Univerzitet u Beogradu
Matematički fakultet

master rad

Rešavanje problema raspoređivanja časova primenom optimizacionih tehnika na predloženi model 0-1 celobrojnog linearnog programiranja

Mentor:
dr Zorica Stanimirović

Student:
Marko Kaznovac, 1047/2009

Beograd
septembar 2012.

Sadržaj

1. Uvod

Konstrukcija rasporeda koji zadovoljava postavljena pravila, ograničenja, želje i zahteve nastavnog osoblja i studenata je zahtevan i često dugotrajan proces.

Problem izrade rasporeda časova podrazumeva pronalaženje zadovoljavajućeg redosleda aktivnosti, kako za predavača tako i za studente, uobičajeno za nedelju dana, uz poštovanje nekih ograničenja. Najčešći problemi koje treba rešiti su da nema preklapanja predavanja za kurseve jedne godine, zauzetost učionica i laboratorija, velike pauze između dva predmeta i sl.

Ručna izrada rasporeda može potrajati i nekoliko dana. Prvo rešenje za automatsku izradu rasporeda časova dao je *Gotlieb 1962* godine na IFIP (*International Federal for Information Processing*) kongresu u Minhenu. [**1**] [**2**]

Mnogi i dalje veruju da se konstruisanje rasporeda časova ne može kompletno automatizovati pa pojedini sistemi dozvoljavaju ručno podešavanje i u tom slučaju govorimo o interaktivnom ili semi-automatskom sistemu za izradu rasporeda.

Najranija tehnika rešenje problema je bazirala na sukcesivnom dodavanju nastavnih časova (predavanja, računarskih i laboratorijskih vežbi) dok se sve ne bi našle u rasporedu. Ideja je da se prvo dodele termini najkritičnijim lekcijama, što je ostavljalo dilemu šta je "najkritičnije". Kasnija istraživanja su koristila algoritme za optimizaciju funkcije cilja celobrojnog linearnog programiranja, protoka mreže i dr. Najnoviji pristup bazira na približnim tehnikama i heuristici, kao što su: simulirano kaljenje, tabu pretraga, genetski algoritmi i dr.

Problem određivanja rasporeda časova može se podeliti na tri grupe:

1. *Izrada školskog rasporeda časova* – raspoređivanje časova za nedelju dana pri čemu treba izbeći preklapanje časova za predavače
2. *Izrada rasporeda fakultetskih predavanja* – raspoređivanje predavanja za grupe studenata bez preklapanja
3. *Određivanje termina ispita* – dodela termina svakom ispitu bez preklapanja i što boljeg, "šireg", rasporeda u datom roku

Osnovna razlika izrade univerzitetskog i školskog rasporeda je u grupama koje pohađaju časove (predavanja). U osnovnim ili srednjim školama odeljenja su nepromenljiva, dok grupe studenata na univerzitetima nisu obavezno disjunktne.

Izrada rasporeda polaganja ispita podrazumeva dodelu dana i vremena polaganja za svaki ispit. Specifična ograničenja kod dodele termina ispitima su:

- postoji samo jedan termin za ispit po predmetu u jednom roku
- preklapanja ispita su striktno zabranjena za individualne studente
- moguće je održavanje više od jednog ispita u učionici
- za svakog studenta poželjno je planirati najviše jedan ispit dnevno, i rasporediti termine tokom roka sa što većim razmakom

U ovom radu prevashodno je dat osvrt na izradu rasporeda za fakultetska predavanja.

2. Izrada rasporeda predavanja na univerzitetu

Izrada rasporeda za predavanja na fakultetima podrazumeva dodelu termina svakom predavanju za svaku grupu poštujući raspoloživ broj učionica za dati period. Za razliku od časova u osnovnim ili srednjim školama kojima prisustvuju cela odeljenja, za predavanja na fakultetu posmatra se pojedinačno svaki student kome treba omogućiti pohađanje svih raspoloživih predmeta jednog semestra u tekućoj godini.

2.1. Karakteristike rasporeda časova

Konstrukcija rasporeda treba da zadovolji sva operativna pravila i potrebe i u isto vreme želje i zahteve nastavnog osoblja i studenata, poštujući uz to i raspoložive resurse.

2.1.1. Struktura univerzitetskih kurseva

Upoznavanje materije svakog predmeta odvija se kroz kombinaciju predavanja, vežbi i praktičnog rada u specijalizovanim laboratorijama.

Predmeti se dodeljuju nastavnom osoblju i mogu se u nedelji održavati jednom ili u više sesija. Poželjno je da se vežbe odvijaju nakon predavanja radi utvrđivanja gradiva. Grupe studenata za vežbe se dele na manje, pri čemu obično svaka grupa ima svog asistenta. Za pojedine predmete organizuje se i praktičan rad u specijalno opremljenim učionicama.

2.1.2. Tipovi kurseva

Predmeti mogu biti obavezni ili izborni. Predavanja za obavezne predmete se organizuju za sve studente jedne godine. Izborni, iako uglavnom imaju manji broj polaznika, ne smeju se preklopiti ni sa jednim predmetom u rasporedu.

U prvim godinama osnovnih studija većinu predmeta čine obavezni kursevi i manji broj izbornih, dok u završnim godinama broj izbornih kurseva raste.

2.1.3. Dostupnost resursa

Pod pojmom raspoloživost resursa podrazumevamo dostupnost nastavnog kadra i učionica. Raspored ne sme dozvoliti da jedan predavač istovremeno drži dva predavanja, da se dva predavanja istovremeno drže u istoj učionici, ili da za istu grupu studenata budu predviđena dva različita predavanja u istom terminu.

2.2. Pravila izrade rasporeda

Pravljenje nedeljnog rasporeda je ozbiljan i vremenski zahtevan posao. Tokom procesa izrade rasporeda treba da se poštuju određena pravila. Postoje pravila koja su bitna i ne smeju se narušiti, tzv. osnovna ograničenja (***hard constraints***) i željene osobine (***soft constraints***) koja se kao manje bitna pravila mogu zadovoljiti samo ako su osnovna zadovoljena, i koja tako unapređuju kvalitet rasporeda.

Osnovna ograničenja definišu prostor pretrage, dok su željene osobine deo funkcije cilja, npr. da u rasporedu nema velikih pauza. [**3**]

Osnovna ograničenja (***hard constraints***):

1. kolizije nisu dozvoljene – pod kolizijom u rasporedu časova podrazumevamo sledeće situacije:
   1.1. dodeljena su dva ili više kursa jednom predavaču u jednom terminu
   1.2. dva ili više predavača jednoj grupi studenata u jednom terminu drže različite kurseve
   1.3. dodeljene su dve ili više učionica jednom kursu jednoj grupi studenata
2. raspored mora biti kompletan – svi predviđeni kursevi, za svaku grupu studenata, moraju biti zastupljeni u planiranom broju termina
3. celovitost termina – treba obezbediti da se predavanja održavaju u istoj učionici i u jednoj nedeljivoj sesiji ako predavač ne zahteva drugačije
4. predavači nisu u mogućnosti da održe čas u određenom terminu

Primeri željenih osobina (***soft constraints***):

1. zahtevi predavača – nastavnici žele da predavanje drže u nekom terminu, preferiraju određene dane, vremenske periode u toku dana, žele određenu učionicu, i dr.
2. zahtevi studenata – omogućiti pauzu za vreme izdavanja ručka u studentskim menzama, smanjiti promenu učionica i sl.
3. kompaktnost rasporeda – Poželjno je da u rasporedu u jednom danu ima što manje pauza i da ne budu duže od 1 časa, kako za studente, tako i za predavače
4. teorija i praksa – za neke kurseve potrebno je održati teorijsku nastavu pre praktičnog rada

3. Definicija problema

Postavljena pravila izrade rasporeda se mogu matematički opisati na sledeći način.

3.1. Osnovni problem pretrage

Pretpostavimo da imamo q kurseva $K_1, K_2, \dots, K_q$ i neka za svaki od kurseva K_i postoji k_i predavanja. Dalje imamo r planova nastave $S_1, S_2, \dots, S_r$ koji grupišu predmete koje može da sluša i polaže jedan student. Ovo znači da predmetima u planu S_l moraju biti dodeljeni različiti termini. Označimo sa j konkretan termin predavanja od mogućih p termina, a sa l_j maksimalni broj predavanja koje mogu biti održani u periodu j, što odgovara broju raspoloživih prostorija u periodu j. Potrebno je naći y_{ij} pri čemu važi:

$$y_{ij} = \begin{cases} 1 & \text{ako se predavanje iz kursa } K_i \text{ drži u periodu } j \\ 0 & \text{u suprotnom} \end{cases} \tag{1}$$

Sva predavanja kursa K_i moraju biti u rasporedu

$$\sum_{j=1}^{p} y_{ij} = k_i \quad i = \overline{1,q} \tag{2}$$

Ne može se održati više predavanja od slobodnog broja učionica

$$\sum_{i=1}^{q} y_{ij} \le l_j \quad j = \overline{1,p} \tag{3}$$

Ne dozvoljavaju se dva ili više predavanja u istom periodu

$$\sum_{i \in S_l} y_{ij} \le 1 \quad l = \overline{1,r}, j = \overline{1,p} \tag{4}$$

3.2. Problem optimizacije

Uslovi (1) – (4) definišu prostor pretrage, odnosno dopustiva rešenja. U cilju odabira najadekvatnijeg uvodi se sledeća funkcija cilja koja maksimizuje saglasnost rasporeda sa zadatim željama

$$max \sum_{i=1}^{q} \sum_{j=1}^{p} d_{ij} y_{ij}$$

gde su d_{ij} koeficijenti želja da se kurs K_i drži u periodu j. [**4**]

Problemi pri korišćenju datog modela su višestruki, izlažemo neke od njih:

- ne vodi se računa o prethodno rezervisanim resursima (učionice, profesora, termina)
- ako je broj studenata veći od raspoloživog kapaciteta učionice, potrebno je deliti ih na podgrupe koje će pohađati isti kurs u različitim terminima u istoj nedelji
- predavanja se moraju držati u nedeljivom terminu zahtevanog trajanja (jedan, dva ili više časova). Potrebno je za svaku rezervaciju razmatrati početni termin kao i predviđeno vreme

trajanja. Svake dve lekcije l_i i l_j koje počinju u terminu p i q respektivno, gde je $q > p$, su u konfliktu ako je $q - p < d_{l_i}$, gde je d_{l_i} trajanje lekcije l_i

- ne uzima se u obzir kapacitet učionice, kao ni opremljenost učionice za specijalne zahteve

Postupak za rešavanje ovako postavljenog problema je ***NP-kompletan***. *Garay* i *Johnson* svode problem na problem bojenja grafa. *Even* sa saradnicima dokazuje NP-kompletnost redukcijom problema na 3-SAT. [**5**] [**6**]

4. Moguće tehnike za rešavanje

U ovoj sekciji navodimo neke metode koje su do sada predložene u literaturi za rešavanje problema izrade rasporeda časova.

4.1. Svođenje na problem bojenja grafa

Problem bojenja geografske karte sa 4 boje još iz 1879 je prvi primer bojenja čvorova grafa.

