UNIVERZITET U BEOGRADU
MATEMATIČKI FAKULTET

Petar B. Zečević

OPTIMIZACIONI ALGORITAM ZA REŠAVANJE PROBLEMA RASPOREDA ČASOVA NA FAKULTETIMA

master rad

Beograd, 2023.

Mentor:

dr Aleksandar Kartelj, docent
Univerzitet u Beogradu, Matematički fakultet

Članovi komisije:

dr Filip Marić, redovni profesor
Univerzitet u Beogradu, Matematički fakultet

dr Miljan Knežević, docent
Univerzitet u Beogradu, Matematički fakultet

Datum odbrane: ____________________

Naslov master rada: Optimizacioni algoritam za rešavanje problema rasporeda časova na fakultetima

Rezime: U ovom radu je razmatran problem rasporeda časova na fakultetima (eng. UCTP – University course timetable problem) koji pedstavlja jednu varijantu problema rasporeda u obrazovanju. Ovaj problem je NP-težak. Ručno formiranje rasporeda časova može da bude vremenski zahtevno i da na kraju i dalje ne zadovoljava potrebe nastavnika i studenata. Za automatizovano formiranje rasporeda, koriste se mnogobrojne tehnike poput programiranja ograničenja, tabu pretrage i genetskog algoritma. Definišu se ograničenja koja moraju da budu ispunjena kako bi rešenje bilo zadovoljavajuće. U ovom radu je predložen metaheuristički pristup zasnovan na metodi promenljivih okolina (eng. VNS – Variable Neighborhood Search). Rad algoritma je demonstriran na istorijskim podacima podele nastave sa Katedre za Računarstvo, Matematičkog fakulteta univerziteta u Beogradu. Dodatno, implementirana je veb aplikacija koja prikazuje raspored časova dobijen iz algoritma i omogućava filtriranje predavanja na osnovu učionice, nastavnika i studentske grupe.

Ključne reči: optimizacija, metoda promenljivih okolina, problem rasporeda časova

Sadržaj

Glava 1

Uvod

Problem rasporeda časova na fakultetima pripada širem skupu problema raspoređivanja resursa. Problem raspoređivanja resursa se definiše kao problem alociranja resursa tokom vremena kako bi se izvršila kolekcija zadataka. Cilj je dodeliti skup entiteta ograničenom broju resursa tokom vremena, tako da se ispunjava skup predefinisanih zahteva [5]. Problem rasopreda je podskup problema raspoređivanja resursa gde se entitetima dodeljuju vremenski termini. Entiteti mogu biti nastavne jedinice (časovi), radnici, lekari, autobuske linije itd. Problem raspoređivanja resursa može da bude raspoređivanje zadataka koji treba da se izvrše jedanput, dok se problem rasporeda fokusira na pravljenje ponavljajućeg obrasca koji se primenjuje svakog dana, sedmice, meseca ili godine [29]. S druge strane, u literaturi se takođe može naći da se problem rasporeda i problem raspoređivanja resursa poistovećuju [17].

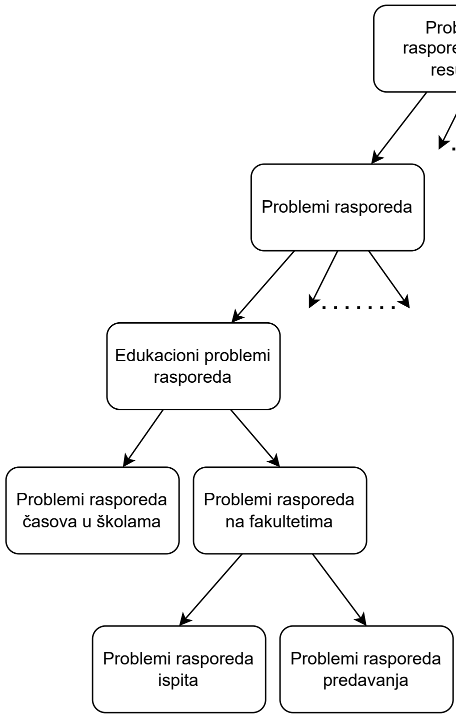
Problem rasporeda se može definisati u raznim domenima: medicina, proizvodnja, transport, obrazovanje itd. U domenu medicine je potrebno napraviti raspored dežurstava lekara, kao i termine pacijenata [2]. U domenu proizvodnje pojavljuje se problem rasporeda radnika po smenama [20]. U domenu transporta se pravi red vožnje za avione, autobuse ili vozove gde se vodi računa o potrebama putnika kao i o bezbednosti saobraćaja [3]. U obrazovanju se koriste rasporedi časova u školama i fakultetima, kao i rasporedi ispita [23].

Jedan od bitnih pojmova u ovom problemu, koji se pojavljuje u svim njegovim domenima, je pojam organičenja. Da bi formirani raspored bio upotrebljiv, on mora da ispoštuje neka predefinisana ograničenja (npr. u problemu reda vožnje autobusa, jedno od ograničenja je da dva autobusa ne mogu u isto vreme da polaze sa istog perona). Ograničenja se dele na *tvrda* (eng. hard constraint) i *meka* (eng. soft constraint). Tvrda ograničenja su ona koja moraju biti ispunjena kako bi raspored imao smisla (npr. dva pacijenta ne mogu imati isti termin kod jednog lekara). Meka ograničenja bi bilo poželjno ispuniti ali nisu neophodna za dopustivost rasporeda (npr. nastavnik u školi nema velike pauze između časova koje predaje).

Objedinjenu definiciju problema u opštem slučaju je teško precizirati, tako da se problem rasporeda definiše za svaki domen ponaosob. U nastavku sledi detaljniji opis problema rasporeda u obrazovanju, a zatim više informacija o konkretnom problemu rasporeda časova na fakultetima.

1.1 Problem rasporeda u obrazovanju

Problemi rasporeda u obrazovanju se dele u tri tipa problema: raspored časova u školama, raspored časova na fakultetima i raspored ispita na fakultetima. Na dijagramu 1.1 se može videti ova podela.



Slika 1.1: Dijagram podele problema rasporeda

Raspored časova u školama predstavlja raspoređivanje časova na nedeljnom nivou u osnovnim i srednjim šoklama. Đaci su podeljeni u odeljenja i svako odeljenje ima svoj raspored časova za svaki dan, a na časovima nastavnici drže odeđene predmete. U većini slučajeva, svaki predmet ima svoju predviđenu učionicu. Obično su odeljenja podeljena u dve smene i đaci moraju da imaju uzastopno nekoliko časova bez prekida svaki dan. Poželjno je da ni nastavnici nemaju velike pauze ali nije nužno. Nastavnik ne može da drži više predmeta u istom školskom času. Napravljeni raspored se koristi u jednoj školskoj godini, a svake nove godine se vrše manje izmene kako bi se prilagodili promenama nastavnog kadra. U radu [26] data je formalna definicija problema kao i razne metode rešavanja.

Raspored časova na fakultetima predstavlja formiranje rasporeda predavanja i vežbi u jednom semestru. Podela nastave definiše predavanja i vežbe gde su studentskim grupama dodeljeni nastavnici koji im predaju određene predmete. Svako predavanje, kao i vežbe, se smešta u neki vremenski termin i učionicu. Ni nastavnici ni studentske grupe ne smeju da imaju više predavanja ili vežbi u isto vreme. Poželjno je da nastavnici i studenti nemaju velike pauze u toku jednog dana, mada su u nekim slučajevima jasno definisane pauze koje moraju da se ispoštuju (npr. za doručak i ručak). Formalna definicja ovog problema data je u glavi 2. U daljem tekstu, pojam *predavanje* obuhvata i predavanja i vežbe. Takođe, pojam *nastavnik* se odnosi i na profesore i na asistente.

Postoje jasne razlike između rasporeda časova u školama i na fakultetima. U školama časovi moraju da počnu u isto vreme svaki dan, dok na fakultetima to nije slučaj. U osnovnim školama đaci ne smeju da imaju pauzu, dok je na fakultetima poželjno da studenti nemaju pauzu, ali nije obavezno. Predavanja na fakultetu se drže u blokovima od više časova, dok u školama se obično jedan predmet drži više puta po jedan školski čas u okviru jedne nedelje. Đaci moraju da imaju svaki dan časove, a studenti ne moraju.

Raspored ispita na fakultetima predstavlja raspodelu ispita za predmete u tekućoj školskoj godini. Ispiti se održavaju u ispitnim rokovima koji imaju jasno definisane datume početka i kraja. Ispiti predmeta iste studentske grupe ne smeju da imaju preklapanja, a svakom ispitu mora da bude dodeljen adekvatan broj asistenata koji dežuraju. Poželjno je da su ispiti iste studentske grupe dovoljno vremenski udaljeni jedni od drugih kako bi studenti imali vremena da ih spreme. U radu [1] data je formalna definicija ovog problema kao i jedna metodologija za rešavanje.

Problem rasporeda u školama i fakultetima je NP-težak. Ovo je dokazano u radu [8] pomoću redukcije na 3-SAT problem [11]. Dokaz je rađen nad primerom rasporeda sa minimalnim brojem ograničenja. U svim problemima rasporeda, postoji veći broj ograničenja i definisanih skupova entiteta (nastavnika, studenata, predmeta, učionica itd.).

1.1.1 Problemi rasporeda časova na fakultetima

Ručno sastavljanje rasporeda časova na fakultetima obično zahteva duži vremenski period. Treba uzeti u obzir da sva predavanja iz podele nastave treba smestiti u ograničen broj učionica i termina, ali tako da nijedan nastavnik i studenstka grupa nemaju predavanja koja se preklapaju. Čak i tad, nekim nastavnicima i studentima mogu da ne odgovaraju velike pauze, lokacije, doba dana u toku kojeg se održavanju časovi itd. Zbog toga, u nauci je dosta pažnje posvećeno automatizaciji ovog posla.

Prvi rad iz ove oblasti datira još iz 1963. godine [12], gde se ovaj problem naziva „Problem predmet-nastavnik". Džon Nigel Gont Britan u svojoj doktorskoj disertaciji iz 1974 [7] formalno definiše problem i analizira moguće pristupe rešavanju. Od tada je objavljeno mnoštvo novih radova na konferencijama, u časopisima i u obliku mnogobrojnih doktorskih disertacija. Na nekim univerzitetima se već uveliko koriste neka od ovih rešenja [23].

Većina početnih tehnika su bazirane na ljudskom pristupu rešavanja problema [24]. Odnosno, u pitanju su direktne heuristike koje koriste sukcesivno povećanje. Delimični raspored od nekoliko predavanja se proširuje, predavanje po predavanje, dok se sva data predavanja ne smeste u neki termin. Primarna ideja je da se prvo smesti predavanje koje ima najviše ograničenja, kako god su ta ograničenja definisana. Kasnije, istraživači su primenjivali neke opšte tehnike poput celobrojnog progamiranja, maksimalnog protoka u transportnim mrežama (eng. maximum flow in transport networks) kao i redukovanje na problem bojenja grafa. U skorije vreme, pristup je sve više baziran na programiranju ograničenja i metaheuristikama poput simuliranog kaljenja, tabu pretrage i genetskog algoritma.

U radu [24] je pokazan način da se problem rasporeda časova na fakultetima redukuje na problem bojenja grafa. Za svaki predmet c_i koji sluša studentska grupa g_j postoji čvor m_{ij} (gde je i indeks predmeta a j indeks studentske grupe). Za svaki g_j se pravi klika između čvorova m_{ij} (za $i = 1, ..., q$). Za svaki vremenski termin se definiše p novih čvorova. Oni su svi međusobno povezani. Ovim se obezbeđuje da svim vremenskim terminima bude dodeljena različita boja. Ukoliko predmet ne sme da se drži u nekom terminu, svi čvorovi koji se odnose na taj predmet su povezani sa tim čvorom termina. Ukoliko predmet mora da se drži u nekom terminu, onda svi čvorovi koji se odnose na taj predme nisu povezani sa čvorom tog termina. Ovaj pristup je takođe primenjen u radu [25].

