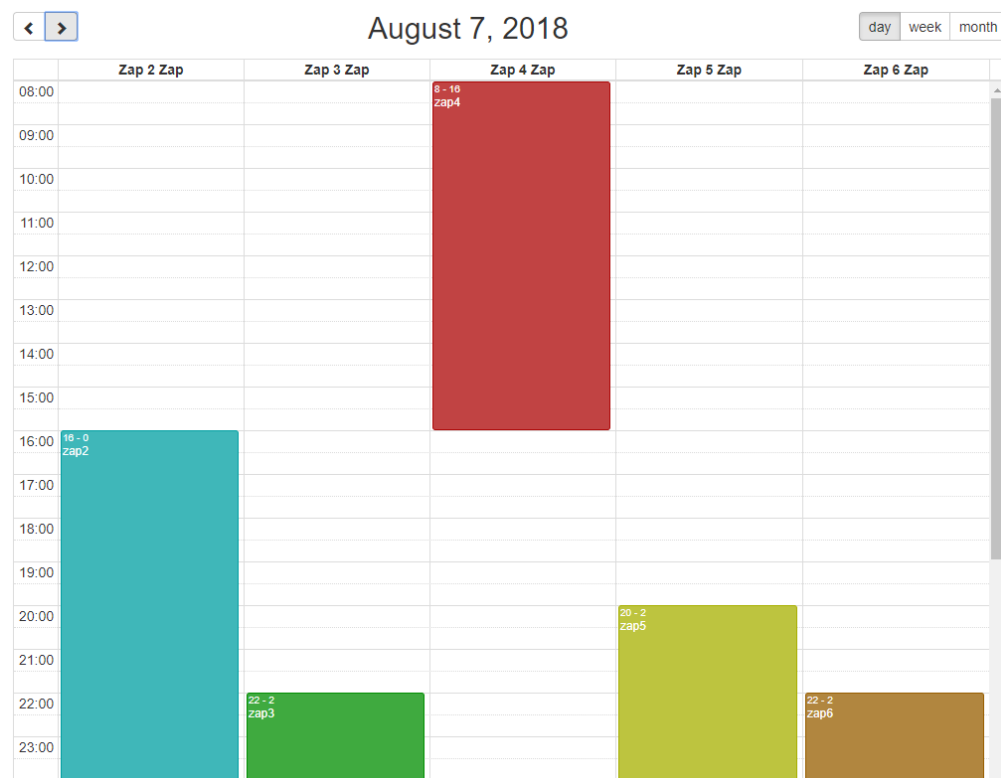
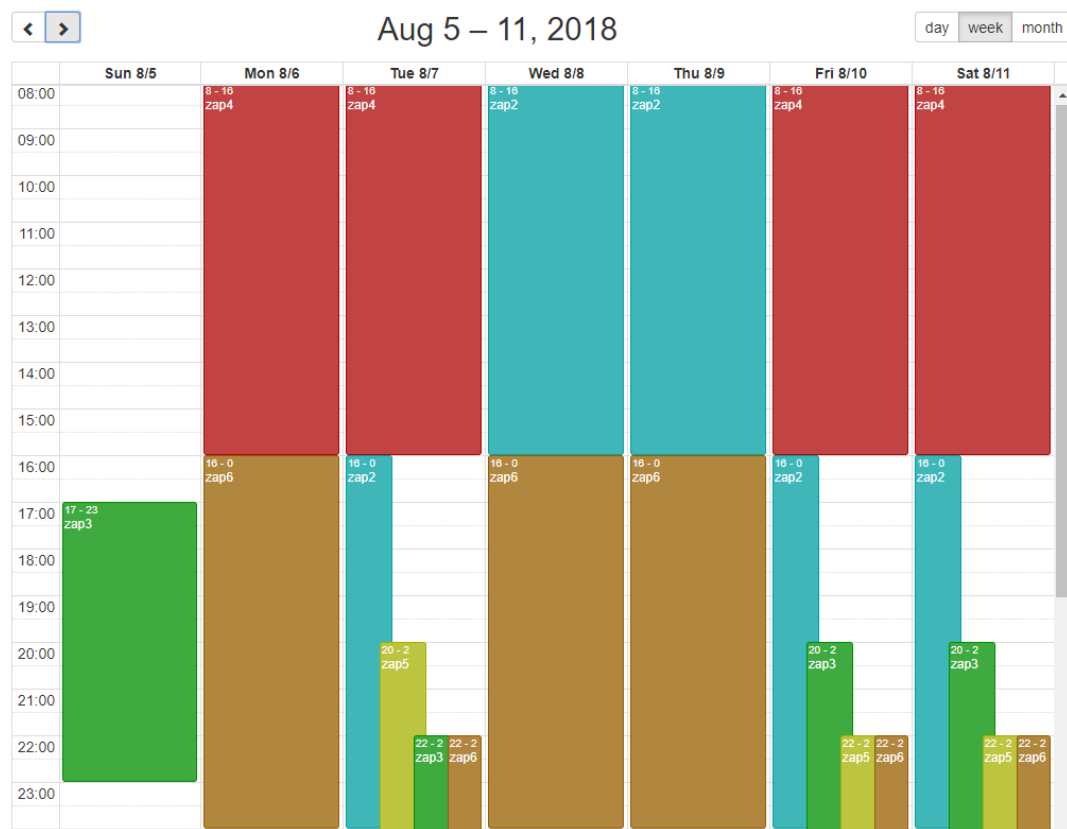
Job Scheduler	Home	About	Contact	Welcome cavar45	Logout
User List	Bar List	City List	Scheduler		
<b>Available Schedules</b>					
Month	Year				
August	2018	<a href="#">View Schedule</a>			
October	2018	<a href="#">View Schedule</a>			
June	2018	<a href="#">View Schedule</a>			