Neka je graf $G = (V, E)$ konačan i bez petlji, pri čemu je V skup čvorova, a E skup grana. Ako se svakom čvoru grafa G dodeli jedan broj (boja) iz skupa $\{1,2,\dots,k\}$ tako da su iskorišćene sve boje iz ovog skupa i bilo koja dva susedna čvora imaju razičite pridružene boje tada se govori o pravilnom k-bojenju grafa ili samo k-bojenju. Graf je k-obojiv ako postoji prirodan broj $l \leq k$ takav da postoji jedno l-bojenje grafa. Hromatski broj $\gamma(G)$ predstavlja najmanje k za koje postoji k-bojenje grafa G. Svako $\gamma(G)$-bojenje grafa G naziva se optimalno bojenje ovog grafa. Određivanje optimalnog bojenja proizvoljnog grafa i njegovog hromatskog broja je ***NP-težak*** problem. Ne postoji analitički izraz za izračunavanje $\gamma(G)$, već se u praksi primenjuju heuristike koje određuju gornju ili donju granicu ovog broja. [**7**] [**6**]

Problem raspoređivanja *de Werra* svodi na problem bojenja grafa ako svaku lekciji l_i iz kursa K_j predstavimo čvorom m_{ij}. Za svaki kurs K_j uvodimo kliku između čvorova $m_{ij},\ i = \overline{1,q}$, takođe pravimo kliku i za svaka dva kursa koja su u konfliktu. Pri konstruisanju grafa grane predstavljaju nedozvoljeno pridruživanje u rasporedu. U slučaju nedostupnosti (nemogućnosti izvršavanja predavanja u nekom periodu) uvodimo p novih tačaka od kojih svaka predstavlja jedan termin. Sve tačke su u kliki što onemogućava da sve budu obojene istom bojom. Ako kurs ne može imati predavanje u datom terminu tada se svi čvorovi koji se odnose na lekcije tog kursa povezuju sa čvorom koji se odnosi na dati termin, suprotno ako je lekcija u rezervisanom terminu, onda se uvode grane ka svim ostalim terminima. [**1**] [**4**]

4.2. Celobrojno linearno programiranje

Opšti oblik linearnog programiranja, za datu matricu $A_{m\times n}$, sa vrstama $V_1,\dots,V_m$ i vektorima $b \in \mathbb{R}^m$ i $c \in \mathbb{R}^n$ može se zapisati na sledeći način:

$$\min c^T x$$

$$\begin{aligned} V_i^T x &= b_i \quad i \in I_1 \\ V_i^T x &\geq b_i \quad i \in I_2 \\ V_i^T x &\leq b_i \quad i \in I_3 \end{aligned}$$

$$x_j \geq 0 \quad j \in J$$

gde je $I_1 \cup I_2 \cup I_3 = \{1,\dots,m\}$ i $I_a \cap I_b = \emptyset$ za $a \neq b,\ a,b \in \{1,\dots,m\}$ i $j \subseteq \{1,\dots,n\}$

Eliminisanjem ograničenja tipa nejednačina pomoću tzv. izravnavajućih promenljivih postavlja se problem kod koga je za sve promenljive nametnut uslov nenegativnosti i tako se dolazi do standardnog oblika problema linearnog programiranja.

$$\min c^T x$$

$$Ax = b$$

$$x \geq 0$$

Opšti oblik se svodi na simetrični oblik problema linearnog programiranja eliminisanjem ograničenja tipa jednačina

$$min\ c^T x$$

$$Ax \geq b$$

$$x \geq 0$$

Najrasprostranjenija tehnika za rešavanje problema linearnog programiranja je ***simpleks*** metoda, a primenjuju se i unutrašnje metode (metod elipsoida, Karmarkarova metoda, i druge). [**7**] [**8**]

Celobrojno programiranje uvodi uslov celobrojnosti koordinata nepoznatog vektora x, što omogućava modeliranje problema iz prakse, jer je broj ljudi, mašina, učionica i sl. po svojoj prirodi diskretna celobrojna veličina. Problem celobrojnog programiranja može se zapisati na sledeći način:

$$min\ c^T x$$

$$Ax = b$$

$$x \geq 0$$

$$x \in \mathbb{Z}^n$$

Najelementarniji način za rešavanje problema optimizacije na konačnom skupu je pretraživanje. Broj koraka pretraživanja može biti veliki, tako da optimalno rešenje nije moguće naći u razumnom vremenskom roku, ni uz pomoć najmoćnijih računara. Iz tog razloga se primenjuju neke od metoda koje eliminišu pretraživanja koja ne dovode do poboljšanja vrednosti funkcije cilja, kao što su metod grananja i odsecanja i metod implicitnog prebrojavanja. Metod grananja i odsecanja odbacuje rešenja koja nisu bolja od već pronađenog. [**7**]

Opšti oblik algoritma grananja i odsecanja:

Korak 0 (ulaz)	$S = \{P\}$, $M(P) = -\infty$, $M = \infty$
Korak 1 (izbor problema)	Izabrati neki problem Q sa spiska
Korak 2 (ograničavanje)	Izračunati $m(Q)$ rešavajući neku relaksaciju problema Q
	Ako je $m(Q) \geq M$ ići na korak 6
Korak 3 (traženje dopustivih rešenja)	Ako je pogodno naći dopustivo rešenje $\hat{x}$ problema P
	Ako je $f(\hat{x}) \geq M$ ili ako je traženje teško ići na korak 5
Korak 4 (popravka najboljeg rešenja)	Staviti $M = f(\hat{x})$
	Ako je $M = -\infty$ postupak se zaustavlja, tada je $v(P) = -\infty$
	Ako je M konačno i ako se može dokazati da je $\hat{x}$ optimalno rešenje za Q ići na korak 6

Korak 5 (grananje)	Zameniti problem Q na spisku sa potproblemima $Q_1, Q_2, \dots, Q_p$ Vratiti se na korak 1
Korak 6 (odsecanje)	Ukloniti Q sa spiska Ako se pri tome spisak isprazni postupak se prekida. Tada je optimalna vrednost jednaka trenutnom rekordu M ako je M konačan broj, a problem nema dopustivih rešenja ako je $M = \infty$ Ako se spisak ne isprazni ići na korak 1

gde su:

- S – trenutni spisak problema
- $v(P)$ optimalna vrednost problema, pri tome je $v(P) = \infty$ ako je problem nemoguć, a $v(P) = -\infty$ ako je vrednost problema neograničena odozdo
- $m(P)$ donja ocena optimalne vrednosti problema P
- M tekuće najbolje rešenje

U slučaju da se u svakom grananju generiše samo konačno mnogo podproblema i da se oni nikad ne ponavljaju, algoritam se posle konačno mnogo iteracija završava nalaženjem optimalne vrednosti i rešenja polaznog problema. [**7**]

Konkretne realizacije metode zavise od načina grananja, relaksacija, izbora i obrade problema.

Poznate metode zasnovane na ovom algoritmu su metode *Land-Doig*-a i *Dakin*-a i metoda implicitnog prebrojavanja.

Problem izrade rasporeda časova primenom celobrojnog linearnog programiranja rešavalo je više autora. *Tripathy* koristi Lagranžovu relaksacionu tehniku. *Ferland* i *Roy* redukuju i rešavaju problem kvadratnog programiranja. U svom radu *Pearl* predlaže korišćenje heurističkih metoda za izradu rasporeda za više od 10 predmeta. [**9**] [**10**] [**11**]

4.3. Problem protoka kroz mrežu

Poznati problem u teoriji grafova je određivanje maksimalnog protoka u mreži. Teorijom protoka u mrežama rešavaju se praktični problemi određivanja optimalnog protoka u saobraćaju, transportu, komunikacionim, informacionim i drugim mrežama. U zavisnosti od prirode posmatranog problema postavljaju se veze između čvorova, ograničenja protoka i intenziteta čvorova.

Neka je data orijentisana mreža $M = (V, E)$, gde $V = \{1,2, \dots, n\}$ predstavlja skup svih čvorova, a $E \subseteq V \times V$ skup grana. Za svaku granu x_{ij} poznati su nam njen l_{ij} donji kapacitet (najčešće 0) i u_{ij} gornji kapacitet.

$$0 \leq l_{ij} \leq x_{ij} \leq u_{ij}$$

Za svaki čvor u mreži treba da važi sledeća jednačina tzv. *jednačina konzervacije protoka*:

$$\sum_{j|(i,j)\in E} x_{ij} - \sum_{j|(j,i)\in E} x_{ji} = b_i, \qquad (\forall i \in G)$$

gde b_i zovemo *intenzitet čvora* i predstavlja razliku toka u čvor i i iz čvora i. Prema intenzitetu možemo svrstati čvor u jednu od klasa:

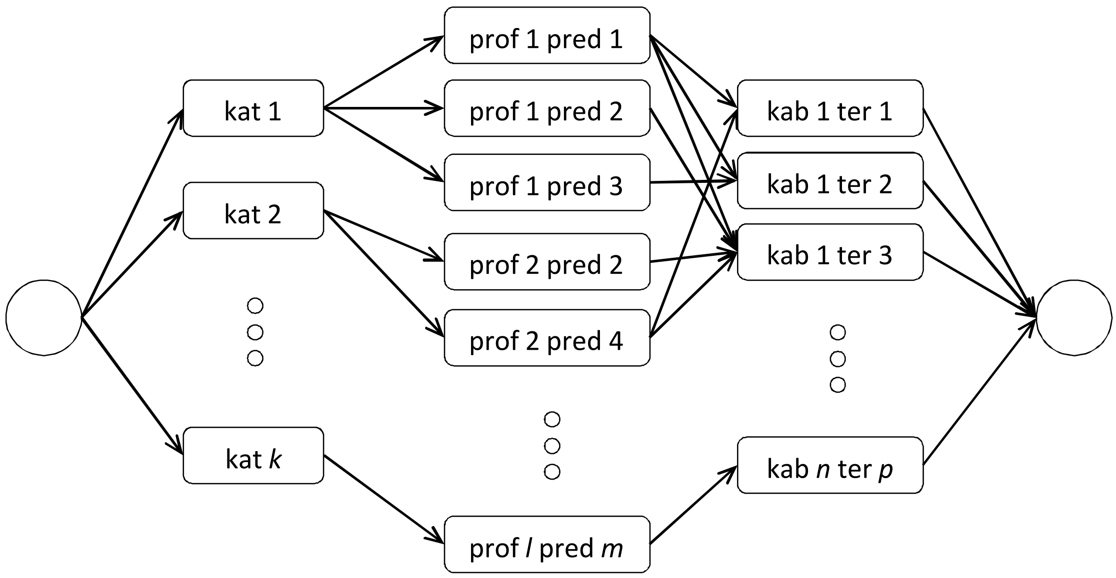
- $b_i > 0$ izvor
- $b_i < 0$ ponor
- $b_i = 0$ neutralni čvor

Ako nam je za svaku granu poznata njena cena upotrebe c_{ij} onda možemo razmatrati sledeću funkciju cilja:

$$min \sum_i \sum_j c_{ij} x_{ij}$$

Ovaj problem zovemo *problem protoka sa minimalnom cenom na mreži* M [**7**].

Dinkel sa saradnicima u mreži izdvaja tri nivoa pored izvora i ponora. Prvi nivo predstavlja podelu na institute ili katedre u okviru fakulteta. Drugi kombinuje predavača i kurseve koji ovaj može da drži pri čemu su povezani sa čvorom odgovarajuće katedre kojoj pripadaju. Na trećem nivou su sve kombinacije veličine učionice i perioda. Ako učionica odgovara za broj studenata i termin predavanja povezuje se sa čvorom nivoa 2. Tok grane između nivoa 2 i 3 predstavlja mogućnost održavanja predavanja uzimajući vrednost 0 ili 1.