Neki od radova koriste tehniku celobrojnog linearnog programiranja. U radu [27] se koristi Lagranžova relaksacija. U [10] se problem formuliše kao problem dodeljivanja i rešava se redukcijom na problem kvadratnog dodeljivanja (eng. QAP – quadratic assignment problem).

Tabu pretraga se takođe pokazala kao korisna tehnika (pogledati radove Herca [14] i [15]). U [14] je korišćena jednostavnija verzija postavke problema, dok se u [15] uzima u obzir da predavanja mogu biti različitih dužina.

Jedna od najpoznatijih metaheuristika, genetski algoritam, se može videti kao tehnika u nekim radovima iz ove oblasti [16] i [19]. Kvalitet rešenja predstavljenog hromozomom se ocenjuje pomoću fitnes funkcije u kojoj figurišu i tvrda i meka ograničenja. Genetski algoritam počinje sa nekom inicijlanom populacijom rasporeda (hromozoma – svaki hromozom kodira ceo raspored), zatim se vrši evaluacija i zaključuje se koji hromozomi postaju roditelji naredne generacije hromozoma. Nad tim hromozomima se vrše mutacije i ukrštanja za pravljenje nove generacije.

Metoda promenljivih okolina [21] (eng. VNS – Variable Neighborhood Search) je još jedan algoritam iz veštačke inteligencije koji je pronašao mesto u oblasti problema rasporeda. U radu [6] se ne koristi metoda promenljivih okolina u potpunosti, već samo njena faza lokalne pretrage. U [28] se koristi hibridni model metode promenljivih okolina i tabu pretrage. Algoritam predstavljen u ovom radu koristi metod promenljivih okolina u celosti kako bi rešio problem rasporeda časova na fakultetima.

U glavi 2 opisan je problem rasporeda i predstavljena formalna definicija problema. U glavi 3 objašnjen je Metod promenljivih okolina i prikazana upotreba algoritma na problem iz rada. U glavi 4 prolazi se kroz rad algoritma na primeru rasporeda časova na Matematičkom fakultetu.

Glava 2

Definicija problema

Problem rasporeda časova na fakultetima odlikuje pravljenje rasporeda za predavanja jednog fakulteta. Jedno predavanje drži jedan nastavnik koji predaje jedan predmet studentskoj grupi u određenom vremenskom terminu i određenoj učionici. Svaki predmet traje određen broj školskih časova i može da zahteva neku opremu (npr. računare). Svaka studentska grupa može da ima svoje podgrupe. Studenti jedne grupe slušaju određeno predavanje u jednom terminu. Svaki student jedne grupe pripada jednoj njenoj podgrupi i u okviru nje sluša određene vežbe u jednom terminu. Svaka učionica se nalazi u nekoj od fakultetskih lokacija i može da bude snadbevena određenom opremom. Između svake dve fakultetske lokacije je jasno definisana vremenska udaljenost.

Raspored časova je dopustiv ukoliko su sva planirana predavanja organizovana i poštuju neka zdravorazumska ograničenja kao što su: da nastavnik ne može da predaje dva predmenta u isto vreme, da studentima ne smeju da se preklapaju predavanja i da je učionica adekvatno opremljena za predmet koji se drži u njoj. Pored toga, postoje ograničenja koja bi bilo poželjno ispoštovati ali nije neophodno. Primeri toga su da nastavnici i studentske grupe imaju što manje pauza između predavanja, kao i da su ispunjene želje nastavnika u vezi sa terminima kada preferiraju da drže predavanje.

Veliki broj radova je formalno definisalo problem rasporeda časova, kao i skupove mekih i tvrdih ograničenja. Neke od definicija se mogu naći u sledećim radovima i doktorskim disertacijama [12], [4], [9], [18], [13] i [22]. Iako su sve definicije iz navedenih radova slične i definišu neka zajednička ograničenja, po dosta toga se razlikuju zbog različitih sistema na raznim univerzitetima i fakultetima. Neka ograničenja definisana u jednom radu za konkretan fakultet mogu da budu kontradiktorna sa ograničenjima iz drugog rada, zbog toga je u ovom radu je predstavljena formalna defincija problema koja uzima u obzir podelu nastave i ograničenja na Matematičkom fakultetu Univerziteta u Beogradu. Ova definicija može da se posmatra kao modifikacija i nadogradnja na prethodne formalne definicije. U nastavku glave sledi ova definicija kao i detaljniji opis korišćenih ograničenja.

2.1 Formalna definicija problema

Neka su date sledeće promenljive:

- $L = \{l_1, l_2, ..., l_n\}$ – skup nastavnika (eng. *lecturers*).
- $C = \{c_1, c_2, ..., c_m\}$ – skup predmeta (eng. *classes*).
- $G = \{g_1, g_2, ..., g_p\}$ – skup studentskih grupa i podgrupa (eng. *groups*).
- $T = \{t_{1,1}, t_{1,2}, ..., t_{a_1,a_2}\}$ – skup uzastopnih vremenskih intervala (eng. *timestamps*) raspoređenih po danima, gde je a_1 broj ukupnih dana, a a_2 broj vremenskih intervala u jednom danu. Dužina jednog vremenskog intervala je jednaka trajanju jednog školskog časa.
- $R = \{r_1, r_2, ..., r_q\}$ – skup učionica (eng. *rooms*).
- $V = \{v_1, v_2, ..., v_s\}$ – skup lokacija (eng. *venues*).
- $E = \{e_1, e_2, ..., e_t\}$ – skup različite opreme (eng. *equipment*), potrebne za pojedina predavanja.

Neka je za svako $c \in C$ definisana uređena dvojka $(e, s) \in (E \times \{1, 2, 3\})$ koja predstavlja neophodnu opremu i broj časova za jedan predmet. Neka je sa $\Gamma \subseteq G \times G$ definisana relacija između studentske grupe i podgrupe u kojoj za svaku uređenu dvojku $(g_1, g_2) \in \Gamma$ važi da je grupa g_1 podgrupa grupe g_2, odnosno da je grupa g_2 nadgrupa grupe g_1. Neka je svakoj učionici (r) dodeljena uređena dvojka $(v, e) \in (V \times E)$ koja predstavlja lokaciju i opremljenost učionice. Neka $d : V \times V \to \mathbb{N}$ preslikava svake dve lokacije u vreme potrebno da se pređe sa jedne lokacije na drugu. Pritom je vreme izraženo celobrojno kao broj školskih časova zaokružen na gore.

Uređena petorka $(l, c, g, t, r) \in x \subseteq L \times C \times G \times T \times R$ predstavlja jedno predavanje u rasporedu časova, odnosno znači da predavač l predaje predmet c studentskoj grupi

g sa početkom u vremenskom intervalu t u učionici r. Neka je X partitivni skup od $L \times C \times G \times T \times R$. Tada je $x \in X$ jedan kandidat rasporeda časova, ne nužno dopustiv.

Skup iskaza P je skup svih mogućih ograničenja nad rasporedom časova $x \in X$. Ograničenje $p \in P$ nad rasporedom časova x može biti narušeno ili nenarušeno. Ograničenja mogu biti, kao što je ranije rečeno, tvrda i meka.

Definiicija 1 *Raspored časova $x \in X$ je dopustiv ukoliko nijedno tvrdo ograničenje $p \in P$ nije narušeno nad njim.*

Potrebno je otkriti rešenje koje je dopustivo i pritom je narušenost mekih ograničenja minimalna.

2.2 Ograničenja

Definišemo četiri tipa ograničenja koja se javljaju u problemu rasporeda časova. To su: eksplicitna tvrda ograničenja, implicitna tvrda ograničenja, eksplicitna meka ograničenja i implicitna meka ograničenja.

2.2.1 Eksplicitna tvrda ograničenja

Neka su dati ulazni podaci u obliku skupa uređenih trojki $\Upsilon \subseteq L \times C \times G$ koji predstavljaju podelu nastave. Uređena trojka $(l, c, g) \in \Upsilon$ predstavlja jedno predavanje predmeta c koje drži nastavnik l studentskoj grupi g. Ova uređena trojka takođe označava jedno eksplicitno tvrdo ograničenje obeleženo sa $p_{eh}(\hat{l}, \hat{c}, \hat{g}) \in P$. Ograničenje $p_{eh}(\hat{l}, \hat{c}, \hat{g})$ je narušeno ukoliko u rasporedu časova ne postoji $(\hat{l}, \hat{c}, \hat{g}, t, r) \in x$ za konkretne $\hat{l}$, $\hat{c}$ i $\hat{g}$.

2.2.2 Implicitna tvrda ograničenja

Sva implicitna tvrda ograničenja $p_{ih}(x) \in P$ se odnose za neku konkretnu petorku $(\hat{l}, \hat{c}, \hat{g}, \hat{t}, \hat{r})$ iz celokupnog rasporeda časova x. Definisana su sledećim skupom pravila (predstavljana su bez matematičke formulacije):

- Nastavnik ne može da drži dva ili više predavanja u isto vreme.
- Studentska grupa ne može da pohađa dva predavanja u isto vreme.

- Ukoliko je studentska grupa A podgrupa grupe B, onda predavanja za grupe A i B ne mogu da budu u isto vreme.
- U određenoj učionici može da se odvija samo jedno predavanje u jednom školskom času.
- Ukoliko nastavnik ili studentska grupa ima predavanja na različitim lokacijama u jednom danu, mora da postoji dovoljno veliki vremenski interval između tih predavanja.
- Svako predavanje mora uzastopno da traje predviđeni broj vremenskih intervala.
- Ukoliko predmet zahteva specifičnu opremu, učionica u kojoj se održava mora biti adekvatno opremljena.

Ograničenje $p_{ih}(x)$ je narušeno ukoliko postoji petorka $(l, c, g, t, r) \in x$ koja ne ispunjava neko od pravila iz skupa pravila.

2.2.3 Eksplicitna meka ograničenja

Eksplicitna meka ograničenja je poželjno ispuniti, ali nije nužno. Uglavnom se odnose na preferencije i pogodnosti za nastavnike ili studente. Neka su dati sledeći ulazni podaci:

- Skup $L_T \subseteq L \times T \times \{true, false\}$ koji predstavlja vremenske preferencije nastavnika. Uređena trojka $(l, t, b) \in L_T$ predstavlja jednu preferenciju nastavnika l vezanu za vremenski interval t. Ukoliko je $b = false$ onda nastavnik l ne želi da drži predavanja u vremenskom intervalu t. Ukoliko je $b = true$ nastavnik l želi da drži predavanja u vremenskom intervalu t. Ukoliko u skupu L_T ne postoji $(\hat{l}, \hat{t}, b)$ za konkretne vrednosti $\hat{l}$ i $\hat{t}$, onda je nastavniku $\hat{l}$ svejedno da li će držati predavanja u terminu $\hat{t}$.
- Skup $L_V \subseteq L \times V$ koji predstavlja preferencije nastavnika prema određenim lokacijama. Uređena dvojka $(l, v) \in L_V$ predstavlja jednu instancu preferencije nastavnika l koji želi da drži sva svoja predavanja na lokaciji v.

Ove preferencije takođe označavaju ograničenja $p_{es}(l, t, b)$ i $p_{es}(l, v)$, koja su narušena ukoliko u rasporedu časova postoji $(l, c, g, t, r) \in x$ koji ne ispunjava neku od gorenavedenih preferencija.

2.2.4 Implicitna meka ograničenja

Sva implicitna meka ograničenja su predstavljena kroz skup iskaza koji mogu biti tačni za raspored časova x. Implicitno meko ograničenje $p_{is}(x)$ je narušeno ukoliko postoji studentska grupa g ili nastavnik l u rasporedu časova za koje je tačno neko od sledećih iskaza:

- Studentska grupa ima pauze između predavanja u toku jednog dana (uzimaju se u obzir i studentske podgrupe).
- Studentska grupa menja lokaciju predavanja u toku jednog dana (uzimaju se u obzir i studentske podgrupe).
- Nastavnik ima pauze između predavanja u toku jednog dana.
- Nastavnik menja lokaciju predavanja u toku jednog dana.