Slika 16 Popis generiranih rasporeda za željeni kafić

Na slikama 17, 18 i 19 je prikazan potvrđeni raspored od strane vlasnika za mjesec kolovoz iz mjesečne, tjedne i dnevne perspektive.

< >		August 2018					day	week	month
Sun	Mon	Tue	Wed	Thu	Fri	Sat			
29	30	31	1	2	3	4			
			8 zap4 16 zap6	8 zap4 16 zap6	8 zap4 16 zap2 20 zap5 22 zap3 22 zap6	8 zap4 16 zap2 20 zap6 22 zap3 22 zap5			
5	6	7	8	9	10	11			
17 zap3	8 zap4 16 zap6	8 zap4 16 zap2 20 zap5 22 zap3 22 zap6	8 zap2 16 zap6	8 zap2 16 zap6	8 zap4 16 zap2 20 zap3 22 zap5 22 zap6	8 zap4 16 zap2 20 zap3 22 zap5 22 zap6			
12	13	14	15	16	17	18			
17 zap5	8 zap2 16 zap6	8 zap4 16 zap2 20 zap3 22 zap5 22 zap6	8 zap2 16 zap6	8 zap4 16 zap6	8 zap4 16 zap2 20 zap3 22 zap5 22 zap6	8 zap4 16 zap2 20 zap6 22 zap3 22 zap5			
19	20	21	22	23	24	25			
17 zap6	8 zap4 16 zap6	8 zap4 16 zap2 20 zap6 22 zap3 22 zap5	8 zap2 16 zap6	8 zap2 16 zap6	8 zap4 16 zap2 20 zap5 22 zap3 22 zap6	8 zap4 16 zap2 20 zap3 22 zap5 22 zap6			
26	27	28	29	30	31	1			
17 zap5	8 zap2 16 zap6	8 zap4 16 zap2 20 zap6 22 zap3 22 zap5	8 zap4 16 zap6	8 zap4 16 zap6	8 zap4 16 zap2 20 zap6 22 zap3 22 zap5				
2	3	4	5	6	7	8			

Slika 17 Mjesečni prikaz rasporeda



Kod izrade aplikacije velika pažnja je posvećena sigurnosti. Spring okvir sadrži jaku komponentu za sigurnost pomoću koje s lakoćom korisnik može kreirati prilagođeni sustav prijava, odjava i pristupa elementima unutar aplikacije [17]. U ovoj aplikaciji sve korisnike unosi administrator te im zadaje predefinirane lozinke koje su korisnici dužni izmijeniti prilikom prve prijave. Slika 20 prikazuje formular za odabir nove lozinke s mogućnošću izmjene osobnih podataka.

The screenshot shows a web application interface for 'Job Scheduler'. At the top, there is a navigation bar with links for Home, About, and Contact, and a user greeting 'Welcome cavar45' with a Logout link. Below this is a section titled 'Edit My Info'. The form contains the following fields:

- Username : cavar45
- Password : Enter New Password
- First Name : Toni
- Last Name : Čavar
- Gender : M
- Date of Birth : 05.05.1992
- Contact Number : 0918949219
- E-Mail : toni.cavar45@gmail.com
- Role : admin (dropdown menu)

A 'Save' button is located at the bottom of the form.