Slika 1 mreža problema izrade rasporeda časova

Koeficijenti funkcije cilja baziraju se na raspoloživosti predavača i učionica, kao i željama profesora. Problem protoka kroz mrežu je rešiv u polinomijalnom vremenu, ali se ne može izbeći rešenje u kome se jednom predavaču dupliraju lekcije u jednom periodu, stoga se ručno interveniše i proces ponavlja dok se ne dođe do zadovoljavajućeg rešenja. [**12**]

4.4. Simulirano kaljenje

Heurističke metode polaze od najopštijeg problema kombinatorne optimizacije

$$\min_{x \in X} f(x)$$

gde funkcija cilja $f(x)$ može biti bilo koja funkcija oblika $f: X \to \mathbb{R}$, pri čemu prostor dopustivih rešenja X sadrži konačno mnogo elemenata.

Simulirano kaljenje je strategija pretrage koja u svakoj iteraciji daje po jedno dopustivo rešenje. U svakom koraku generiše se novo dopustivo rešenje, malo izmenjeno u odnosu na prethodno. Novo rešenje se uzima za sledeću iteraciju ako ima bolju ocenu kvaliteta ili shodno nekoj verovatnoći. Na primer verovatnoća se može definisati kao:

$$p_n = e^{-\frac{f(x)-f(x_n)}{t_n}}$$

Ili na neki drugi način.

Verovatnoća je u vezi sa kontrolnim parametrom t_n (nazvan temperatura analogno stvarnom procesu kaljenja), koji uzima vrednosti $t_1, t_2, \dots$ pri čemu važi $t_1 \geq t_2 \geq \cdots$ i $\lim_{n\to\infty} t_n = 0$ (ovakav niz se naziva ***shema hlađenja***) i on definiše način i brzinu smanjivanja temperature tokom iteracije. Polazeći od dovoljno visoke početne temperature t_1 u početnoj iteraciji se prihvataju skoro svi slučajno generisani susedi trenutnog rešenja. Shema hlađenja, prema analogiji sa procesom kaljenja, obezbeđuje dovoljno sporo smanjenje temperature. U praksi se najčešće koristi ***geometrijska shema hlađenja*** preuzeta iz termodinamičke simulacije procesa kaljenja. Početna temperatura t_1 se smanjuje posle svakih L uzastopnih iteracija, $t_n = \alpha^k t_1$, pri čemu $0 < \alpha < 1$ i $kL + 1 \leq n \leq (k+1)L$, $k = 0, 1, \dots$ Pored geometrijske sheme u praksi se često koristi i formula $t_n = \frac{t_{n-1}}{1+\beta t_{n-1}}$, gde je β dovoljno mali parametar. Posle dovoljnog broja iteracija prihvataju se rešenja koja daju poboljšanje funkcije cilja.

Opšti oblik algoritma ove metode:

Inicijalizacija:

- izabrati početno rešenje $x_1 \in X$
- $x^* = x_1$ najbolje rešenje x^* uzima vrednost x_1
- $f^* = f(x_1)$ najbolja vrednost funkcije cilja je vrednost $f(x_1)$

Iterativni korak: $n = 1, 2, \dots$

1. Na slučajan način naći x u okolini $N(x_n)$ trenutnog rešenja x_n
2. Ako $f(x) < f(x_n)$, tada $x_{n+1} = x$
3. Ako $f(x) < f^*$, tada $x^* = x$ i $f^* = f(x)$
4. Ako $f(x) > f(x_n)$ izabrati slučajan broj p uniforman na $[0, 1]$
   - 4.1. Ako je $p \leq p_n$, tada $x_{n+1} = x$
   - 4.2. Ako je $p > p_n$, tada $x_{n+1} = x_n$

Kraj:

- Ako je zadovoljen kriterijum zaustavljanja x^* se uzima za aproksimaciju optimalnog rešenja.

Nedostaci ove metode su što proces „hlađenja“ zahteva mnogo vremena i iteracija da bi se postigli dobri rezultati. [**7**] [**13**] [**14**]

4.5. Tabu pretraga

Tabu pretraživanje se bazira na principu lokalnog pretraživanja i koristi tzv. adaptivnu memoriju tj. pamti podatke o prethodnim fazama procesa pretraživanja koje kasnije utiču na izbor sledećih tačaka. Za svaku iteraciju n čuva se neka istorija H prethodnog pretraživanja. Okolina $N(x_n)$ trenutne tačke x_n se redukuje ili proširuje u zavisnosti od istorije H i tako definiše skup $N(x_n, H)$ koji predstavlja skup svih kandidata za sledeću tačku x_{n+1}. Zbog veličine ovog skupa bira se njegov podskup $N'(x_n)$. Sada se tačka x_{n+1} određuje kao najbolja u skupu $N'(x_n)$ u odnosu na neku funkciju procene $f(x, H)$.

Opšti oblik algoritma:

Inicijalizacija:

- Izabrati početno rešenje $x_1 \in X$
- $x^* = x_1$ najbolje rešenje x^* uzima vrednost x_1
- $f^* = f(x_1)$ najbolja vrednost funkcije cilja je vrednost $f(x_1)$
- $H = \emptyset$ istorija (lista obrađenih kandidata za rešenje)

Iterativni korak: $n = 1,2, \dots$

1. Definisati modifikovanu okolinu $N(x_n, H)$ i funkciju promene $f(x, H)$
2. Generisati podskup $N'(x_n) \subseteq N(x_n, H)$
3. Odrediti x_{n+1} minimizacijom $f(x, H)$ na $N'(x_n)$
4. Ako $f(x_{n+1}) < f^*$, tada $x^* = x_{n+1}$ i $f^* = f(x_{n+1})$
5. Ažurirati istoriju H

Kraj:

- Ako je ispunjen kriterijum zaustavljanja postupak se prekida i x^* se uzima za aproksimaciju optimalnog rešenja.

Najčešće korišćen način formiranja i ažuriranja istorije H je tzv. ***kratkoročna memorija*** kod koje se pamte karakteristike tačaka generisanih u neposrednoj prošlosti. Istorija H se formalno definiše preko tzv. ***tabu lista*** $T_1, T_2, \dots, T_p$ sa svojim dužinama $L_1, L_2, \dots, L_p$, gde je $p \geq 1$. Tabu lista T_i sadrži jednu ili više izabranih karakteristika posećenih tačaka u zadnjih L_i iteracija. Tabu liste određuju ***tabu status*** svakog pomaka $m(x_n, x)$ iz trenutne tačke x_n u neku od njenih tačaka suseda x. Pomak se smatra ***tabu*** ako bar jedan njegov atribut pripada H. Tabu lista T_i se ažurira u svakoj iteraciji odgovarajućim atributima pomaka $m(x_n, x_{n+1})$ i čuvaju se sledećih L_i iteracija. Ovako definisana kratkoročna memorija sprečava vraćanje na neke od ranije generisanih tačaka (izvestan broj iteracija), efektivno proširujući prostor pretrage.

Ovako definisane tabu liste bazirane na praćenju atributa pomaka mogu se pokazati previše restriktivne i zabraniti neke tačke koje nisu bile generisane u prethodnom pretraživanju. Uvodi se kriterijum aspiracije koji može odbaciti tabu status određenog pomaka ako on vodi u „dovoljno dobru“ tačku. Skup kandidata se sada može definisati na sledeći način:

$$N(x_n, H) = \{x \in N(x_n) | m(x_n, x)\ nije\ tabu\ ili\ a(m(x_n, x)) < A(m(x_n, x))\}$$

gde su $a(m(x_n, x))$ ***željeni nivo aspiracije*** i $A(m(x_n, x))$ ***prag aspiracije***, najjednostavnije definisani kao $f(x)$ i $f(x^*)$ respektivno. Tabu status pomaka $m(x_n, x)$ se odbacuje ako je tačka x bolja od trenutno najbolje x^* dobijene do iteracije n. [**7**]

U svom radu *Hertz* razmatra optimizaciju problema uvodeći slučaj predavanja koja zahtevaju više povezanih časova, ali ne razmatra ograničenje lokacije učionice. Ograničenja su podeljena na „teška" *(hard)* (kada su aktivnosti neizvodljive, npr. u istoj učionici dva predavanja ili isti profesor za dve grupe studenata) i „laka" *(soft)* (manje zahtevna ograničenja, npr. kada profesor preferira večernje termine). *Hertz* pretpostavlja da dopustivo rešenje nije definisano hard ograničenjima jer ona ne garantuju povezan prostor pretrage. Stoga dozvoljava da procedura pretrage prolazi kroz rasporede koji ne zadovoljavaju osnovna ograničenja. U ovako dobivenom rasporedu moguće je da dva puta angažujemo u isto vreme istog predavača ili uključimo istu grupu studenata. Ostala ograničenja su zadovoljena. [**15**]

Prostor pretrage $N(s)$, gde je s rešenje, sastoji se od svih mogućih rasporeda koji se mogu generisati za lekciju l u različitim periodima t. Od svih rešenja iz $N(s)$ algoritam bira ono kome dodeli najveću ocenu kvaliteta pri čemu se ocena dobija kao suma konflikata za predavače ili studente za svaki period. [**1**]

4.6. Ekspertski sistemi

Solotorevsky sa saradnicima definiše novi jezik RAPS kojim se opisuje opšti problem raspoređivanja, a između ostalog može se primeniti i za opis problema rasporeda časova i to tako što se lekcije smatraju aktivnostima kojima treba da se dodele odgovarajući resursi (period, grupa studenata, profesori, itd.) [**1**] [**16**]

RAPS ima pet tipova pravila:

- Pravila dodele – definišu koja lekcija će biti održana u kom terminu
- Pravila ograničenja – definišu ograničenja koja rešenje mora da zadovolji, i proveravaju se prilikom svake dodele. Dele se na pozitivna (moraju biti zadovoljena) i negativna (opisuju konflikte)
- Pravila izmene – definišu akcije „popravke" rešenja ako dodela aktivnosti narušava pravila ograničenja
- Pravila konteksta – definišu više grupa pravila za različite kontekste
- Pravila prioriteta – dodeljuje se različit prioritet za pojedine lekcije u različitom periodu za različite kontekste

Sistem može raditi u dva moda:

- pohlepan (*greedy*) – u slučaju konflikta pokušava pretragu unazad *(backtrack)*
- ne-pohlepan (*non-greedy*) – koristi pravila lokalne pretrage *(local search)*

4.7. Logičko programiranje

Sistem programiranja logičkih ograničenja *Constraint Logic Programming* (CLP) ima mogućnost da modelira problem pretrage preko mogućnosti da definiše promenljive i njihove domene, kao i ograničenja nad njima. U potrazi za rešenjima, generišu se vrednosti za sve promenljive, imajući u vidu postavljena ograničenja, kako bi se smanjilo polje pretrage gde vrednosti narušavaju ista. Osnovna metoda implementacije je pretraga unazad uz heuristiku koja prepoznaje posledice odluke i konflikte pre nego što do njih dođe. [**1**] [**17**]

Azevedo i *Barahona* definišu CLP jezik DOMLOG baziran na jeziku CHIP. DOMLOG omogućava definisanje konačnih domena promenljivih i korisnicima daje mogućnost da definišu sopstvene heuristike. [**18**] [**19**]

4.8. Genetski algoritmi

Genetski algoritmi su zasnovani na teoriji Darvinove evolucije jedne populacije jedinki pod dejstvom genetskih operatora. „Populacija" mogućih rešenja se održava tako što „najjači" preživljavaju. Svakom rešenju iz prostora dopustivih rešenja X problema dodeljuje se niz konačne dužine koji opisuje njegove atribute i naziva se ***genetski kod*** ovog rešenja. Skup kodova svih dopustivih rešenja iz X čini ***prostor kodiranih rešenja*** $\bar{X}$. Genetski algoritam u jednoj iteraciji vrši pretraživanje u prostoru kodiranih rešenja. Skup trenutnih rešenja u jednoj iteraciji algoritma naziva se ***populacija*** u oznaci $P_n = \{x_1^n, \dots, x_N^n\}$, $x_i^n \in X$, $i = \overline{1, N}$, gde je N zadati broj tačaka populacije i njemu odgovarajući skup ***populacija kodiranih rešenja*** $\bar{P}_n = \{\bar{x}_1^n, \dots, \bar{x}_N^n\}$, $\bar{x}_i^n \in \bar{X}$, $i = \overline{1, N}$. Za svaku tačku $\bar{x}_i^n \in \bar{P}_n$ izračunava se njena funkcija prilagođenosti definisana kao $F: \bar{X} \to \mathbb{R}$. Najčešće se za funkciju prilagođenosti uzima $F(\bar{x}) = -f\big(d(\bar{x})\big)$, gde je f funkcija cilja, a $d(\bar{x})$ dekodirana vrednost od $\bar{x}$.