Za neku studentsku grupu ili nastavnika jedan od iskaza može biti tačan više puta (npr. nastavnik menja lokaciju dva ili više puta u toku jednog dana, nastavnik menja lokaciju više dana). Tada je više puta narušeno isto ograničenje $p_{is}(x)$. Idealno, $p_{is}(x)$ treba da bude 0, odnosno nijedan iskaz ne treba da bude tačan. Drugim rečima, potrebno je minimizovati broj tačnih iskaza i time minimizovati broj narušenih ograničenja $p_{is}(x)$

Glava 3

Predloženi algoritam za raspored časova

Ovaj rad predlaže algoritam za pravljenje rasporeda časova na Matematičkom fakultetu pomoću VNS metodologije. U ovoj glavi su prvo opisani ulazni podaci i kako se oni pripremaju za algoritam, zatim, prikazan je VNS algoritam kao i njegova primena na konkretnom problemu iz rada. Kompletan kôd se može naći na [30].

3.1 Ulazni podaci i njihova priprema

Ulazni podaci neophodni za rad algoritma su:

- Podela nastave svih katedri na fakultetu.
- Informacije o studentskim grupama.
- Spisak svih dostupnih lokacija.
- Spisak svih dostupnih učionica i raspoložive opreme u njima.
- Preferencije nastavnika.

Podela nastave sadrži informacije o predviđenim predavanjima za tekuću školsku godinu. Za svako predavanje je definisano ime predmeta, nastavnik koji ga drži, grupa studenata koji ga slušaju, fond časova, kao i semestar u kom se drži. Fond časova pruža informaciju koliko treba da traju predavanja za jedan predmet. Predavanja imaju onoliko termina koliko imaju grupa tj. dve ili više grupa ne mogu da imaju isti termin jednog predavanja.

Informacije o studentskim grupama predstavljaju podelu studenata u disjunktne grupe. Grupe dalje mogu da se dele na disjunktne podgrupe. Podgrupe se tretiraju kao grupe, s tim što njihova predavanja ne smeju da imaju preklapanja sa predavanjima njihove nadgrupe.

Spisak svih dostupnih lokacija predstavlja podatke o lokacijama gde predavanja mogu da se održe. Svaka lokacija ima svoje ime i definisanu listu udaljenosti od ostalih lokacija. Te udaljenosti su izražene u vremenskim intervalima.

Spisak svih dostupnih učionica predstavlja informaciju o prostorijama u kojima je moguće držati predavanje. Učionica ima definisanu lokaciju, kao i informaciju da li je opremljena dodatnom opremom, poput računara.

Preferencije nastavnika predstavljaju spisak željenih i neželjenih termina, kao i željenih lokacija nastavnika. Ove podatke nastavnici popunjavaju ponaosob i algoritam se trudi da ih ispuni.

Pomoću ovih podataka, pripremaju se podaci za rad algoritma. Oni su predstavljeni u vidu sledećih tabela:

- Nastavnici;
- Grupe;
- Predmeti;
- Nastavni program;

Tabela nastavnici predstavlja metapodatke nastavnika. Ona ima sledeće kolone:

- Ime – ime nastavnika.
- Poželjno vreme – željena satnica nastavnika.
- Nepoželjno vreme – satnica kada nastavnik ne želi da drži nastavu.
- Poželjna lokacija – lista željenih lokacija nastavnika.

Tabela Grupe predstavlja metapodatke grupa. Ona ima sledeće kolone:

- Ime – ime grupe.
- Roditelj – ime roditelj grupe (ukoliko postoji).
- Deca – lista dece grupe (ukoliko barem jedno dete postoji).

Tabela Predmeti predstavlja metapodatke predmeta. Ona ima sledeće kolone:

- Ime – ime predmeta. Sadrži sufiks „vežbe" ukoliko su u pitanju vežbe.
- Oprema – oprema potrebna za održavanje predmeta (na primer računari).
- Broj časova – nedeljni broj sati nepohodan za održavanje predmeta. Na Matematičkom fakultetu svi časovi istog predmeta se održavaju vezano.

Tabela Nastavni program je najbitnija tabela za rad algoritma. Ona sadrži relacije između tabela Nastavnici, Grupe i Predmeti. Odnosno, jedan red ove tabele predstavlja predavanje konkretnog predmeta koji sluša konkretna grupa i koji drži konkretan nastavnik. Umesto grupe, sadrži listu grupa koje slušaju to predavanje.

Navedene tabele, zajedno sa Spiskom svih dostupnih lokacija i učionica, predstavljaju neophodne podatke za rad algoritma. U tabeli 3.1 su prikazani primeri podataka koji se mogu naći u ovim tabelama.

Na Matematičkom fakultetu, predavanja se drže radnim danima od ponedeljka do petka. Svakog radnog dana predavanja se mogu održavati od 8h do 21h, što znači da vremenskih intervala, u trajanju od jednog sata, za jedan dan ima 13, a ukupno 65 u toku radne nedelje.

Tabela 3.1: Pripremljeni podaci za algoritam

Tabela Nastavnici

Ime nastavnika	Preferirano vreme	Nepreferirano vreme	Preferirana lokacija
Petar Petrović	Ponedeljak 9:00 - 18:00	Četvrtak 12:00 - 20:00	Vračar

Tabela studentske grupe

Ime grupe	Roditelj	Deca
m1	Nema	m1a, m1b
m1a	m1	Nema
m1b	m1	Nema

Tabela predmeti

Ime predmeta	Oprema	Broj časova
Matematika 1	Nema	3
Matematika 1 - vežbe	računari	2

Tabela Nastavni program

Ime nastavnika	Ime predmeta	Ime grupe
Petar Petrović	Matematika 1	m1

Tabela Učionice

Ime učionice	Lokacija	Oprema
101	Vračar	računari

Tabela Lokacije

Ime lokacije	Lista udaljenosti
Vračar	1h, 2h

3.2 Predloženi VNS metod

Algoritam predložen u radu koristi gore navedene podatke i kao izlaz proizvodi tabelu raspored časova, koja svakom predavanju dodeljuje uređeni par sačinjen od vremenskog intervala i učionice. Ovo se postiže upotrebom VNS metoda koji će biti opisan u daljem tekstu.

3.2.1 VNS metod

Metoda promenljivih okolina (eng. VNS – Variable Neighbourhood Search) je metaheuristički algoritam za rešavanje kombinatornih i globalnih optimizacionih problema. Prvi put je izložena 1997. u radu [21]. Metoda je razvijena za rešavanje problema diskretne optimizacije i zbog svoje jednostavnosti, u pogledu implementacije i uštede na memorijskim i vremenskim resursima, je pogodna za široku upotrebu. Glavna ideja algoritma je sistematična promena okoline u dve faze. Prva faza je faza spuštanja (eng. descent) kako bi se našao lokalni optimum. Druga faza je faza razmrdavanja (eng. shaking) kako bi se izašlo iz odgovarajuće doline, odnosno lokalnog optimuma. Algoritam prolazi ove dve faze naizmenično kako bi težio ka globalnom optimumu.

Neka je x tačka u prostoru pretrage, tj. neko rešenje posmatranog optimizacionog problema. Sa $N_k(x)$ je označena k-ta okolina tačke x, odnosno skup svih rešenja koja su do na parametar k *blizu* rešenju x. U praksi se opseg vrednosti k bira iz $\{k_{min}, ..., k_{max}\}$. Većina heuristika lokalnih pretraživanja koristi samo jednu strukturu okoline, to jest važi da je $k_{max} = 1$. Često važi da su strukture okolina ugnježdene tj. $N_1 \subset N_2 \subset N_3 \subset ... \subset N_{k_{max}}$. Tačka $x \in X$ je lokalni minimum za okolinu N_k ukoliko ne postoji $x^{'} \in N_k(x)$ takvo da važi $f(x^{'}) < f(x)$. Nakon nalaženja lokalnog optimuma, algoritam nastavlja pretragu na neke druge načine, obično razmrdavanjem. VNS algoritam podrazumeva da:

- Lokalni minimum jedne okoline tačke x nije nužno isti kao lokalni minimum neke druge okoline tačke x.
- Globalni minimum okoline tačke x je najmanji lokalni minimum svih mogućih okolina tačke x.
- U većini slučajeva, lokalni minimumi više struktura okoline su relativno blizu jedan drugom.

Poslednja stavka ukazuje na to da lokalni optimum često daje neku informaciju o globalnom optimumu. Sledi algoritam za promenu strukture okoline [1]:

Algorithm 1 Promena okoline

function PromenaOkoline($x, x^{'}, k$) :
  if $f(x^{'}) < f(x)$ **then**
    $x \leftarrow x^{'}$ // Napravi prelaz u novo rešenje
    $k \leftarrow k_{min}$ // Vrati se na inicijalnu okolinu
  else
    $k \leftarrow k + 1$ // Idi na sledeću okolinu
  end if
  return x, k
end function

Funkcija *PromenaOkoline()* poredi trenutnu vrednost $f(x)$ sa novom vrednošću $f(x^{'})$ dobijenom iz k-te okoline. Ukoliko je nova vrednost manja, ažurira se trenutna vrednost i k se postavlja na prvu okolinu. U suprotnom, uzima se sledeća okolina.

Osnovni VNS algoritam kombinuje sistematske i stohastičke promene okoline. Sistematski deo je ostvaren pomoću lokalne pretrage. Sastoji se iz tri koraka:

1. Izbor početnog x.

2. Pronalaženje smera silaska iz x unutar okoline $N_{\hat{k}}(x)$ – gde je $\hat{k}$ konstanta, i obično iznosi jedan ili dva.

3. Pomeranje u minimalno $f(x)$ unutar okoline $N_{\hat{k}}(x)$ u tom smeru.

Ukoliko u iteraciji nije nađeno bolje rešenje, pretraga prestaje, u suprotnom sledi naredna iteracija. Ovo je prikazano u algoritmu [2], gde je pretpostavljeno da je početno x dato. Povratna vrednost je lokalni minimum $x^{'}$.

Pošto najstrmiji spust može da bude vremenski zahtevan, alternativa je heuristika *prvog spusta* (eng. first descent). Elementi $x^{''} \in N_{\hat{k}}(x)$ su slučajno sortirani i pomeraj je postignut čim se naiđe na prvo spuštanje. Ovo je prikazano u algoritmu [3].

Stohastički deo algoritma, pod nazivom razmrdavanje (eng. shaking), je ostvaren pomoću nasumičnog izbora tačke iz k-te okoline. Ovo je prikazano u algoritmu [4].

Rezultat stohastičkog dela algoritma se uzima kao inicijalno x za sistematski deo algoritma. Zatim se vrši promena strukture okoline. Ovaj postupak se ponavlja iterativno dok nije ispunjen kriterijum zaustavljanja. U narednom primeru kriterijum

Algorithm 2 Najbolje poboljšanje (najstrmiji spust – eng. steepest descent)

function NAJBOLJEPOBOLJŠANJE(x) :
  repeat
    $x^{'} \leftarrow x$
    for $x^{''}$ in $N_{\hat{k}}(x)$ **do**
      if $(f(x^{''}) \leq f(x))$ **then**
        $x \leftarrow x^{''}$
      end if
    end for
  until $(f(x) \geq f(x^{'}))$
  return $x^{'}$
end function

Algorithm 3 Prvo pobojšanje (prvi spust – eng. first descent)

function PRVOPOBOLJŠANJE(x) :
  repeat
    $x^{'} \leftarrow x$
    $nasumi\check{c}noSortirajElementeOkoline(N_{\hat{k}}(x))$
    for $x^{''}$ in $N_{\hat{k}}(x)$ **do**
      if $(f(x^{''}) \leq f(x))$ **then**
        $x \leftarrow x^{''}$
        break
      end if
    end for
  until $(f(x) \geq f(x^{'}))$
  return $x^{'}$
end function

Algorithm 4 Razmrdavanje

function SHAKE(x, k) :
  $x^{'} \leftarrow izaberiNasumi\check{c}niElementIzOkoline(N_k(x))$
  return $x^{'}$
end function

zaustavljanja je procesorsko vreme i koristi se najbolje poboljšanje za deterministički deo [5].