*Slika 20 Odabir nove lozinke prilikom prve prijave*

## 4.5. Pregled i usporedba rješenja

Kako bi se utvrdila uspješnost izrađene aplikacije uspoređen je rezultat aplikacije za mjesec kolovoz s dobivenim rezultatom koristeći programom LP Solve.

U programu LP Solve, kao konačni rezultat, dobivena je vrijednost od 1.282 kn za dvije vrste dana. Mjesec kolovoz sadrži 26 radnih dana (13 manje frekventnih i 13 više frekventnih). Kako bi se dobio mjesečni trošak potrebno je dobivenu vrijednost pomnožiti s 13. Nakon množenja mjesečni trošak bez nedjelja iznosi 16.666 kn. Kada se pridoda cijena četiri nedjelja od 504 kn dobije se konačni mjesečni trošak od 17.170 kn. Koristeći aplikaciju dobiven je mjesečni trošak od 16.683 kn. Razlika između rezultata iznosi 487 kn. Iako mjesec kolovoz sadrži jedan blagdan te je aplikacija za taj dan satnice uvećala za određeni postotak, detaljnije izrađeni model unutar aplikacije pridonio je boljem optimalnom rješenju i većoj uštedi.

## 5. Zaključak

U radu je prikazano moguće rješenje problema raspodjele radnika metodom cjelobrojnog linearnog programiranja pomoću postupka grananja i ograđivanja. Linearno programiranje ima sve češću primjenu u stvarnom životu, što dokazuje konkretan problem raspodjele smjena u ugostiteljskom objektu koji je obrađen u ovome radu.

Izrađena aplikacija pokriva dinamične situacije kao što su godišnji odmori, bolovanja, blagdani, itd., te time omogućava da se izradi raspored po želji vlasnika i njegovih zaposlenika. Putem aplikacije vlasnik kafića s lakoćom može predvidjeti mjesečni trošak za plaće te optimalnije rasporediti mjesečni budžet kafića.

Kao mogućnost za daljnji rad aplikacija bi se mogla nuditi kao servis na internetu te time omogućiti pristup s bilo koje lokacije. U cilju optimiziranja troškova izračun mjesečnih plaća bi se mogao prikazati kao bruto svota. Kao zadnja ideja nadogradnje bi bila povezanost izrađenih kalendara s Google Kalendarom kako bi se olakšalo slanje izrade i promjena na e-mail zaposlenika.

## 6. Literatura

- [1] T. Mateljan i Ž. Jurić, »Radni materijali za kurs “Osnove operacionih istraživanja” na Elektrotehničkom fakultetu u Sarajevu,« 2014. [Mrežno]. Available: [https://c2.etf.unsa.ba/file.php/45/\\_POGLAVLJE\\_6\\_Problem\\_rasporedjivanja\\_asignacije\\_.pdf](https://c2.etf.unsa.ba/file.php/45/_POGLAVLJE_6_Problem_rasporedjivanja_asignacije_.pdf). [Pokušaj pristupa 6 8 2018].
- [2] M. Marinović i M. Holenko Dlab, »Materijal iz kolegija Operacijska Istraživanja 1, dostupni u sustavu za učenje Canvas,« 2016. [Mrežno]. Available: [https://canvas.instructure.com/courses/1074004/files/44662399?module\\_item\\_id=10707128](https://canvas.instructure.com/courses/1074004/files/44662399?module_item_id=10707128). [Pokušaj pristupa 15 7 2018].
- [3] W. Stemberger, »Uporaba genetskih algoritama u problemima raspoređivanja,« Fakultet elektrotehnike i računarstva, Zagreb, 1997.
- [4] M. L. Pinedo, Planning and Scheduling in Manufacturing and Services, Springer Science + Business Media, 2009.
- [5] Hillier i Lieberman, Introduction to Operations Research, McGraw - Hill, 2001.
- [6] R. Ostermann i D. de Werra, »Some experiments with a timetabling system,« *Operations Research Spektrum*, svez. 3, br. 4, pp. 199-204, 1982.
- [7] D. Kalpić i V. Mornar, Operacijska istraživanja, Zagreb: ZEUS, 1996.
- [8] ABC Roster, »ABC Roster,« ABC Roster, 2017. [Mrežno]. Available: <http://www.abc-roster.com/>. [Pokušaj pristupa 5 8 2018].
- [9] 247 Shift, »247 Shift,« Faster Smarter Solutions Limited, 2017. [Mrežno]. Available: <http://www.247shift.com/>. [Pokušaj pristupa 5 8 2018].
- [10] M. Berkelaar, D. Jeroen, E. Kjell i P. Notebaert, »LP Solve reference guide,« 24 9 2016. [Mrežno]. Available: <http://lpsolve.sourceforge.net/5.5/>. [Pokušaj pristupa 6 2018].
- [11] GearHost, »GearHost,« GearHost, 2018. [Mrežno]. Available: <https://www.gearhost.com>. [Pokušaj pristupa 6 2018].