Opšti oblik genetskog algoritma:

Inicijalizacija:

- Definisati prostor kodiranih rešenja $\bar{X}$ i funkciju prilagođenosti $F(\bar{x})$
- $\bar{P}_1 = \{\bar{x}_1^1, \dots, \bar{x}_N^1\} \subseteq \bar{X}$ — bira se početna populacija
- $f^* = \min\left\{f\left(d(\bar{x}_i^1)\right), i = \overline{1, N}\right\}$ — najbolja vrednost funkcije cilja
- $\bar{x}^* = \arg f^*$ — najbolje rešenje $\bar{x}^*$ uzima vrednost argumenta f^*
- $x^* = d(\bar{x}^*)$ — dekodirano najbolje rešenje

Iterativni korak: $n = 1,2, \dots$

1. Odrediti prilagođenost $F(\bar{x}_i^n)$ za svaku tačku $\bar{x}_i^n \subseteq \bar{P}_n$, $i = \overline{1, N}$
2. Generisati $\bar{P}_{n+1} = \{\bar{x}_1^{n+1}, \dots, \bar{x}_N^{n+1}\} \subseteq \bar{X}$ primenom genetskih operatora na slučajno odabrane tačke iz $\bar{P}_n$ koje imaju veću prilagođenost.
3. $f_{min} = \min\left\{f\left(d(\bar{x}_i^{n+1})\right), i = \overline{1, N}\right\}$, ako $f_{min} < f^*$, tada $f^* = f_{min}$ i $x^* = \arg f_{min}$

Kraj:

- Posle dovoljnog broja generacija (iteracija) procedura se zaustavlja i $d(\bar{x}^*)$ se uzima za aproksimaciju rešenja.

U genetskom algoritmu deluju ***operatori selekcije***, ***ukrštanja*** i ***mutacije***. Operatori selekcije, na osnovu prilagođenosti jedinke, biraju jedinke koje će učestvovati u kreiranju novih jedinki, primenom operatora ukrštanja i mutacije. Operatori ukrštanja se definišu kao postupak u kome se slučajno uzajamno razmenjuju delovi kodova dva rešenja iz $\bar{X}$ (***roditelji***) i tako dobijaju kodovi dva nova rešenja (***potomci***). Ukrštanje se obavlja sa unapred zadatom verovatnoćom ukrštanja i tako dobijeni potomci nose „dobre“ osobine svojih roditelja. Najčešće se primenjuje jednopozicioni operator ukrštanja, ali operator može biti definisan i u odnosu na više tačaka ukrštanja. Operator mutacije sa zadatom verovatnoćom mutacije (reda veličine 10^{-3}) vrši promenu sadržaja koda slučajnom zamenom pojedinih simbola drugim simbolima i ima sekundarnu ulogu u formiranju novih „jedinki“. [**7**]

Rasporedi sa najmanjom upotrebnom cenom se biraju i služe kao baza za sledeću „generaciju“, ovako poboljšavajući globalni kvalitet i održavaju raznolikost rešenja. Rasporedi se predstavljaju nizovima bitova koji predstavljaju kada i gde će se održavati koje predavanje. Dva rasporeda se odabiraju, njima odgovarajući nizovi bitova se ukrštaju (polove i spajaju), generišući nove rasporede.

Corne, *Fang* i *Mellish* uvode inteligentan operator mutacije koji daje bolje rezultate od jednostavnog ukrštanja „roditelja“, tzv. GATT. Njihov sistem se koristi za izradu rasporeda ispita na Univerzitetu u Edinburgu. [**1**] [**20**]

Erben i *Keppler* su primenili navedenu tehniku za generisanje zadovoljavajućih rasporeda časova sa velikim brojem ograničenja na Univerzitetu primenjenih nauka u Konstancu u Nemačkoj. [**21**]

4.9. Primenjene metode i praktični rezultati

Neke od navedenih metoda su našle praktičnu primenu.

Daskalaki, *Birbas* i *Housos* u svom radu iznose rezultate rasporeda časova za Fakultet elektronskog i kompjuterskog inženjerstva Univerziteta u Patrasu rešavanog metodom linearnog celobrojnog programiranja. Korišćen je rešavač CPLEX 5.1 na radnoj stanici HP J7000. [**22**]

Modelu su zadata tri problema prema veličini ulaznih podataka:

Broj predmeta	Broj vežbi	Potrebno termina	Broj učionica	Broj laboratorija	Broj jednačina	Broj nepoznatih	Vreme [min]
25	8	139	3	6	7543	4100	2,5
47	19	187	4	10	12734	13527	18,5
92	27	326	6	12	17159	19295	95,0

Murray, *Muller* i *Rudova* rešavaju problem rasporeda časova u Purdue Univerzitetu, Indijana SAD i postignute rezultate sumiraju u svom radu. Na pomenutom univerzitetu obrađivan je raspored za 27881 studenata gde je u proseku svaki pohađao 3,15 predmeta (kurseva). Za predmete za koje postoji potreba specijalne nastave u laboratorijama svaki student je pohađao u proseku 1,14 predmeta. [**23**]

Autori posmatraju četiri različita problema:
- LLR (*Large lecture problem*) – problem lekcija koje zahtevaju više od jednog termina
- LAB (*Laboratory problem*) – problem zauzeća laboratorija
- D1 (*I different departmental problem*) – specifičan problem jedne katedre
- D2 (*II different departmental problem*) – specifičan problem druge katedre

Svaki problem prvo rešavaju pojedinačno u zadatom roku od 30 minuta, a zatim se u završnoj zajedničkoj obradi sva četiri problema zadaje vremenski okvir od 120 minuta.

Korišćen je rešavač koji bazira na tehnici zadovoljenja ograničenja i optimizaciji, rezultati su izneti u sledećoj tabeli:

Broj predmeta	Broj učionica	Vreme [min]	Nezadovoljena ograničenja		
			studenata	termin [%]	učionica [%]
804	55	5,20±4,90	1207	15,3	11,1
440	25	20,80±3,60	11	32,6	23,8
69	6	0,08±0,07	3	18,4	0,0
443	36	5,30±3,40	14	12,4	24,2

He Yan i *Song-Nian Yu* u svom radu daju eksperimentalne rezultate za metod simuliranog kaljenja primenjen na sledećim podacima:

Broj predavanja	Broj predavača	Broj termina	Broj učionica
500	275	20	250

I dobijeni su sledeći eksperimentalni podaci:

Koeficijent hlađenja	Broj iteracija	Početna temperatura	Vreme izvršenja [s]	Narušena ograničenja	Nezadovoljene osobine	Dostignuta temperatura
0,99	100	10000	81,153	2	65	2065
			98,893	3	41	3041
			81,136	2	46	2046
	200	20000	179,278	1	16	1016
			200,549	1	14	1014
			174,171	2	7	2007
	300	3000	298,409	0	2	2
			269,989	0	4	4
			269,085	0	0	0

Program je puštan po tri puta sa istim argumentima, a zatim i različitim, pri čemu je u relativno kratkom roku dostizao skoro optimalno rešenje, ali nijednom nije generisao isto rešenje što je posledica slučajnog izbora suseda tokom pretrage. [**14**]

5. Predloženi matematički model problema rasporeda časova

Definišemo osnovna ograničenja problema

5.1. Prostor rešenja

Osnovnu strukturu za matematički model koji posmatramo čine parametri sledećih šest skupova:

- Skup raspoloživih dana u kojima se odvija nastava od ponedeljka do petka, označen sa $I = \{1,2,3,4,5\}$, $|I| = 5$
- Skup radnog vremena (termina) od 08:00 do 21:00 pri čemu se nastava odvija u trajanju od 45 minuta sa pauzom od 15 minuta i taj skup označavamo sa $J = \{1,2,\dots,13\}$, $|J| = 13$
- Skup grupa studenata za koje se izrađuje raspored, npr. studenti prve godine, druge i dr. označavamo sa K pri čemu je $K = \{grupa_1, grupa_2, \dots, grupa_k\}$, $|K| = k$
- Skup predavača za koje se izrađuje raspored, označavamo sa L pri čemu je $L = \{predavač_1, predavač_2, \dots, predavač_l\}$, $|L| = l$
- Skup kurseva (predmeta) za grupu studenata u oznaci M pri čemu je $M = \{predmet_1, predmet_2, \dots, predmet_m\}$, $|M| = m$. Ako predmet ima i teorijski i praktični deo, navode se odvojeno. Predmet jednog predavača za dve različite grupe navodi se dva puta.
- Skup raspoloživih učionica u oznaci N pri čemu je $N = \{učionica_1, učionica_2, \dots, učionica_n\}$, $|N| = n$

U cilju kompaktnijeg i preglednijeg zapisa uvodimo pomoćne skupove:

- K_l – skup studenata kojima predavač l drži bar jedan kurs, $l \in L$
- K_m – skup studenata koji pohađaju kurs m, $m \in M$
- K^{sp} – skup spojenih grupa studenata
- K_k^{sp} – skup spojenih grupa studenata kojima pripada grupa k, $k \in K$
- L_k – skup predavača koji drže bar jedan kurs grupi studenata k, $k \in K$
- L_{km} – skup predavača koji drže kurs m grupi studenata k, $k \in K$, $m \in M$
- M_{kl} – skup kurseva predavača l za grupu studenata k, $k \in K$, $l \in L$
- M_{kln} – skup kurseva predavača l za grupu studenata k koji se održavaju u učionici n, $k \in K$, $l \in L$, $n \in N$
- M_{kl}^{obav} – skup obaveznih kurseva predavača l za grupu studenata k, $k \in K$, $l \in L$
- M_{kl}^{izb} – skup izbornih kurseva predavača l za grupu studenata k, $k \in K$, $l \in L$
- M^{pra} – skup vežbi sa praktičnom nastavom
- N_m – skup učionica raspoloživih za kurs m, $m \in M$