Algorithm 5 Osnovni VNS algoritam

function MetodaPromeneOkoline(x, k_{max}, t_{max}) :
  $t \leftarrow 0$
  while $t < t_{max}$ **do**
    $k \leftarrow k_{min}$
    repeat
      $x^{'} \leftarrow Ramrdavanje(x, k)$
      $x^{''} \leftarrow NajboljePoboljšanje(x^{'})$
      $x, k \leftarrow PromenaOkoline(x, x^{''}, k)$
    until $k = k_{max}$
    $t \leftarrow CpuTime()$
  end while
  return x
end function

3.2.2 Primena VNS na pravljenje rasporeda časova

U ovoj sekciji je predstavljen opis implementacije VNS algoritma za pravljenje rasporeda časova na Matematičkom fakultetu. Skup rešenja X je skup svih rasporeda časova koje je moguće napraviti pomoću datog nastavnog programa. Rešenje x je jedan raspored časova iz tog skupa. Rešenje je predstavljeno kodiranjem datim u narednoj sekciji. Fitnes funkcija $f(x)$ predstavlja sumu narušenih ograničenja u datom rasporedu časova x. Detaljniji opis sledi u sekciji Fitnes funkcija.

Razmdravanje korišćeno u algoritmu je prikazano u sekciji Razmdravanje. Lokalna pretraga koristi najstrmiji spust i detaljnije je prikazana u sekciji Lokalna pretraga. Poslednja sekcija sadrži prikaz celokupnog algoritma.

Kodiranje

Jedno predavanje na fakultetu je definisano sa nastavnikom koji ga drži, predmetom koji se predaje i studentskom grupom koja ga sluša, $(l, c, g) \in T_C \subseteq L \times C \times G$. Kôd je, dakle, predstavljen skupom svih predavanja. Svakom predavanju se dodeljuje promenljiva $(t, r) \in T_T \subseteq T \times R$, gde t predstavlja vremenski interval u kom predavanje počinje, a r učionicu u kojoj se održava. Ovim je obezbeđeno da je za svako predavanje jedinstveno definisano vreme i mesto održavanja, što je dovoljno da raspored časova, $x \in X$, bude u potpunosti definisan.

Alternativno, kodiranje je moguće definisati i kao matricu vremenskih intervala i učionica, dimenzije $|T| \times |R|$. Svakom elementu te matrice se dodeljuje promenljiva $(l, c, g) \in T_C \subseteq L \times C \times G$. Veliki broj elemenata ostaje prazan, zato što su u većini vremenskih intervala neke učionice slobodne. U ovom radu je korišćeno prvo kodiranje.

Inicijalizacija

Za inicijalni kod, potrebno je svakom elementu $(l, c, g) \in T_C = L \times C \times G$ dodeliti nasumično $(t, r) \in T_T = T \times R$, bez obzira na to da li je time narušeno neko ograničenje.

Fitnes funkcija

U ovoj sekciji, definisana je fitnes funkcija koju je potrebno minimizovati.

Uzmimo dva rasporeda časova, x_1 i x_2. Neka su $P_h(x)$, $P_s(x)$ redom broj narušenih tvrdih i mekih ograničenja u rasporedu časova x. Sva meka ograničenja iz $P_s(x)$

se podjednako vrednuju. Neka je $f(x)$ vrednost fitnes funkcije u x. Trebalo bi da bude zadovoljeno sledeće:

1. $P_h(x_1) = P_h(x_2), P_s(x_1) < P_s(x_2) \Rightarrow f(x_1) < f(x_2)$

2. $P_h(x_1) < P_h(x_2) \Rightarrow f(x_1) < f(x_2)$

Drugim rečima, funkcija pretrage predstavlja broj narušenih tvrdih i mekih ograničenja, s tim što se strožije gleda na tvrda ograničenja. Ukoliko jedan raspored časova ima manje narušenih tvrdih ograničenja od drugog, onda je on bolji. Ukoliko oba rasporeda časova imaju jednak broj narušenih tvrdih ograničenja. onda je bolji onaj koji ima manji broj narušenih mekih ogrničenja.

Neka je P_{sMax} najveći mogući broj narušenih mekih ograničenja u jednom rasporedu. Onda se može definisati fitnes funkcija kao:

$$f(x) = P_h(x) + \frac{P_s(x)}{P_{sMax} + 1}$$

Dopustivo rešenje rasporeda časova ne sme da ima nijedno narušeno tvrdo ograničenje. Ovako definisana fitnes funkcija će biti manja od vrednosti jedan *akko* rešenje ne narušava nijedno tvrdo ograničenje. Dakle, za kranje rešenje rasporeda časova x mora da važi $f(x) < 1$.

Ovakvom formulom je obezbeđeno da je broj tvrdih ograničenja uvek prioritet, čak i kad su narušena sva moguća meka ograničenja. Zatim, ukoliko jedan raspored časova nema nijedno narušeno tvrdo ograničenje, a ima puno narušenih mekih ograničenja, biće bolji od rasporeda koji ima narušeno samo jedno tvrdo ograničenje i nijedno meko ograničenje. Ovo ide u prilog činjenici da raspored časova sa narušenim bar jednim tvrdim ograničenjem nije dopustiv.

Ovakvo definisanom fitnes funkcijom se obezbeđuje da male popravke u rasporedu dovode do malih popravki u fitnes funkciji. Time će se sekvencijalnim popravkama rasporeda težiti ka optimumu fitnes funkcije.

Razmrdavanje

Minimalna perturbacija je najmanja moguća promena rasporeda časova. Definišemo dve vrste minimalne perturbacije:

1. 1-zamena - za dve uređene petorke $(l_1, c_1, g_1, t_1, r_1)$ i $(l_2, c_2, g_2, t_2, r_2)$ formiraju se nove petorke $(l_1, c_1, g_1, t_2, r_2)$ i $(l_2, c_2, g_2, t_1, r_1)$.

2. 1-promena - za jednu uređenu petorku (l, c, g, t, r) formira se izmenjena petorka (l, c, g, t', r') gde su $t' \in T$ i $r' \in R$ nasumične vrednosti.

Razmdravanje je postignuto tako što se vrši k minimalnih perturbacija. Broj k je ulazni parametar u funkciju i zavisi od trenutne okoline.

Algorithm 6 Razmrdavanje

```
function RAZMRDAVANJE(x, k_max) :
    k ← 0
    while k < k_max do
        x ← minPerturbation(x)
        k ← k + 1
        return x
```

U radu se koriste samo 1-promena minimalne perturbacije. Razlog za to je što 1-zamena minimalna perturbacija nikad neće uzeti u obzir neke nove t' i r' koji ne figurišu ni u jednoj uređenoj petorci (l, c, g, t, r) iz trenutnog rasporeda.

Lokalna pretraga

U fazi lokalne pretrage, algoritam pokušava da u svakom koraku nađe bolje rešenje u određenoj okolini. Skup svih rasporeda časova koji se dobiju primenom jedne minimalne perturbacije nad x se zove okolina od x. U jednom koraku lokalne pretrage, proverava se da li može da se nađe bolji raspored u okolini od x. Ukoliko postoji takav x' raspored, on sada postaje novi x. Ovaj korak se ponavlja sistematično dok god postoji bolji raspored u okolini. Novi raspored iz okoline x može da se izabere na dva načina:

1. Prvo poboljšanje – uzima se prvi bolji raspored od x iz njegove okoline.

2. Najbolje poboljšanje – uzima se najbolji raspored u okolini x.

U radu se koristi Najbolje poboljšanje s obzirom da nije mnogo vremenski zahtevnije a obično daje bolji rezultat.

Skup svih minimalnih perturbacija nad x koje se mogu dobiti su definisane na sledeći način:

1. Za svaku uređenu trojku (l, c, g) promena njegovog t na bilo koji $t_1 \in T$ definiše jednu minimalnu perturbaciju.

2. Za svaku uređenu trojku (l, c, g) promena njegovog r na bilo koji $r_1 \in R$ definiše jednu minimalnu perturbaciju.

Drugim rečima, algoritam svako predavanje pokušava da smesti u bilo koji drugi termin ili u bilo koju drugu učionicu. U koraku kada je x najbolji raspored u svojoj okolini, faza lokalne pretrage se prekida.

Algorithm 7 Lokalna pretraga

```
function LokalnaPretraga(x) :
    while true do
        x' ← NaiNajboljiUOkolini(x)
        if f(x') < f(x) then
            x ← x'
        else
            return x
```

Funkcija $NaiNajboljiUOkolini$ vraća x sa najboljom vrednošću funkcije pretrage iz okoline, kao što je već obrazloženo u radu.

Celokupan algoritam

Neka k predstavlja trenutni parametar koji se prosleđuje funkciji za razmdravanje. Neka je k_{min} inicijalno, a k_{max} najveće dozvoljeno k. Neka je t_t početna podela nastave. U jednoj iteraciji algoritma se vrši prvo razmrdavanje sa datim parametrom k, a zatim lokalna pretraga nad rasporedom časova x. Ukoliko smo našli bolje rešenje, uzimamo to rešenje i vraćamo parametar k na inicijalnu vrednost. Zatim počinjemo sledeću iteraciju sa novim rasporedom časova. Ukoliko nismo našli bolje rešenje, povećavamo k za 1.

Glavna ideja je da ukoliko nije pronađeno bolje rešenje znači da je algoritam došao do nekog lokalnog minimuma. Da bi algoritam izašao iz njega, potrebno je da se odradi razmdravanje. Što je veće k dato, okolina je šira pa će razmdravanje potencijalno dati rešenje dosta različito od trenutnog. Postepeno se povećava k kako bi se algoritam pomerio najmanje koliko je potrebno da bi izašao iz lokalnog optimuma. Razlog za ovo je što je pretpostavka da su lokalni optimumi blizu globalnog optimuma. Parametar k_{max} služi za kriterijum zaustavljanja. Ukoliko se dovoljno povećalo k, a i dalje se nailazi na isto rešenje, to rešenje je sasvim moguće optimalno. U radu je početno k, odnosno k_{min}, postavljeno na vrednost dva.

Algorithm 8 VNS

function $\textsc{Vns}(t_t, k_{max})$:
    $k \leftarrow k_{min}$
    $x \leftarrow inicijalizujRaspored(t_t)$
    while $k < k_{max}$ **do**
        $x^{'} \leftarrow Razmrdavanje(x, k)$
        $x^{'} \leftarrow LokalnaPretraga(x^{'})$
        if $f(x^{'}) < f(x)$ **then**
            $x \leftarrow x^{'}$
            $k \leftarrow k_{min}$
        else
            $k \leftarrow k + 1$
            return x

Rezultat dobijen agoritmom VNS je predstavljen u obliku $x \in X$ gde je jedan element $(l, c, g, t, r) \in x$ uređena petorka koja predstavlje jedno predavanje u rasporedu časova. U tom predavanju predavač l predaje predmet c studenstkoj grupi g sa početkom u vremenskom intervalu t u učionici r.

Ukoliko je $f(x) \geq 1$ ovaj raspored nije dopustiv i ne može se uzeti u obzir, zato što to znači da je narušeno neko tvrdo ograničenje. Što je $f(x)$ manji, to je raspored bolji. Ukoliko je $f(x) = 0$ nađen je raspored koji ne narušava nijedno dato ograničenje.

Glava 4

Demonstracija

U ovoj glavi je prikazana demonstracija rada algoritma na realnom primeru. Uzeti su istorijski ulazni podaci koju su se koristili za kreiranje rasporeda časova na Matematičkom fakultetu, na Katedri za računarstvo i informatiku, u letnjem semestru 2023. godine.