- [12] Pivotal Software, »Spring boot,« Pivotal Software, 2018. [Mrežno]. Available: <https://spring.io/projects/spring-boot>. [Pokušaj pristupa 6 2018].
- [13] S. Scarioli, »The simplex method in Java,« 19 7 2018. [Mrežno]. Available: <http://www.ssclab.org/en/index.html>. [Pokušaj pristupa 6 2018].
- [14] Pivotal Software, »Spring Data,« Pivotal Software, 2018. [Mrežno]. Available: <https://spring.io/projects/spring-data>. [Pokušaj pristupa 6 2018].
- [15] FullCalendar LLC, »FullCalendar,« FullCalendar LLC, 2018. [Mrežno]. Available: <https://fullcalendar.io/>. [Pokušaj pristupa 7 2018].
- [16] Pivotal Software, »Spring Tool Suite,« Pivotal Software, 2018. [Mrežno]. Available: <https://spring.io/tools>. [Pokušaj pristupa 6 2018].
- [17] Pivotal Software, »Spring Security,« Pivotal Software, 2018. [Mrežno]. Available: <https://spring.io/projects/spring-security>. [Pokušaj pristupa 6 2018].
- [18] A. T. Ernst, H. Jiang, M. Krishnamoorthy i D. Sier, »Staff scheduling and rostering: A review of applications, methods and models,« *EUROPEAN JOURNAL OF OPERATIONAL RESEARCH*, pp. 2-4, 2004.
- [19] M. Marinović i M. Holenko Dlab, »Materijal iz kolegija Operacijska Istraživanja 1, dostupni u sustavu za učenje Canvas,« 2016. [Mrežno]. Available: [https://canvas.instructure.com/courses/1074004/files/44662269?module\\_item\\_id=10707111](https://canvas.instructure.com/courses/1074004/files/44662269?module_item_id=10707111). [Pokušaj pristupa 15 7 2018].

## 7. Popis slika

Slika 1 Primjer mrežnog toka .....	4
Slika 2 Stablo enumeracija .....	7
Slika 3 Stablo rješenja nakon prvog koraka ograđivanja .....	8
Slika 4 Stablo rješenja nakon drugog koraka .....	10
Slika 5 Stablo rješenja nakon trećeg koraka .....	11
Slika 6 Stablo rješenja nakon četvrtog koraka .....	11
Slika 7 Sučelje aplikacije ABC Roster .....	13
Slika 8 Sučelje aplikacije 247 Shift .....	14
Slika 9 Izgled sučelja programa LP Solve .....	21
Slika 10 DEV model baze podataka .....	23
Slika 11 Popis vlastitih kafića .....	25
Slika 12 Popis zaposlenika kafića .....	25
Slika 13 Početni formular za izradu rasporeda .....	26
Slika 14 Legenda radnika u kalendaru, broj mjesečnih sati i mjesečni trošak radnika .....	26
Slika 15 Generirani početni kalendar rasporeda .....	27
Slika 16 Popis generiranih rasporeda za željeni kafić .....	28
Slika 17 Mjesečni prikaz rasporeda .....	28
Slika 18 Tjedni prikaz rasporeda .....	29
Slika 19 Dnevni prikaz rasporeda .....	29
Slika 20 Odabir nove lozinke prilikom prve prijave .....	30

## 8. Popis tablica

Tablica 1 Potrebni radnici po danu .....	15
Tablica 2 Raspored konobara kroz 2. mjesec u 2017. godini .....	15
Tablica 3 Smjene za dane kada nije potrebno više od dva radnika .....	17
Tablica 4 Smjene za dane kada je potrebno tri ili više radnika .....	18