Promenljive odluke

Model koristi dva različita skupa binarnih promenljivih. Prvi se naziva osnovni skup promenljivih u oznaci x_{ijklmn} pri čemu je:

$$\forall i \in I, \forall j \in J, \forall k \in K, \forall l \in L_k, \forall m \in M_{kl}, \forall n \in N_m$$

$$x_{ijklmn} = \begin{cases} 1 & \text{ako kurs } m \text{ drži predavač } l \text{ grupi studenata } k \text{ u terminu } j \text{ dana } i \text{ u učionici } n \\ 0 & \text{u suprotnom} \end{cases}$$

$$|x_{ijklmn}| = |I| * |J| * |K| * |L| * |M| * |N|$$

Drugi skup čine pomoćne promenljive z_{ijk} pri čemu je:

$$\forall i \in I, \forall j \in J, \forall k \in K \setminus \{master\}$$

$$z_{ijk} = \begin{cases} 1 & \text{ako je kurs grupe } k \text{ preklopljen sa kursom viših godina u terminu } j \text{ dana } i \\ 0 & \text{u suprotnom} \end{cases}$$

$$|z_{ijk}| = |I| * |J| * |K|$$

Promenljiva z_{ijk} se koristi za rešavanje rasporeda predmeta za studente koji su preneli neki ispit. [**24**]

5.2. Konstrukcija matematičkog modela

Konstrukcija matematičkog modela se vrši prema osnovnim kriterijumima koje treba zadovoljiti u rasporedu koje definišemo preko sledećih ograničenja:

Preklapanje kurseva

U okviru jedne grupe studenata ne sme biti preklapanja predmeta ni profesora u jednom terminu. Matematički zapis ovog ograničenja prema gore definisanim skupovima je:

$(\forall i \in I)(\forall j \in J)(\forall k \in K)$

$$\sum_{\substack{m \in M_k \\ alt(m)=0}} \sum_{l \in L_{km}} \sum_{n \in N_m} x_{ijklmn} \leq 1 \qquad (5.2.1)$$

gde je $alt(m)$ funkcija alternativnih kurseva, od kojih se bira samo jedan.

Ako student u okviru jednog semestra mora odabrati jedan od izbornih predmeta $\{m_1, m_2\}$ imamo:

$(\forall i \in I)(\forall j \in J)(\forall k \in K)$

$$\sum_{l \in L_{km_1}} \sum_{n \in N_{m_1}} x_{ijklm_1n} + \sum_{l \in L_{km_2}} \sum_{n \in N_{m_2}} x_{ijklm_2n} \leq 1 \qquad (5.2.1')$$

Preklapanje predavača

U datom terminu, predavač može držati jednoj grupi ili jednoj spojenoj grupi jedan kurs:

$(\forall i \in I)(\forall j \in J)(\forall l \in L)$

$$\sum_{k \in K_l} \sum_{m \in M_{kl}} \sum_{n \in N_m} x_{ijklmn} \leq 1 \qquad (5.2.2)$$

Preklapanje učionica

U jednoj učionici u datom terminu može se održavati samo jedan kurs:

$(\forall i \in I)(\forall j \in J)(\forall n \in N)$

$$\sum_{k \in K} \sum_{l \in L_k} \sum_{m \in M_{kln}} x_{ijklmn} \leq 1 \qquad (5.2.3)$$

Preklapanje grupa

Određena predavanja se izvode za više grupa istovremeno. Treba obezbediti da te grupe nemaju obaveza u tom terminu, ili ako je neka od grupa zauzeta ne dozvoliti dodelu spojene grupe:

$(\forall i \in I)(\forall j \in J)(\forall k \in K)$

$$\sum_{m \in M_k} \sum_{l \in L_{km}} \sum_{n \in N_m} x_{ijklmn} + \sum_{s \in K_k^{sp}} \sum_{m \in M_s} \sum_{l \in L_{sm}} \sum_{n \in N_m} x_{ijslmn} \leq 1 \qquad (5.2.4)$$

Obavezan broj časova nedeljno

Da bi upisao narednu godinu svaki student treba da obezbedi potreban broj bodova što je u direktnoj vezi sa brojem časova koje treba da pohađa u nedelji. U ovom slučaju $c(k)$ je funkcija koja daje ukupan broj časova za kurseve koje treba da pohađa grupa studenata k.

$(\forall k \in K)$

$$\sum_{i \in I} \sum_{j \in J} \sum_{l \in L_k} \sum_{m \in M_{kl}} \sum_{n \in N_m} x_{ijklmn} = c(k) \qquad (5.2.5)$$

Da bi osigurali da su svi kursevi zastupljeni u rasporedu, mora biti dodato sledeće ograničenje:

$(\forall m \in M)$

$$\sum_{i \in I} \sum_{j \in J} \sum_{k \in K_m} \sum_{l \in L_{km}} \sum_{n \in N_m} x_{ijklmn} = h(m) \qquad (5.2.5')$$

gde je $h(m)$ funkcija koja daje broj časova zahtevanih za kurs m

Kontinuitet termina

Svaki kurs treba predstaviti u kontinualnom periodu da bi zadovoljio broj svojih kredita (broj časova predavanja i broj časova vežbi). Ovo ograničenje se postavlja za predavanja za koja je potrebno više od jednog termina ($h(m) > 1$). Za ispunjenje ovog zahteva postavlja se sledeće ograničenje, sa izuzećem prvog sabirkaz sume za $j = 1$.

$(\forall i \in I)(\forall j \in J)(\forall k \in K)(\forall l \in L_k)(\forall m \in M_{kl})(\forall n \in N_m)$

$$-(h(m) - 1)x_{ij-1klmn} + (h(m) - 1)x_{ijklmn} - \sum_{\substack{t=1}}^{\substack{t+j \le |J| \\ t < h(m)}} x_{ij+tklmn} \le 0 \qquad (5.2.6)$$

Kontinuitet u korišćenju učionice

Svi termini jednog kursa u jednom danu treba da se odvijaju u jednoj učionici:

$(\forall j \in J)(\forall m \in M)(\forall n \in N_m)$

$$h(m) * \sum_{i \in I} \sum_{k \in K_m} \sum_{l \in L_{km}} x_{ijklmn} \le \sum_{i \in I} \sum_{p \in J} \sum_{k \in K_m} \sum_{l \in L_{km}} x_{ipklmn} \qquad (5.2.7)$$

Izostavljanje određenih termina za predefinisane aktivnosti

Ako iz nekog razloga treba ostaviti određene termine slobodne, radi održavanja nekih aktivnosti (seminari i dr.), tada važi sledeće:

$(\forall k \in K)(\forall p \in \{12,13\})$

$$\sum_{l \in L_k} \sum_{m \in M_{kl}} \sum_{n \in N_m} x_{5pklmn} = 0 \qquad (5.2.8)$$

5.3. Specifična ograničenja

Sledeća ograničenja nisu obavezna za kreiranje rasporeda.

Teorija pre prakse

Ako je za kurs predviđeno održavanje i teorijske i praktične nastave može se predvideti održavanje teorijskog dela pre praktičnog rada:

$(\forall m \in M^{pra})$

$$\sum_{i\in I}\sum_{j\in J}\sum_{k\in K_m}\sum_{l\in L_{km}}\sum_{n\in N_m} i * x_{ijklmn}/h(m) > \sum_{i\in I}\sum_{j\in J}\sum_{k\in K_{t(m)}}\sum_{n\in N_{t(m)}} i * x_{ijklt(m)n}/h\big(t(m)\big) \tag{5.3.1}$$

gde je $t(m)$ funkcija koja daje indeks teorijskog kursa iz skupa M za praktični deo m.

Vremenska ograničenja

Za grupe studenata master studija mogu biti predviđeni termini posle $15{:}00$ (termin 8) zbog eventualnih radnih obaveza:

$(\forall k \in K^{master})$

$$\sum_{i\in I}\sum_{\substack{j\in J\\ j<8}}\sum_{l\in L_k}\sum_{m\in M_{kl}}\sum_{n\in N_m} x_{ijklmn} = 0 \tag{5.3.2}$$

Kursevi za studente koji su preneli neki ispit

Ovo ograničenje obezbeđuje da se ne preklope obavezni kursevi i kursevi prenetih ispita, u slučaja da se ovi kursevi preklope promenljiva z_{ijk} nepovoljno utiče na funkciju cilja.

$(\forall i \in I)(\forall j \in J)(\forall k \in K \setminus \{studenti\ završnih\ godina\})$

$$\sum_{l\in L_k}\sum_{m\in M_{kl}^{obav}}\sum_{n\in N_m} x_{ijklmn} + \sum_{l\in L_{k+1}}\sum_{m\in M_{k+1,l}^{obav}}\sum_{n\in N_m} x_{ijk+1lmn} \le 1 + z_{ijk} \tag{5.3.3}$$

Studenti završnih godina su isključeni jer ne upisuju sledeći semestar.

5.4. Funkcija cilja

Funkcija cilja ovog modela opisuje nezadovoljstvo svih učesnika u rasporedu pri čemu se ukupno nezadovoljstvo predstavlja kao zbir pojedinačnih:

- nezadovoljstvo studenata
- nezadovoljstvo predavača (redovnih, vanrednih, docenata, asistenata, saradnika u nastavi)
- nezadovoljstvo studenata koji prenose ispit

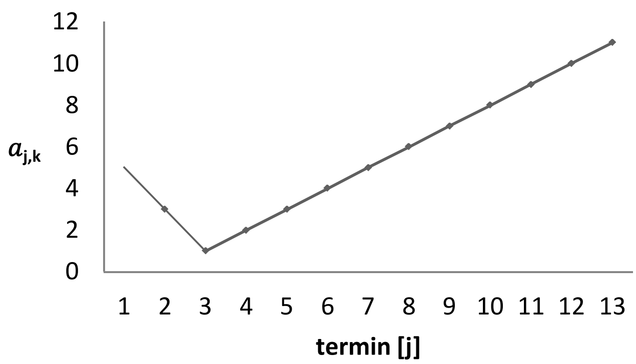
Za sve studente važi da njihovo nezadovoljstvo raste ako imaju velike pauze između predavanja ili vežbi. Nezadovoljstvo studenata osnovnih studija raste ako su im dodeljeni časovi u večernjim terminima, i suprotno, za studente master studija nisu poželjni jutarnji termini zbog mogućeg studiranja uz zaposlenje. Za redovne profesore koeficijent „nezadovoljstva" raste ako se ne ispoštuju dani ili termini kada žele da predaju, slično i za vanredne, takođe treba izbeći velike pauze između predavanja u jednom danu i slično.

Gore navedena funkcija ima sledeću matematičku formu:

$$\min\left(\sum_{j\in J}\sum_{k\in K} a_{jk}\left(\sum_{i\in I}\sum_{l\in L_k}\sum_{m\in M_{kl}}\sum_{n\in N_m} x_{ijklmn}\right)+\sum_{i\in I}\sum_{j\in J}\sum_{l\in L} b_{ijl}\left(\sum_{k\in K_l}\sum_{m\in M_{kl}}\sum_{n\in N_m} x_{ijklmn}\right)\right.$$
$$\left.+\,100\sum_{i\in I}\sum_{j\in J}\sum_{k\in K} z_{ijk}\right)$$

Koeficijent a_{jk} se definiše kao funkcija nezadovoljstva studenata $a\colon J \times K \to \mathbb{N}$.

Na Slici 1 dat je primer u kome se preferira treći termin od 10h do 11h.