4.1 Ulazni podaci

Ulazni podatak korišćen za kreiranje rasporeda časova je podela nastave na Katedri za računarstvo. Na slici 4.1 prikazan je deo tabele podela nastave za predmet „Programiranje 1" na prvoj godini smera Matematika.

	Akreditacija	Šifra	Naziv	Semestar	Fond	Grupe Pr	Grupe V	Tok	Predavanja	Tok vežbe	Vežbe
0	Osnovne akademske studije 2022	RM01	Programiranje 1	Jesenji semestar	2 + 3	4.0	8.0	1o1	Nina Radojičić Matić	1o1a	Milan Kocić
1					2 + 3	4.0	8.0	1o2	Nina Radojičić Matić	1o1b	Milan Kocić
2					2 + 3	4.0	8.0	1o3	Jovana Kovačević	1o2a	Ognjen Milinković
3					2 + 3	4.0	8.0	1o4	Jovana Kovačević	1o2b	Ognjen Milinković
4					2 + 3	4.0	8.0			1o3a	Milan Kocić
5					2 + 3	4.0	8.0			1o3b	Andrija Urošević
6					2 + 3	4.0	8.0			1o4a	Nikola Katić
7					2 + 3	4.0	8.0			1o4b	Nikola Katić

Slika 4.1: Podela nastave za predmet Programiranje 1

Korišćene su sledeće kolone:

1. Akreditacija – godina i tip studija (osnovne, master ili doktorske);
2. Šifra – interna šifra predmeta;
3. Naziv – naziv predmeta;

4. Semestar – jesenji ili prolećni semestar;
5. Fond – broj predavanja i vežbi;
6. Grupe Pr – broj predviđenih grupa za predavanje;
7. Grupe V – broj predviđenih grupa za vežbe;
8. Tok – studentska grupa koja sluša predavanje;
9. Predavanja – ime profesora koji drži predavanje;
10. Tok vežbe – studentska podgrupa koja sluša vežbe;
11. Vežbe – ime asistenta koji drži vežbe;

Pored podele nastave, dat je i ulazni podatak o relacijama između studentskih grupa 4.2. Ti podaci pokazuju ime roditelja studentske grupe, ukoliko postoji.

	Ime	Roditelj
0	1o1	None
1	1o1a	1o1
2	1o1b	1o1
3	1o2	None
4	1o2a	1o2
5	1o2b	1o2
6	1o3	None
7	1o3a	1o3
8	1o3b	1o3
9	1o4	None
10	1o4a	1o4
11	1o4b	1o4

Slika 4.2: Relacije između studentskih grupa na prvoj godini opšteg smera

Na osnovu ovih podataka jasno je da se Programiranje 1 sluša u 4 grupe predavanja, od kojih je svaka podeljena u po dve podgrupe za vežbe. Fond časova je dva za predavanja i tri za vežbe.

Neophodni podatak je takođe spisak i svih dostpunih učionica. Na slici 4.3 prikazane su neke od učionica. Može se videti na kojoj se lokaciji učionice nalaze kao i njihova opremljenost.

Ove podatke dalje obrađujemo kako bismo ih koristili u samom algoritmu.

	Ime	Lokacija	Oprema
0	406	Studentski trg	
1	704	Studentski trg	racunari
2	706	Studentski trg	
3	718	Studentski trg	racunari
11	N102	Sv. Nikole	
12	N103	Sv. Nikole	
13	N151	Sv. Nikole	
14	N152	Sv. Nikole	
23	Jag1	Jagic	racunari
24	Jag2	Jagic	racunari
25	Jag3	Jagic	
26	Jag4	Jagic	

Slika 4.3: Prikaz dela spiska učionica

4.2 Priprema podataka

Ulazni podaci se prečišćavaju i prave se četiri nove tabele: Nastavnici, Predmeti, Grupe i Podela nastave.

Tabela nastavnici predstavlja izdvojene podatke o svim nastavnicima. Kolone „Željeno vreme” i „Neželjeno vreme” predstavljaju preferncije termina koji nastavnicima odgovaraju ili ne odgovaraju. Uređeni brojevi sekvencijalno predstavljaju sate u okviru jedne radne sedmice, uzimajući u obzir samo interval 8:00 - 21:00 svakog radnog dana. Na primer broj 0 predstavlja termin 8:00 - 9:00 ponedeljkom, broj 12 je termin 20:00 - 21:00 ponedeljkom, dok je broj 13 termin 8:00 - 9:00 utorkom. Poslednji termin u radnoj nedelji, 20:00 - 21:00 petkom, predstavljen je brojem 64. Kolona „Željene lokacije” predstavlja spisak lokacija koje nastavnicima više odgovaraju. Ukoliko je ovaj spisak prazan, to znači da nastavnik nema preferenciju. Tabela je prikazana na slici 4.4

Tabela predmeti predstavlja izdvojene podatke o svim predmetima koji se drže na fakultetu. Kolona „Ime” predstavlja šifru predmeta. Može da se desi da više pred-

	Ime	Željeno vreme	Neželjeno vreme	Željene lokacije
0	Nina Radojičić Matić	[1, 2, 3, 4, 5, 6]	[8, 9, 10]	[Studentski trg]
1	Mirjana Maljković	[1, 2, 3, 4, 5, 6, 11, 12, 13]	[26, 27, 28, 29, 30, 31]	[Studentski trg]
2	Ivan Drecun	[31, 32, 33]	[1, 2, 3, 4, 5, 6]	[Sv. Nikole]
3	Luka Jovičić	[]	[1, 2, 3]	[Sv. Nikole, Jagic]
4	Marko Spasić	[0, 1, 2]	[]	[Sv. Nikole, Jagic]
5	Sana Stojanović Đurđević	[]	[]	[Jagic]
6	Ivana Tomašević	[]	[]	[Sv. Nikole, Jagic]
7	Jelena Marković	[]	[1, 2, 3, 4, 5, 6]	[Studentski trg, Sv. Nikole]
8	Dara Milojković	[22, 23, 24, 32, 33, 34]	[1, 2, 3, 4, 5, 6]	[Studentski trg]
9	Milan Čugurović	[]	[]	[]

Slika 4.4: Prikaz dela spiska nastavnika

meta ima isto ime, pa se predmeti jedinstveno određuju prema njihovim šiframa. Sufiks „vežbe” označava da konkretni podatak predstavlja vežbe. Vežbe i predavanja se tretiraju isto prilikom rada algoritma. Kolona „Oprema” predstavlja informacije o potrebnoj opremi za držanje predmeta. Neophodno je da učionica u kojoj se preraspodeli predmet bude opremljena odgovarajućom opremom. Kolona „Broj časova” predstavlja broj nedeljnih sati predviđenih za predmet. Na Matematičkom fakultetu predmeti se održavaju jednom nedeljno, što znači da svi sati moraju da budu neposredno jedan za drugim. Na slici 4.5 je prikazana ova tabela.

Za tabelu Grupe se koristi ulazna tabela Relacije između grupa 4.2. Dodata je kolona „Deca”, koja predstavlja listu podgrupa jedne grupe, kako bi se olakšale provere tokom rada algoritma.

Tabela Podela nastave je najbitnija tabela za sastavljanje rasporeda časova. Ona predstavlja relaciju između nastavnika, predmeta i grupa. Jedno predavanje u podeli nastave može da se odnosi na više grupa, zato što je moguće da više grupa sluša isto predavanje u istom terminu. Na primer, „RM07” zajedno slušaju grupe studenata „2n”, „4v” i „4l” kod nastavnika Mirka Spasića. Tabela je prikazana na slici 4.6

4.3 Rezultat algoritma

Algoritam svakom predavanju iz tabele Podela nastave dodeljuje početne vrednosti za vremenski interval i učionicu. Ovo je predstavljeno tabelom Raspored časova. U ovoj tabeli red sa indeksom ”n” se odnosi na predavanje u tabeli Podela nastava

	Ime	Oprema	Broj časova
17	R210 - vežbe	računari	2
18	R240 - vežbe	računari	3
19	R245 - vežbe	računari	3
20	RM01		2
21	RM10 - vežbe	računari	2
22	RM01 - vežbe	računari	3
23	R376		2
24	MFRI		2
25	R332		2
26	R390		2
27	UUAAVI		2
28	R272 - vežbe	računari	3
29	RM16 - vežbe	računari	2
30	RM14		2
31	O12		2
32	R310 - vežbe	računari	3
33	R260 - vežbe	računari	2
34	R269 - vežbe	računari	3
35	UUHJVI		1
36	R392 - vežbe	računari	3
37	R220		3

Slika 4.5: Prikaz dela spiska predmeta

sa tim istim indeksom. Na osnovu ovoga je u potpunosti definisan raspored časova. Tabela je prikazana na slici 4.7.

Tabela Podela nastave i raspored časova se spajaju u jednu tabelu kako bi se dobio konačan raspored časova sa svim informacijama. Takođe, prikazana je i kolona Broj časova koja zajedno sa početnim vremenskim intervalom definiše satnicu svakog predavanja. Ova tabela je prikazana na slici 4.8.

4.4 Prikaz rasporeda

Za potrebe prikaza rasporeda dobijenog iz algoritma, napravljena je veb aplikacija. Korisnik može da pregleda raspored konkretne studentske grupe, nastavnika ili učionice, prikaz rasporeda se može videti na slici 4.9.

	Nastavnik	Predmet	Grupe
0	Sana Stojanović Đurđević	RM02	[1o1]
1	Vladan Kovačević	RM02 - vežbe	[1o1a]
2	Ivana Tomašević	RM02	[1o2]
3	Vladan Kovačević	RM02 - vežbe	[1o1b]
4	Ivan Čukić	RM02	[1o3]
5	Vladan Kovačević	RM02 - vežbe	[1o2a]
6	Ivan Čukić	RM02	[1o4]
7	Andrijana Aleksić	RM02 - vežbe	[1o2b]
8	Filip Vidojević	RM02 - vežbe	[1o3a]
9	Vukan Antić	RM02 - vežbe	[1o3b]
10	Filip Vidojević	RM02 - vežbe	[1o4a]
11	Dara Milojković	RM02 - vežbe	[1o4b]
12	Mirko Spasić	RM07	[4v, 2n, 4l]
13	Dara Milojković	RM07 - vežbe	[4v, 2n, 4l]
14	Staša Vujičić Stanković	RM04	[2v, 2n, 2m]
15	Bojana Jošić	RM04 - vežbe	[2va, 2ma, 2na]
16	Aleksandar Kartelj	RM04	[2l1, 2r1]
17	Bojana Jošić	RM04 - vežbe	[2mb, 2vb, 2nb]
18	Aleksandar Kartelj	RM04	[2r2, 2l2]
19	Bojana Jošić	RM04 - vežbe	[2r1a, 2l1a]

Slika 4.6: Prikaz dela podele nastave

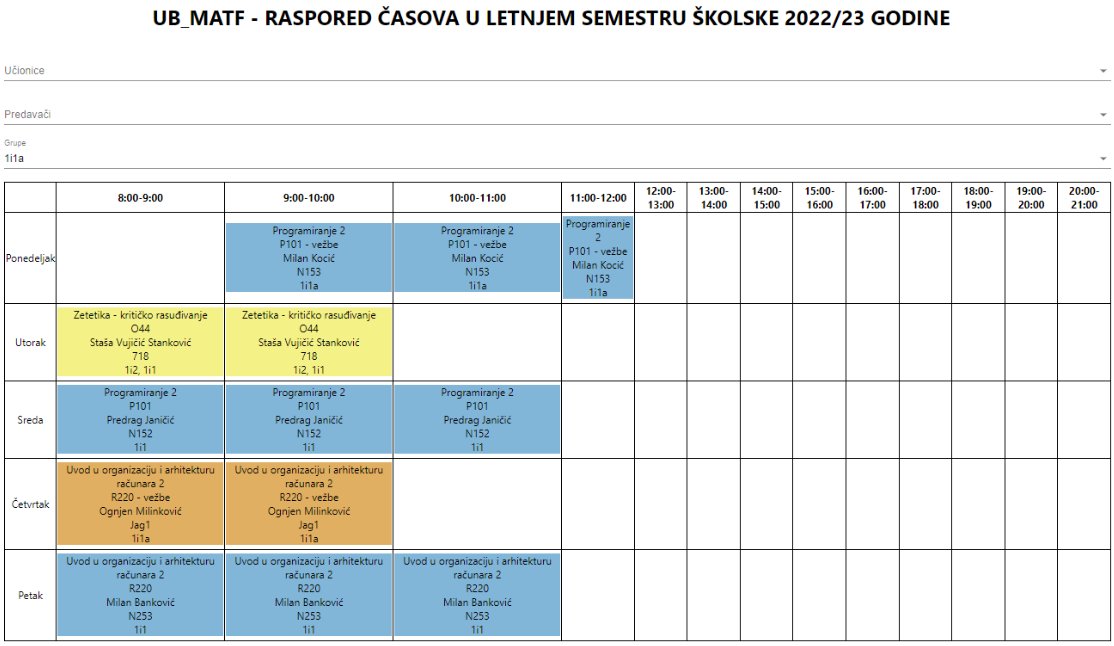
	Učionica	Početni vremenski interval
0	406	0
1	704	13
2	706	13
3	718	16
4	821	0
5	830	0
6	840	26
7	843	0
8	844	28
9	BIM	13
10	RLAB	3
11	N102	6
12	N103	13
13	N151	15
14	N152	0
15	N153	26
16	N201	23
17	N202	17
18	N206	2
19	N225	28

Slika 4.7: Prikaz dela rasporeda časova

	Nastavnik	Predmet	Grupe	Učionica	Početni vremenski interval	Broj časova
0	Sana Stojanović Đurđević	RM02	[1o1]	406	0	2
1	Vladan Kovačević	RM02 - vežbe	[1o1a]	704	13	3
2	Ivana Tomašević	RM02	[1o2]	706	13	2
3	Vladan Kovačević	RM02 - vežbe	[1o1b]	718	16	3
4	Ivan Čukić	RM02	[1o3]	821	0	2
5	Vladan Kovačević	RM02 - vežbe	[1o2a]	830	0	3
6	Ivan Čukić	RM02	[1o4]	840	26	2
7	Andrijana Aleksić	RM02 - vežbe	[1o2b]	843	0	3
8	Filip Vidojević	RM02 - vežbe	[1o3a]	844	28	3
9	Vukan Antić	RM02 - vežbe	[1o3b]	BIM	13	3
10	Filip Vidojević	RM02 - vežbe	[1o4a]	RLAB	3	3
11	Dara Milojković	RM02 - vežbe	[1o4b]	N102	6	3
12	Mirko Spasić	RM07	[4v, 2n, 4l]	N103	13	2
13	Dara Milojković	RM07 - vežbe	[4v, 2n, 4l]	N151	15	2
14	Staša Vujičić Stanković	RM04	[2v, 2n, 2m]	N152	0	2
15	Bojana Jošić	RM04 - vežbe	[2va, 2ma, 2na]	N153	26	2
16	Aleksandar Kartelj	RM04	[2l1, 2r1]	N201	23	2
17	Bojana Jošić	RM04 - vežbe	[2mb, 2vb, 2nb]	N202	17	2
18	Aleksandar Kartelj	RM04	[2r2, 2l2]	N206	2	2
19	Bojana Jošić	RM04 - vežbe	[2r1a, 2l1a]	N225	28	2

Slika 4.8: Prikaz dela kompletnog rasporeda časova

Na slikama 4.10, 4.11 i 4.12 su prikazane padajuće liste za odabir učionice, nastavnika i studentske grupe. Na slici 4.13 prikazana je dostupnost učionice „718”. Na 4.14 se može videti u kojim terminima predavač „Boris Cvitak” drži nastavu. Raspored časova za Master studije na smeru Informatika (studentska grupa „5i”) je prikazan na slici 4.15.



UB_MATF - RASPORED ČASOVA U LETNJEM SEMESTRU ŠKOLSKE 2022/23 GODINE

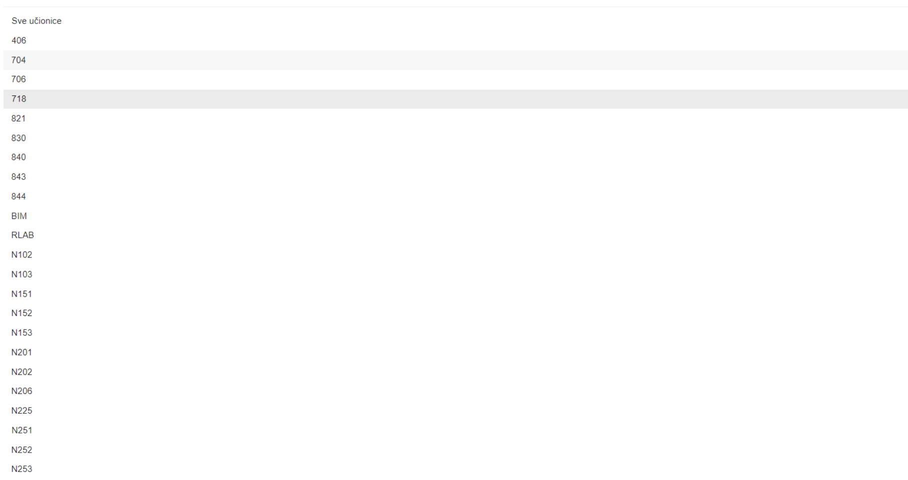
Učionice

Predavači

Grupe
1i1a

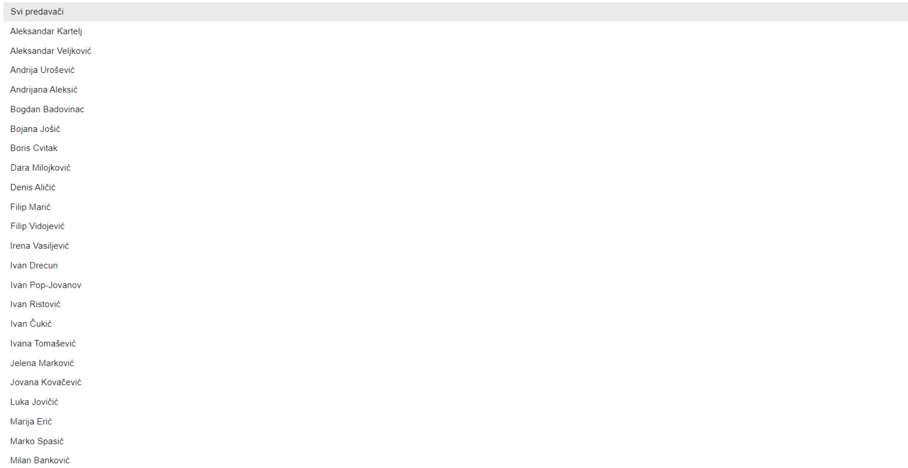
	8:00-9:00	9:00-10:00	10:00-11:00	11:00-12:00	12:00-13:00	13:00-14:00	14:00-15:00	15:00-16:00	16:00-17:00	17:00-18:00	18:00-19:00	19:00-20:00	20:00-21:00
Ponedeljak		Programiranje 2 P101 - vežbe Milan Kocić N153 1i1a	Programiranje 2 P101 - vežbe Milan Kocić N153 1i1a	Programiranje 2 P101 - vežbe Milan Kocić N153 1i1a									
Utorak	Zetetika - kritičko rasuđivanje O44 Staša Vujičić Stanković 718 1i2, 1i1	Zetetika - kritičko rasuđivanje O44 Staša Vujičić Stanković 718 1i2, 1i1											
Sreda	Programiranje 2 P101 Predrag Janičić N152 1i1	Programiranje 2 P101 Predrag Janičić N152 1i1	Programiranje 2 P101 Predrag Janičić N152 1i1										
Četvrtak	Uvod u organizaciju i arhitekturu računara 2 R220 - vežbe Ognjen Milinković Jag1 1i1a	Uvod u organizaciju i arhitekturu računara 2 R220 - vežbe Ognjen Milinković Jag1 1i1a											
Petak	Uvod u organizaciju i arhitekturu računara 2 R220 Milan Banković N253 1i1	Uvod u organizaciju i arhitekturu računara 2 R220 Milan Banković N253 1i1	Uvod u organizaciju i arhitekturu računara 2 R220 Milan Banković N253 1i1										

Slika 4.9: Raspored časova za studentsku grupu 1i1a

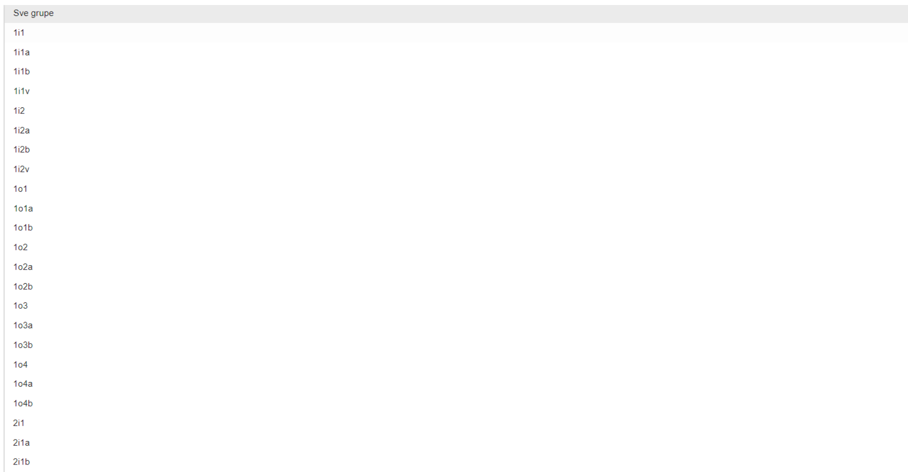


Sve učionice
406
704
706
718
821
830
840
843
844
BIM
RLAB
N102
N103
N151
N152
N153
N201
N202
N206
N225
N251
N252
N253

Slika 4.10: Odabir učionice



Slika 4.11: Odabir nastavnika

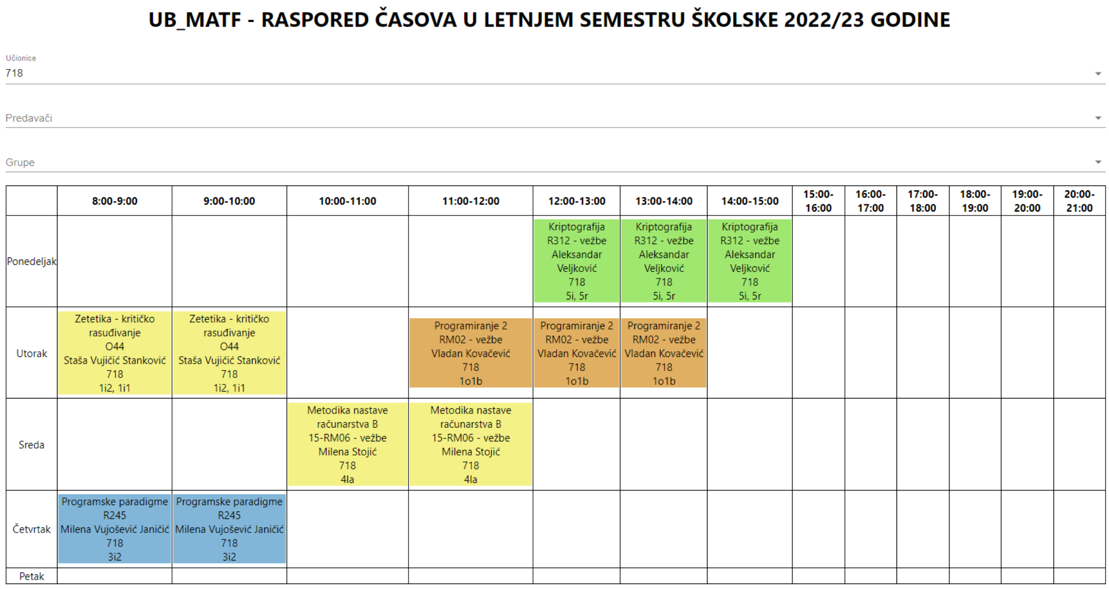


Slika 4.12: Odabir studentske grupe

4.5 Analiza rezultata

Podaci nad kojima je demostriran algoritam su predavanja sa Katedre za Računarstvo Matematičkog fakulteta u letnjem semestru 2023. godine. Na tabeli 4.1 se mogu videti veličine svih tabela korišćenih u algoritmu.