Slika 2 Koeficijent nezadovoljstva studenata grupe $\boldsymbol{k}$

Koeficijent nezadovoljstva profesora definisan je sa:

$$b_{ijl} = \begin{cases} 100 & \text{ako profesor } l \text{ ne želi da predaje dana } i \text{ u terminu } j \\ 0 & \text{u suprotnom} \end{cases}$$

Vrednost b_{ijl} je proizvoljno izabrana tako da favorizuje želje profesora u odnosu na želje studenata.

5.5. Smanjenje veličine modela

Bez izazivanja bitnijih promena u modelu možemo ignorisati indeks učionice u promenljivoj x. Da bismo izbegli preklapanja učionica uvodimo sledeća ograničenja:

$$(\forall i \in I)(\forall j \in J) \sum_{k \in K} \sum_{l \in L_k} \sum_{m \in M_{kl}} x_{ijklm} \leq 20$$

$$(\forall i \in I)(\forall j \in J) \sum_{k \in K} \sum_{l \in L_k} \sum_{m \in M_{kl}} n(m) * x_{ijklm} \leq 111$$

$$(\forall i \in I)(\forall j \in J) \sum_{k \in K} \sum_{l \in L_k} \sum_{\substack{m \in M_{kl} \\ n(m)=0}} x_{ijklm} \leq 5$$

Prvo ograničenje ne dozvoljava dodelu više od 20 učionica. Drugo reguliše pridruživanje učionica kursu na osnovu njihovog kapaciteta, ako na raspolaganju imamo 13 učionica većeg kapaciteta i 7 učionica za manje grupe studenata. Treće ograničenje dozvoljava dodelu ne više od pet učionica opremljenih računarima. Prema kapacitetu učionice i zahtevima kursa funkcija $n(m)$ se definiše na sledeći način:

$$n(m) = \begin{cases} 0 & \text{kurs } m \text{ se mora držati u učionici sa računarima} \\ 1 & \text{kurs } m \text{ se može držati u manjoj učionici} \\ 8 & \text{kurs } m \text{ se može držati samo u većoj učionici} \end{cases}$$

5.6. Uticaj formulacije modela na performanse

Pažljivo postavljena formulacija mešovitog celobrojnog linearnog programiranja može dovesti do rešenja modela u razumnom vremenskom roku, dok u suprotnom može problem učiniti praktično nerešivim.

Smanjivanjem broja ekvivalentnih rešenja (simetrije problema) smanjuje se broj koraka (grana) u osnovnom procesu rešavanja problema primenom algoritma grananja i odsecanja. Jedan od načina za izbegavanje simetrije je formulisanje problema preko obrazaca umesto korišćenja promenljivih koje označavaju dodele.

Posmatramo primer u kome imamo 25 kamiona podeljenih u 5 klasa prema nosivosti: 40, 50, 60, 80, 100 i njima odgovarajućih cena korišćenja: 40, 50, 60, 76, 98, respektivno. Treba prebaciti 20 nedeljivih paketa sledećih težina: 10, 10, 20, 20, 20, 30, 30, 40, 40, 40, 40, 50, 50, 50, 50, 60, 60, 70, 70, 80. Ako sa a_i označene težine paketa i, a sa n_j i c_j nosivost i cenu kamiona j problem možemo formulisati na sledeći način:

$$\min \sum_{j=1}^{25} c_j y_j \tag{5.5.1}$$

pri ograničenjima:

$$\sum_{j=1}^{25} x_{ij} = 1, \qquad i \in \{1, \dots, 20\} \tag{5.5.2}$$

$$\sum_{i=1}^{20} a_i x_{ij} \leq n_j y_j, \qquad j \in \{1, \dots, 25\} \tag{5.5.3}$$

primenjive:

$$x_{ij} = \begin{cases} 1 & \text{ako teret } i \text{ prevozi kamion } j \\ 0 & \text{u suprotnom} \end{cases}, \qquad i \in \{1, \dots, 20\}, j \in \{1, \dots, 25\} \tag{5.5.4}$$

$$y_j = \begin{cases} 1 & \text{ako je kamion } j \text{ zauzet} \\ 0 & \text{u suprotnom} \end{cases}, \qquad j \in \{1, \dots, 25\} \tag{5.5.5}$$

Prvi skup ograničenja (5.5.2) predstavlja nedeljivost paketa, tj. svakom paketu se dodeljuje tačno jedan kamion. Drugi skup (5.5.3) uzima u obzir kapacitet kamiona i težinu paketa.

U navedenom primeru možemo uočiti sledeće ekvivalentne dodele:

- kako postoje klase kamiona, dodela različitih kamiona iz iste klase jednom konkretnom paketu ne dovodi do promene vrednosti funkcija cilja, takođe
- kako postoje paketi istih težina, dodela paketa konkretne težine jednom kamionu se ne razlikuje od dodele drugog paketa iste težine tom kamionu

U stablu pretrage, navedeno generiše grane koje bespotrebno proširuju prostor pretrage, ali kako one predstavljaju iste LP podprobleme, obilaženje ovih grana ne dovodi do optimizacije funkcije cilja.

Uvođenjem dodatnih ograničenja npr. dodeljivanje prioriteta kamionima u jednoj klasi zabranjuje se dodela kamiona $j+1$ pre kamiona j, i ovako se smanjuje broj grana koje opisuju zamenu kamiona u istoj klasi.

Problem simetrije možemo izbeći modeliranjem problema preko obrazaca. Za svaki od tipova kamiona generišemo sve dopustive obrasce (kombinacije paketa) i biramo koje koristimo i koliko puta. Vektor obrasca je vektor čija n-ta koordinata predstavlja broj „spakovanih“ paketa n-te težine. Moguće kombinacije paketa za kamion nosivosti 40 su (40), $(30, 10)$, $(20, 20)$, $(20, 10, 10)$ i $(10, 10, 10, 10)$. One se mogu prikazati kao njima odgovarajući vektori obrazaca $(0, 0, 0, 1, 0, 0, 0, 0)$, $(1, 0, 1, 0, 0, 0, 0, 0)$, $(0, 2, 0, 0, 0, 0, 0, 0)$, $(2, 1, 0, 0, 0, 0, 0, 0)$ i $(4, 0, 0, 0, 0, 0, 0, 0)$, respektivno. Označimo ove vektore sa d_{kl}, gde je k klasa kamiona, a l indeks obrasca. Pretpostavimo da je L_k obrazaca popunjavanja za kamion klase k ($k \in \{1,2,3,4,5\}$), potrebno jer generisati samo obrasce za potpuno iskorišćavanje nosivosti kamiona. Sa b označimo vektor $(2, 3, 2, 4, 4, 2, 2, 1)$ čije koordinate predstavljaju broj zahtevanih paketa određene težine. Sada model ima sledeći oblik:

$$\min \sum_{k=1}^{5} g_k \sum_{l=1}^{L_k} v_{kl} \tag{5.5.6}$$

pri ograničenjima:

$$\sum_{k=1}^{5} \sum_{l=1}^{L_k} d_{kl} v_{kl} \geq b \tag{5.5.7}$$

$$\sum_{l=1}^{L_k} v_{kl} \leq 5, \qquad k \in \{1,2,3,4,5\} \tag{5.5.8}$$

promenljive:

$$v_{kl} \in \mathbb{N}_0, \qquad k \in \{1,2,3,4,5\}, l \in \{1, \ldots, L_k\} \tag{5.5.9}$$

gde je sa g_k označena cena upotrebe kamiona tipa k. Promenljiva v_{kl} je broj primenjenih obrazaca l za klasu kamiona k. Prvi skup ograničenja (5.5.7) obezbeđuje da bude rezervisan kapacitet za zahtevane pakete, a nejednakost „veće ili jednako“ se koristi jer smo generisali samo obrasce za pune kamione. Iako rezervisan kapacitet može da prevaziđe potrebu, rešenje je i dalje validno, u tom slučaju neki kamioni neće biti puni. Drugi skup ograničenja (5.5.8) limitira broj kamiona tipa k, u našem slučaju je po 5.

Prilikom procesa grananja nad vektorom obrazaca svaka grana predstavlja upotrebu novog obrasca, iako vrednost funkcije cilja može ostati ista. MILP podproblemi se razlikuju, tako da proces pretrage napreduje u svakom koraku. Takođe modeliranje preko obrazaca ima korisnu osobinu da je dopustiv skup LP dosta bliže MILP dopustivom skupu.

Nedostatak metode modeliranja preko obrazaca je što broj obrazaca rapidno raste sa brojem kombinacija. Kako bi se ovo izbeglo koristi se modeliranje preko kolona, pri čemu se kolone obrazaca ne generišu sve odjednom, nego po potrebi u toku procesa rešavanja. Ovakav algoritam generisanja kolona naziva se i algoritam dekompozicije.

6. Primena modela

U cilju testiranja predloženog modela posmatramo problem izrade rasporeda časova za Matematički fakultet, Univerziteta u Beogradu.

6.1. Podaci

Skupovi raspoloživih radnih dana i radnog vremena (termina) I i J preuzimaju se kako su opisani u 5.1, a za formiranje ostalih pomoćnih skupova preuzeti su neoficijelni podaci sa sajta Matematičkog fakulteta i moguće su nenamerne greške.

6.1.1. Grupe studenata

U školsku godinu 2011/2012 na osnovne studije upisano je 395 studenata i to 250 na matematički smer, 120 na informatički smer i 25 na astronomiju i astrofiziku, kao i 205 studenata na master studije za sve smerove. Po godinama, za navedene smerove, nastava se odvija po sledećim grupama:

1I1	I godina, Informatika, I grupa, podgrupa A, B, C
1I2	I godina, Informatika, II grupa, podgrupa A, B, C
2I1	II godina, Informatika, I grupa, podgrupa A, B
2I2	II godina, Informatika, II grupa, podgrupa A, B
3I	III godina, Informatika
1O1	I godina, Matematika, I grupa, podgrupa A, B
1O2	I godina, Matematika, II grupa, podgrupa A, B
1O3	I godina, Matematika, III grupa, podgrupa A, B
1O4	I godina, Matematika, IV grupa, podgrupa A, B
2L	II godina, Matematika, profesor matematike i računarstva, podgrupa A, B
2R	II godina, Matematika, računarstvo i informatika, podgrupa A, B
2MNV	II godina, Matematika, teorijska, primenjena, statistika i dr. podgrupa A, B
3M	III godina, Matematika, teorijska matematika i primena
3R	III godina, Matematika, računarstvo i informatika
3N	III godina, Matematika, primenjena matematika
3L	III godina, Matematika, profesor matematike i računarstva
3V	III godina, Matematika, statistika, aktuarska i finansijska matematika
4M	IV godina, Matematika, teorijska matematika i primena
4R	IV godina, Matematika, računarstvo i informatika
4N	IV godina, Matematika, primenjena matematika
4L	IV godina, Matematika, profesor matematike i računarstva
4V	IV godina, Matematika, statistika, aktuarska i finansijska matematika
4I	IV godina, Informatika, I godina master studija
5I	V godina, Informatika, II godina master studija
5ML	V godina, Matematika, master studije profesor matematike i računarstva
5MM	V godina, Matematika, master studije teorijska matematika i primena
5MN	V godina, Matematika, master studije primenjena matematika
5MR	V godina, Matematika, master studije računarstvo i informatika
5MV	V godina, Matematika, master studije statistika, aktuarska i finansijska matematika
1A	I godina, Astronomija sa astrofizikom
2A	II-godina, Astronomija sa astrofizikom
3A	III-godina, Astronomija sa astrofizikom
4A	IV godina, Astronomija sa astrofizikom

Ovako definisane grupe su elementi skupa označenog sa K u postavljenom modelu.