Algoritam prikazan u radu je uspešno sastavio dopustiv raspored časova, to jest bez narušavanja tvrdih ograničenja. Dodatno, u dobijenom rasporedu časova nije narušeno nijedno meko ograničenje. Nastavnici i studenti ne menjaju lokaciju i nemaju pauze između predavanja u jednom danu. Iako je algoritam našao optimalno rešenje, treba imati u vidu da je za ulazne podatke uzeta samo jedna katedra fakulteta. Ukoliko bi se koristila potpuna podela nastave fakulteta, moguće je da ne bi moglo da se nađe rešenje bez ijednog narušenog mekog ograničenja.



UB_MATF - RASPORED ČASOVA U LETNJEM SEMESTRU ŠKOLSKE 2022/23 GODINE

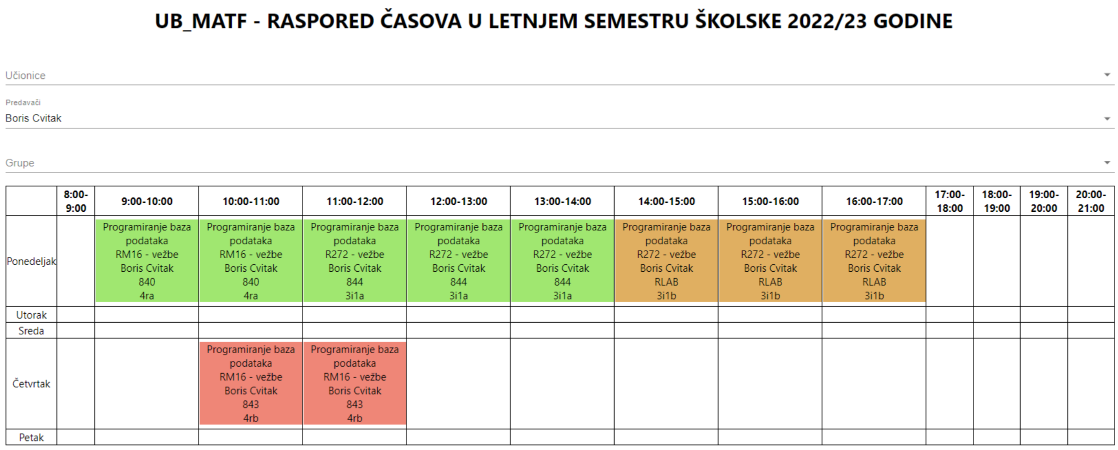
Učionice: 718

Predavači

Grupe

	8:00-9:00	9:00-10:00	10:00-11:00	11:00-12:00	12:00-13:00	13:00-14:00	14:00-15:00	15:00-16:00	16:00-17:00	17:00-18:00	18:00-19:00	19:00-20:00	20:00-21:00
Ponedeljak					Kriptografija R312 - vežbe Aleksandar Veljković 718 5i, 5r	Kriptografija R312 - vežbe Aleksandar Veljković 718 5i, 5r	Kriptografija R312 - vežbe Aleksandar Veljković 718 5i, 5r						
Utorak	Zetetika - kritičko rasuđivanje O44 Staša Vujičić Stanković 718 1i2, 1i1	Zetetika - kritičko rasuđivanje O44 Staša Vujičić Stanković 718 1i2, 1i1		Programiranje 2 RM02 - vežbe Vladan Kovačević 718 1o1b	Programiranje 2 RM02 - vežbe Vladan Kovačević 718 1o1b	Programiranje 2 RM02 - vežbe Vladan Kovačević 718 1o1b							
Sreda			Metodika nastave računarstva B 15-RM06 - vežbe Milena Stojić 718 4la	Metodika nastave računarstva B 15-RM06 - vežbe Milena Stojić 718 4la									
Četvrtak	Programske paradigme R245 Milena Vujošević Janičić 718 3i2	Programske paradigme R245 Milena Vujošević Janičić 718 3i2											
Petak													

Slika 4.13: Predavanja u učionici 718



UB_MATF - RASPORED ČASOVA U LETNJEM SEMESTRU ŠKOLSKE 2022/23 GODINE

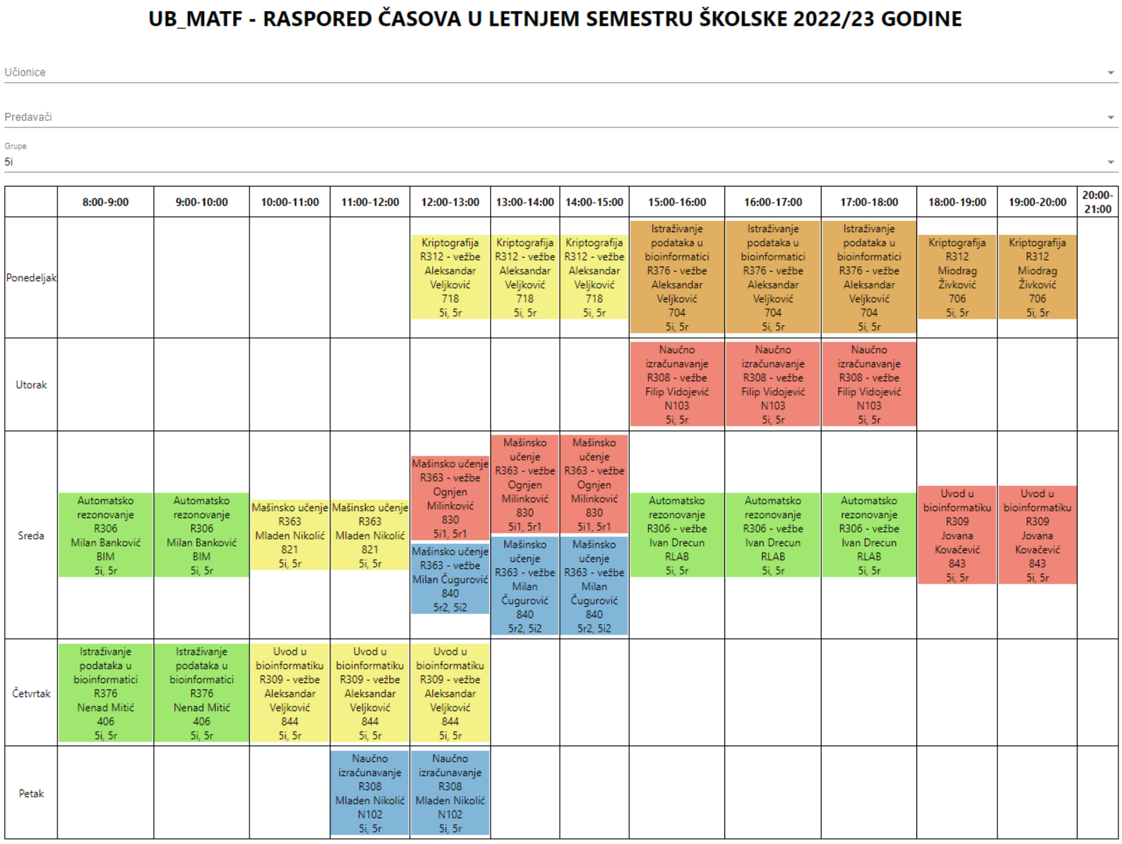
Učionice

Predavači: Boris Cvitak

Grupe

	8:00-9:00	9:00-10:00	10:00-11:00	11:00-12:00	12:00-13:00	13:00-14:00	14:00-15:00	15:00-16:00	16:00-17:00	17:00-18:00	18:00-19:00	19:00-20:00	20:00-21:00
Ponedeljak		Programiranje baza podataka RM16 - vežbe Boris Cvitak 840 4ra	Programiranje baza podataka RM16 - vežbe Boris Cvitak 840 4ra	Programiranje baza podataka R272 - vežbe Boris Cvitak 844 3i1a	Programiranje baza podataka R272 - vežbe Boris Cvitak 844 3i1a	Programiranje baza podataka R272 - vežbe Boris Cvitak 844 3i1a	Programiranje baza podataka R272 - vežbe Boris Cvitak RLAB 3i1b	Programiranje baza podataka R272 - vežbe Boris Cvitak RLAB 3i1b	Programiranje baza podataka R272 - vežbe Boris Cvitak RLAB 3i1b				
Utorak													
Sreda													
Četvrtak			Programiranje baza podataka RM16 - vežbe Boris Cvitak 843 4rb	Programiranje baza podataka RM16 - vežbe Boris Cvitak 843 4rb									
Petak													

Slika 4.14: Raspored časova nastavnika Borisa Cvitka



UB_MATF - RASPORED ČASOVA U LETNJEM SEMESTRU ŠKOLSKE 2022/23 GODINE

Učionice

Predavači

Grupe
5i

	8:00-9:00	9:00-10:00	10:00-11:00	11:00-12:00	12:00-13:00	13:00-14:00	14:00-15:00	15:00-16:00	16:00-17:00	17:00-18:00	18:00-19:00	19:00-20:00	20:00-21:00
Ponedeljak					Kriptografija R312 - vežbe Aleksandar Veljković 718 5i, 5r	Kriptografija R312 - vežbe Aleksandar Veljković 718 5i, 5r	Kriptografija R312 - vežbe Aleksandar Veljković 718 5i, 5r	Istraživanje podataka u bioinformatici R376 - vežbe Aleksandar Veljković 704 5i, 5r	Istraživanje podataka u bioinformatici R376 - vežbe Aleksandar Veljković 704 5i, 5r	Istraživanje podataka u bioinformatici R376 - vežbe Aleksandar Veljković 704 5i, 5r	Kriptografija R312 Miodrag Živković 706 5i, 5r	Kriptografija R312 Miodrag Živković 706 5i, 5r	
Utorak								Naučno izračunavanje R308 - vežbe Filip Vidojević N103 5i, 5r	Naučno izračunavanje R308 - vežbe Filip Vidojević N103 5i, 5r	Naučno izračunavanje R308 - vežbe Filip Vidojević N103 5i, 5r			
Sreda	Automatsko rezonovanje R306 Milan Banković BIM 5i, 5r	Automatsko rezonovanje R306 Milan Banković BIM 5i, 5r	Mašinsko učenje R363 Mladen Nikolić 821 5i, 5r	Mašinsko učenje R363 Mladen Nikolić 821 5i, 5r	Mašinsko učenje R363 - vežbe Ognjen Milinković 830 5i1, 5r1 Mašinsko učenje R363 - vežbe Milan Čugurović 840 5r2, 5i2	Mašinsko učenje R363 - vežbe Ognjen Milinković 830 5i1, 5r1 Mašinsko učenje R363 - vežbe Milan Čugurović 840 5r2, 5i2	Mašinsko učenje R363 - vežbe Ognjen Milinković 830 5i1, 5r1 Mašinsko učenje R363 - vežbe Milan Čugurović 840 5r2, 5i2	Automatsko rezonovanje R306 - vežbe Ivan Drecun RLAB 5i, 5r	Automatsko rezonovanje R306 - vežbe Ivan Drecun RLAB 5i, 5r	Automatsko rezonovanje R306 - vežbe Ivan Drecun RLAB 5i, 5r	Uvod u bioinformatiku R309 Jovana Kovačević 843 5i, 5r	Uvod u bioinformatiku R309 Jovana Kovačević 843 5i, 5r	
Četvrtak	Istraživanje podataka u bioinformatici R376 Nenad Mitić 406 5i, 5r	Istraživanje podataka u bioinformatici R376 Nenad Mitić 406 5i, 5r	Uvod u bioinformatiku R309 - vežbe Aleksandar Veljković 844 5i, 5r	Uvod u bioinformatiku R309 - vežbe Aleksandar Veljković 844 5i, 5r	Uvod u bioinformatiku R309 - vežbe Aleksandar Veljković 844 5i, 5r								
Petak				Naučno izračunavanje R308 Mladen Nikolić N102 5i, 5r	Naučno izračunavanje R308 Mladen Nikolić N102 5i, 5r								

Slika 4.15: Raspored časova studentske grupe 5i

Tabela	Broj redova
Predavanja	131
Nastavnici	55
Predmeti	148
Studentske grupe	82
Učionice	27
Školski časovi	65
Lokacije	3

Tabela 4.1: Rezultati

Za ulazne podatke korišćene u radu, algoritam se u proseku izvrši za 17 minuta. Prvo dopustivo rešenje se pronađe nakon 2-3 iteracije algoritma. Do tad, broj mekih ograničenja ostaje relativno isti inicijalnom rešenju. U prvoj iteraciji nakon pronalaska prvog dopustivog rešenja, broj narušenih mekih ograničenja eksponencijalno opada do nekog lokalnog optimuma. U toj tački može ostati nekoliko narednih iteracija, sve dok se okolina ne proširi dovoljno i time nađe rešenje koje ne konvergira ka istom lokalnom optimumu. U sledećoj iteraciji se može doći do nekog drugog lokalnog optimuma. Ovakvo ponašanje se u proseku ponavlja 2-3 puta, nakon čega se nalazi rešenje u kom nijedno meko ograničenje nije narušeno. U nekoj iteraciji proširivanja okoline može da se desi da se iz dopustivog rešenja pređe u nedopustivo sa malim brojem narušenih tvrdih ograničenja. Takvo stanje je trenutno, i u narednoj lokalnoj pretrazi se ponovo nađe neko novo dopustivo rešenje.

Glava 5

Zaključak i pravci daljeg razvoja

Ideja ovog rada je bila prikazivanje mogućeg rešenja za problem rasporeda časova na fakultetima. Cilj algoritma je bio da nad datom podelom nastave pronađe dopustiv raspored časova sa što manje narušenih ograničenja. Pretpostavlja se da je u podeli nastave već određeno koji nastavnik će kojoj studentskoj grupi držati određeni predmet. Teorijska predavanja i praktične vežbe se tretiraju kao odvojeni predmeti koje nezavisno predaju profesori i asistenti. Razmatrana su i ograničenja uvedena podelom nastave i preferencama nastavnika (eksplicitna ograničenja), kao i ograničenja zasnovana nad skupom razumnih pravila poput toga da jedan nastavnik ne može držati dva predavanja u isto vreme (implicitna ograničenja). Neka ograničenja moraju biti ispunjena kako bi raspored časova bio dopustiv (tvrda ograničenja) dok je za neka poželjno da budu ispunjena, ali nije neophodno (meka ograničenja). Neka od ograničenja definisanih u ovom radu su podrazumevana za generalni problem rasporeda časova, dok su neka definisana za konkretan primer iz rada. Teško je naći univerzalna ograničenja koja važe za sve fakultete, jer neki fakulteti imaju specifične zahteve koji se krše sa zahtevima drugih fakulteta.

Posmatrani problem je NP-težak i obično se rešava tehnikama programiranja ograničenja što je prirodan pristup, s obzirom da nema jasnu funkciju cilja, već samo ograničenja. Problem je takođe moguće definisati kao problem optimizacije tako što se funkcija cilja deifiniše kao suma svih narušenih ograničenja rešenja. U radu je implementirana upravo jedna takva optimizacija, primenom Metode promenljivih okolina.

Rezultat nad datim ulaznim podacima je pokazao da algoritam može da napravi dopustiv raspored časova bez narušenih ograničenja. S obzirom da su kao podaci korišćena samo predavanja sa jedne katedre (Katedra za Računarstvo), jedan od daljih koraka može biti testiranje algoritma nad celokpunom podelom nastave svih katedri na fakultetu. To bi moglo da znači unošenje novih ograničenja, kao na primer mogućnost da se blok jednog predavanja rastavi u više termina koji su u različitim danima. Dodatno, još jedan korak dalje može biti testiranje algoritma nad podacima sa drugih fakulteta. To bi zahtevalo dodatne modifikacije kôda i dodavanje/oduzimanje ograničenja. Ovo bi dosta sigurnije odredilo da li je algoritam upotrebljiv u praktičnoj primeni. Ukoliko je algoritam primenljiv, mogao bi da se zaista koristi prilikom sastavljanja rasporeda časova u budućnosti.

Postoji još potencijalnih poboljšanja koje bi trebalo istražiti. Pauze između predavanja mogu biti drugačije vrednovane u odnosu na dužinu i vremenski period. Mekim ograničenjima se mogu dodati težine u skladu sa njihovim značajem, što bi zahtevalo prilagođavanje fitnes funkcije tim težinama.

Budući rad bi mogao da se fokusira na rešavanje problema rasporeda časova nekom drugom optimizacionom metodom. Taj rad bi mogao da se testira nad podacima prezentovanim u ovom radu i time bi moglo da se uporedi koji algoritam se ponaša bolje za ovaj konkretan problem.

Bibliografija

[1] Ozan Aki. University exam timetabling using genetic algorithms. In *International Scientific Conference "UNITECH*, volume 1, page 395, 2020.

[2] Intesar Al-Mudahka and Reem Alhamad. On a timetabling problem in the health care system. *RAIRO-Operations Research*, 56(6):4347–4362, 2022.

[3] Diego Arenas, Rémy Chevrier, Said Hanafi, and Joaquin Rodriguez. Solving the train timetabling problem, a mathematical model and a genetic algorithm solution approach. In *6th international conference on railway operations modelling and analysis (RailTokyo2015)*, 2015.

[4] Hishammuddin Asmuni. *Fuzzy methodologies for automated university timetabling solution construction and evaluation*. PhD thesis, University of Nottingham, 2008.

[5] Kenneth R Baker. *Introduction to sequencing and scheduling*. Wiley, 1974.

[6] Rahma Borchani, Abdelkerim Elloumi, and Malek Masmoudi. Variable neighborhood descent search based algorithms for course timetabling problem: Application to a tunisian university. *Electronic Notes in Discrete Mathematics*, 58:119–126, 2017.

[7] John Nigel Gaunt Brittan. *The construction of academic timetables by computer*. PhD thesis, Imperial College London, UK, 1974.

[8] Shimon Even, Alon Itai, and Adi Shamir. On the complexity of time table and multi-commodity flow problems. In *16th Annual Symposium on Foundations of Computer Science (sfcs 1975)*, pages 184–193, 1975.

[9] Mohammad-Reza Feizi-Derakhshi, Hamed Babaei, and Javad Heidarzadeh. A survey of approaches for university course timetabling problem. In *Proceedings*

of 8th international symposium on intelligent and manufacturing systems, Sakarya University Department of Industrial Engineering, Adrasan, Antalya, Turkey, pages 307–321, 2012.

[10] Jacques A. Ferland and Serge Roy. Timetabling problem for university as assignment of activities to resources. *Comput. Oper. Res.*, 12:207–218, 1985.

[11] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., 1990.

[12] Calvin C. Gotlieb. The construction of class-teacher time-tables. In *IFIP Congress*, 1962.

[13] Joe Henry Obit. *Developing novel meta-heuristic, hyper-heuristic and cooperative search for course timetabling problems.* PhD thesis, University of Nottingham, 2010.

[14] Alain Hertz. Tabu search for large scale timetabling problems. *European Journal of Operational Research*, 54(1):39–47, 1991.

[15] Alain Hertz. Finding a feasible course schedule using tabu search. *Discrete Applied Mathematics*, 35(3):255–270, 1992.

[16] Liviu Lalescu and Costin Badica. Timetabling experiments using genetic algorithms. *Proc. TAINN*, 3:221–225, 2003.

[17] Rhydian Lewis. A survey of metaheuristic-based techniques for university timetabling problems. *OR Spectrum*, 30:167–190, 2008.

[18] Rhydian M. R. Lewis. *Metaheuristics for university course timetabling.* PhD thesis, Edinburgh Napier University, 2006.

[19] Kim-Fung Man, Kit-Sang Tang, and Sam Kwong. Genetic algorithms: concepts and applications [in engineering design]. *IEEE Transactions on Industrial Electronics*, 43(5):519–534, 1996.

[20] Amnon Meisels and Andrea Schaerf. Modelling and solving employee timetabling problems. *Annals of Mathematics and Artificial Intelligence*, 39:41–59, 2003.

[21] Nenad Mladenović and Pierre Hansen. Variable neighborhood search. *Computers Operations Research*, 24(11):1097–1100, 1997.

[22] Timothy Anton Redl. *A study of university timetabling that blends graph coloring with the satisfaction of various essential and preferential conditions*. Rice University, 2004.

[23] Andrea Schaerf. A survey of automated timetabling. *Artificial intelligence review*, 13:87–127, 1999.

[24] Gunther Schmidt and Thomas Ströhlein. Timetable construction – an annotated bibliography. *The Computer Journal*, 23(4):307–316, 1980.

[25] Selim M. Selim. Split vertices in vertex colouring and their application in developing a solution to the faculty timetable problem. *The Computer Journal*, 31(1):76–82, 1988.

[26] Joo Siang Tan, Say Leng Goh, Graham Kendall, and Nasser R. Sabar. A survey of the state-of-the-art of optimisation methodologies in school timetabling problems. *Expert Systems with Applications*, 165:113943, 2021.

[27] Arabinda Tripathy. School timetabling – a case in large binary integer linear programming. *Management Science*, 30(12):1473–1489, 1984.

[28] Dalessandro Vianna, Carlos Bazilio, Thiago Lima, Marcilene Vianna, and Edwin Mitacc Meza. Hybrid vns-ts heuristics for university course timetabling problem. *Brazilian Journal of Operations Production Management*, 17(2):1–20, 2020.

[29] Anthony Wren. Scheduling, timetabling and rostering — a special relationship? In *Practice and Theory of Automated Timetabling*, pages 46–75, 1996.

[30] Petar Zecevic. Automatic timetable generation. `https://github.com/PetarZecevic97/automatic-timetable-generation`, 2023.

Biografija autora

Petar Zečević rođen je u Užicu 13.02.1997.

Pohađao je Osnovnu školu "Nada Matić", gde je ostvario Vukovu diplomu. Od 2012. do 2016. godine pohađao je Užičku gimnaziju, društveno-jezički smer. Godine 2015. dobio je sertifikat Kembridž Napredni Engleski (eng. Cambridge Advanced English - CAE).

Od 2016. do 2020. godine studirao je na Matematičkom fakultetu Univerziteta u Beogradu, na informatičkom smeru. Diplomirao je 2020. godine i upisao master studije na istom smeru. Ušestvovao je na SYM-OP-IS 2020 simpozijumu sa naučnim radom iz oblasti klasterovanja.

Godine 2021. počeo je svoju karijeru u softverskom inženjerstvu u firmi *Zühlke Engineering* prvobitno kao praktikant, zatim kao Profesionalni Softver Inženjer, i danas kao Napredni Softver Inženjer. U toku karijere je dobio sertifikat Profesionalni Skram Programer (eng. Professional Scrum Developer - PSD). U saradnji firme *Zühlke Engineering* i Matematičkog fakulteta, 2023. godine je držao stručni kurs "Razvoj Bezbednog Softvera" na fakultetu.