6.1.2. Predavači

Ukupan broj predavača za jesenji semestar je 130 od toga 24 redovnih, 18 vanrednih, 18 docenta, 36 asistenata i 34 saradnika u nastavi. (Prilog 1)

Tako imamo sledeće skupove predavača za koje se izrađuje raspored:

- $L_r = \{profesor_1^r, profesor_2^r, \dots\}$ skup redovnih profesora $|L_r| = 24$
- $L_v = \{profesor_1^v, profesor_2^v, \dots\}$ skup vanrednih profesora $|L_v| = 18$
- $L_d = \{profesor_1^d, profesor_2^d, \dots\}$ skup docenata $|L_d| = 18$
- $L_a = \{asistent_1, asistent_2, \dots\}$ skup asistenata $|L_a| = 36$
- $L_s = \{saradnik_1, saradnik_2, \dots\}$ skup saradnika u nastavi $|L_s| = 34$

6.1.3. Predmeti po grupama

Za jesenji semestar predviđeno je 64 kursa (predmeta) za tri godine osnovnih studija smera Informatike i četiri godine smera Matematike i Astronomije nejednako raspoređenih po godinama, kao i 21 predmeta za master studije.

Sledeća tabela prikazuje predviđene predmete za studente prve godine informatičkog smera, pri čemu imamo 4 obavezna i 1 izborni predmet u prvom semestru.

	Predmet	fond časova
1	Programiranje I	3+3+0
2	Uvod u organizaciju računara	3+2+0
3	Diskretne strukture I	3+2+0
4	Linearna algebra i analitička geometrija	3+2+0
5	Izborni predmet	2+1+0

Za svaki predmet u rasporedu se određuje poseban termin za predavanje, vežbe i za praktikum (ako je predviđen). Predavanja su zajednička za grupu, a vežbe i praktikum se održavaju posebno za podgrupu, tako da svaki predmet u okviru grupe posmatramo razdvojeno predavanja-vežbe-praktikum prema sledećem obrascu:

- Predmet-predavanje
- Predmet-vežbe-podgrupa A
- Predmet-vežbe-podgrupa B
- Predmet-vežbe-podgrupa C
- Predmet-praktikum-podgrupa A
- Predmet-praktikum-podgrupa B
- Predmet-praktikum-podgrupa C

Za predmete za koje je predviđeno 4 časa predavanja ili vežbi nedeljno najčešće se predviđaju po dva termina od dva časa, ali nije obavezno.

Prema definisanim grupama predavača u 6.1.2 ostvaruju se sledeće veze:

- predavanja izvodi redovan profesor, vanredan ili docent
- vežbe i praktikume mogu da izvode docenti, asistenti i saradnici, ali isto tako i pojedini profesori koji to za svoja predavanja izričito zahtevaju

Sledeća tabele daju predmete koje treba uvrstiti u raspored:

I godina, Informatika, 1 grupa, podgrupa A, B, C

Predmet	šifra	fond časova pred	vežbe	prakt	lab	Predavač
Programiranje I	1I1PR1	3				Predrag Janičić
	1I1PR1VA		3			Danijela Petrović
	1I1PR1VB		3			Aleksandar Kartelj
	1I1PR1VC		3			Jelena Dunjić
Uvod u organizaciju računara	1I1UOR	3				Nenad Mitić
	1I1UORVA		2			Anđelka Zečević
	1I1UORVB		2			Danijela Petrović
	1I1UORVC		2			Anđelka Zečević
Diskretne strukture I	1I1DS1	3				Aleksandar Jovanović
	1I1DS1VA		2			Slavko Moconja
	1I1DS1VB		2			Slavko Moconja
	1I1DS1VC		2			Slavko Moconja
Linearna algebra i analitička geometrija	1I1LAGVA	3				Miroslava Antić
	1I1LAGVA		2			Miloš Antić
	1I1LAGVB		2			Miloš Antić
	1I1LAGVC		2			Miloš Antić
Izborni predmet (kultura komunikacija)	1I1IZB	2				Zoran Marković
	1I1IZBVA		1			Marko Radovanović
	1I1IZBVB		1			Marko Radovanović
	1I1IZBVC		1			Marko Radovanović

I godina, Informatika, 2 grupa, podgrupa A, B, C

Predmet	šifra	fond časova pred	vežbe	prakt	lab	Predavač
Programiranje I	1I2PR1	3				Filip Marić
	1I2PR1VA		3			Milena Vujoševic-Janičić
	1I2PR1VB		3			Milena Vujoševic-Janičić
	1I2PR1VC		3			Milena Vujoševic-Janičić
Uvod u organizaciju računara	1I2UOR	3				Zorica Stanimirović
	1I2UORVA		2			Danijela Petrović
	1I2UORVB		2			Anđelka Zečević
	1I2UORVC		2			Danijela Petrović
Diskretne strukture I	1I2DS1	3				Žarko Mijajlović
	1I2DS1VA		2			Maja Roslavcev
	1I2DS1VB		2			Maja Roslavcev
	1I2DS1VC		2			Maja Roslavcev
Linearna algebra i analitička geometrija	1I2LAGVA	3				Predrag Tanović
	1I2LAGVA		2			Sonja Telebaković
	1I2LAGVB		2			Sonja Telebaković
	1I2LAGVC		2			Sonja Telebaković
Izborni predmet (kultura komunikacija)	1I2IZB	2				Zoran Marković

	1I2IZBVA	1	Marko Radovanović
	1I2IZBVB	1	Marko Radovanović
	1I2IZBVC	1	Marko Radovanović

Ukupno je potrebno rezervisati 44 termina u nedelji za ovu grupu,odnosno 88 za I godinu Informatike.

Za I godinu Matematike, opšti smer imamo 4 grupe sa po 2 podgrupe studenata sa sledećim predmetima:

I godina, Matematika, I grupa, podgrupa A, B

Predmet	šifra	fond časova			lab	Predavač
		pred	vežbe	prakt		
Analiza IA	1O1AN1	2+2				Darko Milinković
	1O1AN1VA		2+2			Jovana Đuretić
	1O1AN1VB		2+2			Jovana Đuretić
	1O1AN1PA			1		Milan Perić
	1O1AN1PB			1		Nikola Mutavdžić
Linearna algebra A	1O1LAA	3				Žarko Mijajlović
	1O1LAAVA		3			Tanja Stojadinović
	1O1LAAVB		3			Tanja Stojadinović
Uvod u matematičku logiku	1O1UML	2				Milan Božić
	1O1UMLVA		2			Slavko Moconja
	1O1UMLVB		2			Slavko Moconja
Programiranje I	1O1PR1	2				Gordana Pavlović-Lažetić
	1O1PR1VA		2			Jovana Kovačević
	1O1PR1VB		2			Jovana Kovačević
	1O1PR1PA			1		Jovana Kovačević
	1O1PR1PB			1		Jovana Kovačević
Engleski jezik I	1O1EN1	2				Irena Pavlović

Ukupno je potrebno rezervisati 156 termina u nedelji.

Analogno ovome za sve smerove i godine za osnovne studije imamo 713 potrebnih termina za nedelju dana.

Prema datom predlogu matematičkog modela definiše se skup predmeta u oznaci

$M = \{predmet_1, predmet_2, \dots\}$

Date tabele mogu da se iskoriste za formiranje i rad postavljenog modela i odnose se na skup $M_{k,l}$, tj. opisuju vezu između para (grupa, predavač) i predmeta koji taj predavač drži toj grupi. Ako se tabele prošire kolonom učionica za održavanje navedenog predmeta dolazi se do skupa $M_{k,l,n}$.

U Prilogu 2 date su predefinisane veze za svako predavanje (vežbe ili praktikum).

6.1.4. Raspoložive učionice

Predavanja i vežbe se održavaju na dve lokacije u prostorijama u ***Jagićevoj ulici br. 5*** i ***Studentskom trgu 16***, i to u 20 učionica. U Jagićevoj predavanja se drže u 4 učionice (JAG1, JAG2, JAG3, JAG4, od kojih se po potrebi JAG3 i JAG4 spajaju u jednu). Postoji 5 (2 u Jagićevoj) učionica opremljenih računarima za potrebe praktične nastave. (Prilog 3)

U zavisnosti od kapaciteta i opremljenosti imamo sledeće tipove učionica:

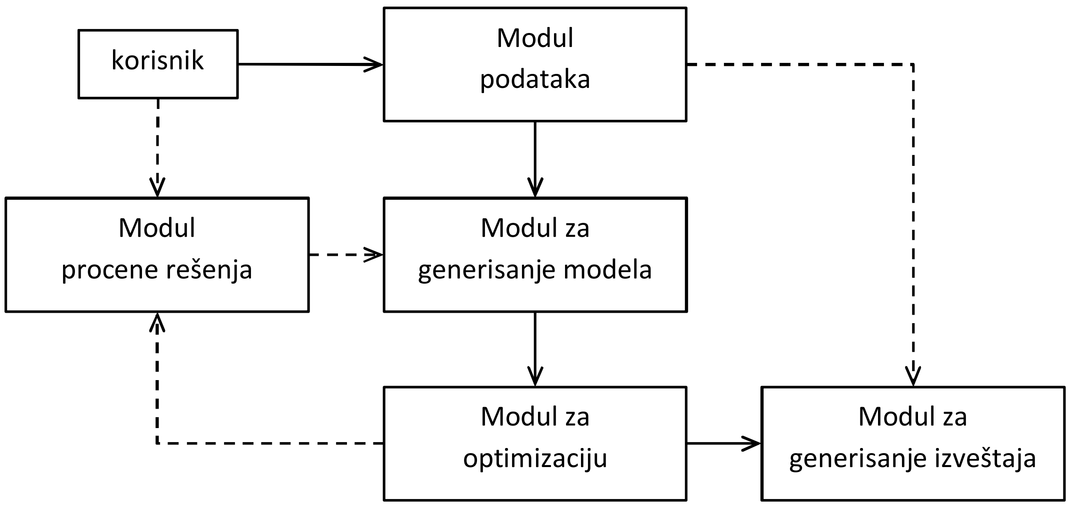
1. za predavanja većim grupama studenata: JAG1, JAG2, JAG3, JAG4, 704, 706, 718, 821, 830, 840, 843, 844, BIM, DLAB, RLAB
2. za predavanja manjim grupama: 710, 809, KABM, KABN, KAB825
3. učionice opremljene računarima: JAG1, JAG2, 718, RLAB, DLAB

Na osnovu do sada iznetih podataka možemo kreirati skup N_m.

6.2. Detalji implementacije

Ovako definisan model za raspored časova Matematičkog fakulteta spada u veći (komplikovaniji) problem 0-1 celobrojnog linearnog programiranja. Za rešavanje problema linearnog programiranja koristi se ***IBM ILOG CPLEX 12.2***. Za korisnički interfejs primenjen je programski jezik Java. Korisna osobina ove implementacije je što se korisniku omogućava promena ograničenja i generisanje novih rasporeda uzimajući u obzir dodele iz prethodno generisanih rasporeda.

Grafički prikaz aktivnosti i interakcije između disjunktnih modula softverskog rešenja dati su na slici 3.



Slika 3 Grafički prikaz interakcije modula

Modul podataka sačinjavaju procedure za pristup podacima i tabelama. Podaci su zadati u ***CSV*** (***comma separated values***) formatu.

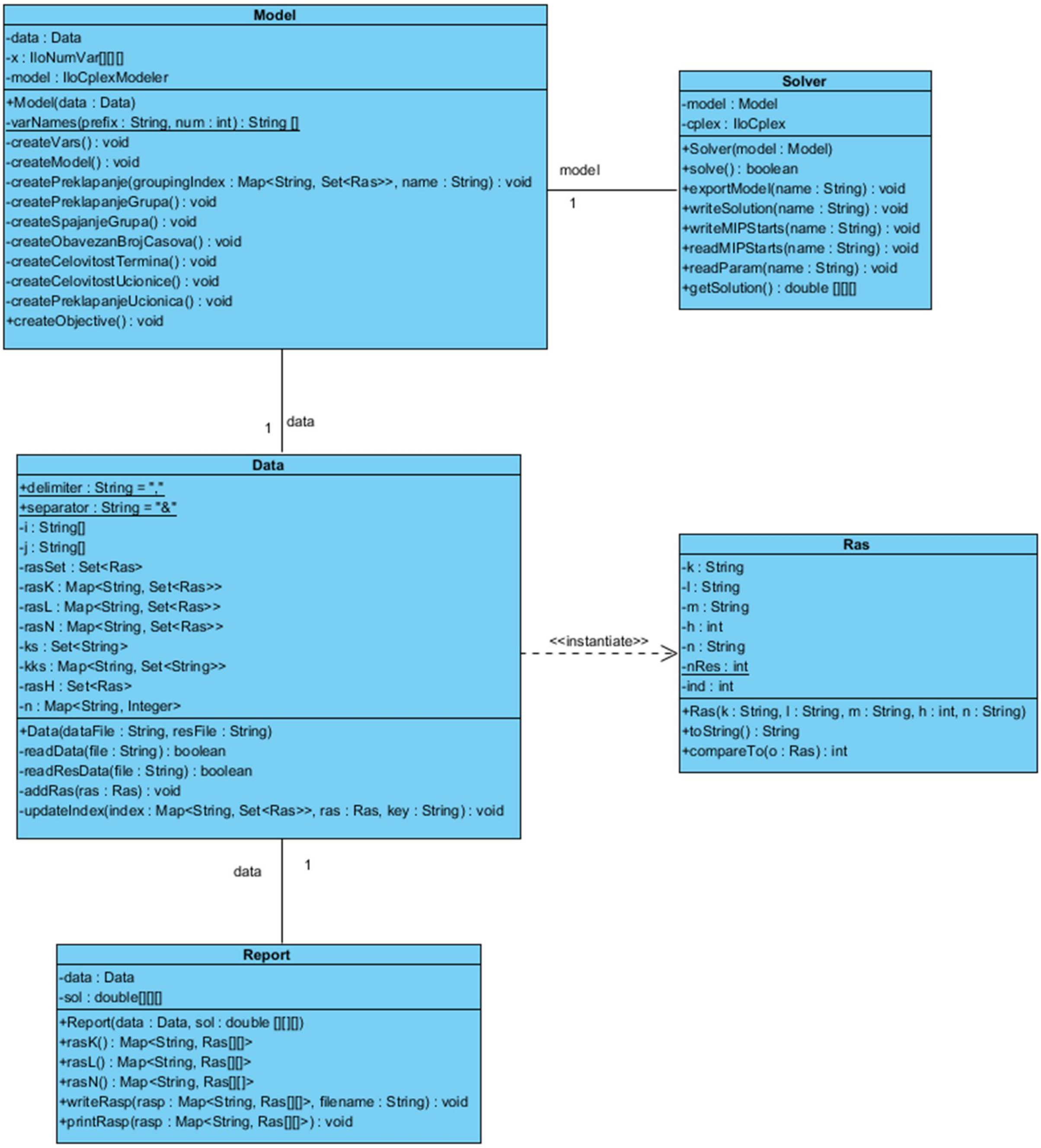
Modul za generisanje modela sadrži logiku koja prevodi podatke i korisničke zahteve u matematički model.

Modul za optimizaciju obuhvata rešavač IP modela, i omogućava postavljanje i izmenu parametara strategije optimizacije.

Modul za generisanje izveštaja interpretira rešenje modela u korisnički čitljive izveštaje:

- Raspored po grupama studenata
- Raspored po predavaču
- Raspored zauzeća učionica

Modul procene rešenja dozvoljava da, ako se uoče veće nedoslednosti, korisnik interaktivno menja dodeljene resurse, ili dodaje nova ograničenja i po potrebi nastavi pretragu.



Slika 4 Klasni dijagram aplikacije

Na prethodnoj slici prikazan je klasni dijagram programskog rešenja. Prikazane klase (date u prilogu) *Data*, *Model*, *Solver* i *Report* sadrže potrebne procedure za opisano funkcionisanje modula podataka, generisanje modela, optimizaciju i generisanje izveštaja respektivno. Instance klase *Ras* su elementi koji se moraju naći u rasporedu (nastavne jedinice).

6.3. Rešenje

U zadatom vremenskom roku od jednog sata program je dao moguće rešenje sa sledećim osobinama:

- ukupan broj promenljivih: 23530
- broj ograničenja: 32082
- teorijski minimum (bez uslova celobrojnosti): 3634.8333

Sa istim podacima za različita podešavanja dostizane su sledeće ocene kvaliteta rasporeda:

	test 1	**test 2**	**test 3**
Vreme [sec]	1800	3600	7200
Solution Polisher	< 5%	< 5%	> 3600sec
Funkcija cilja *(gap)*	3717 (2,21%)	3696 (1,63%)	3662 (0,74%)

6.3.1. Tabelarni prikaz rezultata

U Prilogu 4 dat je generisan raspored časova za I godinu smera Informatika.

Prilog 5 prikazuje dodeljen raspored predavanja za jednog predavača.

Izvorni kod programa kojim je rešavan problem raspoređivanja časova Matematičkog fakulteta, kao i svi generisani izveštaji (rasporedi časova za sve grupe i rasporedi predavanja za sve predavače) dati su na CD-ROM-u i sastavni su deo ovog rada (Prilog 6).

7. Zaključak

Kreiranje rasporeda je zahtevan i dugotrajan proces, ali se može značajno olakšati automatizacijom primenom računara. Velika složenost i varijacije problema nameću potrebu za optimizacijom postojećih rešenja. Najuspešniji algoritmi kombinuju moderne meta-heuristike sa već poznatim heurističkim znanjem.

Do dopustivog rešenja je moguće doći za par minuta, u zavisnosti od kompleksnosti ulaznih podataka, ali nalaženje optimalnog rešenja je često teško dostići na personalnim računarima sadašnje generacije. Ovakvi problemi zahtevaju paralelnu obradu kako bi do rešenja došli u razumnom vremenskom roku.

Problem automatizovanog raspoređivanja se kontinualno razmatra i unapređuje u poslednjih 40 godina i u skorašnje vreme privlači sve veći broj istraživača širom sveta. Svake godine se u Torontu održava konferencija *Practice and Theory of Automated Timetabling (PATAT)* gde se izlažu postignuti rezultati. Drugi značajan skup je *EURO Working group of Automated TimeTabling (WATT).* Iako se ogroman broj radova objavljuje svake godine, još uvek ne postoji standard za određivanje performansi i kvaliteta rezultata različitih metoda. U većini slučajeva, bez obzira na izabranu tehniku za rešavanje problema, u okviru razumnog vremena i raspoloživog memorijskog prostora računara, za sada su postignuta samo suboptimalna rešenja.

Prikazane metode i praktični rezultati pomenutih matematičkih modela za izradu rasporeda u ovom radu daju specifična rešenja za određeni univerzitet (problem). Svaki sledeći fakultet treba posmatrati ispočetka. S tim u vezi treba, npr. nakon sprovedene ankete ili na nekoj od tematskih konferencija koje se bave ovom problematikom, definisati standardan zajednički jezik (sistem) za opis zahteva problema i postaviti standarde koji daju kriterijume za ocenjivanje. Tek tada bi bilo mogućno dati uporedne ocene performansi različitih metoda i kvaliteta dobijenih rasporeda.

Predloženi matematički model za konstrukciju rasporeda časova na Matematičkom fakultetu Univerziteta u Beogradu rešavan je metodom celobrojnog linearnog programiranja. Teorijski broj promenljivih 5060874000 $(5 * 13 * 42 * 130 * 713 * 20)$ primenom predloženih redukcija smanjen je na 23530, što je učinilo problem rešivim u razumnom vremenskom roku.

Generisan raspored časova, kako za studente po godinama i smerovima, tako i za predavače, koji je dat u ovom radu samo je jedan od dopustivih rešenja modela. Kvalitet rešenja je zavisan od parametara i veličine ulaznih podataka zadatih rešavaču. Primenjen na test podacima sa manjim brojem resursa, program je dostizao optimalno rešenje.

Za dalji razvoj aplikacije treba razmotriti klijent-server arhitekturu kako bi rešavač mogao da iskoristi postojeću mrežnu infrastrukturu za paralelnu pretragu, i kako bi se većem broju klijenata ponudile usluge aplikacije. Takođe klijent-server arhitektura može obezbediti i korišćenje alternativnih rešavača (SCIP, CBC, lpsolve, GLPK, Xpress, Gurobi).

Literatura

1. Schaerf A.: A Survey of Automated Timetabling. Artificial Intelligence Review 13, 87-127 (1995)

2. Gotlieb C.: The construction of class-teacher timetables. In Popplewell C., ed. : IFIP congress, Munich, vol. 62, p.73–77 (1962)

3. Eiselt H., Laporte G.: Combinatorial optimization problems with soft and hard requirements. Journal of the Operational Research Society 38(9), 785–795 (1987)

4. de Werra D.: An introduction to timetabling. European Journal of Operational Research 19(2), 151–162 (1985)

5. Even S., Itai A., Shamir A.: On the Complexity of Timetable and Multi-Commodity Flow. SIAM Journal on Computing 5, 691-703 (1976)

6. Garey M.: Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1990)

7. Cvetković D., Čangalović M., Dugošija Đ., Simić S., Vuleta J.: Kombinatorna optimizacija - Matematička teorija i algoritmi. Društvo operacionih istraživača Jugoslavije - DOPIS, Beograd (1996)

8. Karmarkar N.: A New Polynomial Time Algorithm for Linear Programming. Combinatorica 4(4), 373–395 (1984)

9. Tripathy A.: School timetabling – A case in large binary integer linear programming. Management Science 30(12), 1473-1489 (1984)

10. Ferland J., Roy S.: Timetabling problem for university as assignment of activities to resources. Computers & Operations Research 12(2), 207–218 (1985)

11. Pearl J.: Heuristics: Intelligent search strategies for computer problem solving. Addison-Wesley (1984)

12. Dinkel J., Mote J., Venkataramanan M.: An Efficient Decision Support System for Academic Course Scheduling. Operations Research 37(6), 853–864 (1989)

13. Abramson D.: Constructing School Timetables using Simulated Annealing: Sequential and Parallel Algorithms. Management Science 37(1), 98-113 (1991)

14. Yan H., Yu S.-N.: A Multiple-Neighborhoods-Based Simulated Annealing Algorithm For Timetable Problem. (2010)

15. Hertz A.: Tabu search for large scale timetabling problems. European Journal of Operational Research 54, 39–47 (1991)

16. Solotorevsky G., Gudes E., Meisels A.: RAPS: a rule-based language for specifying resource allocation and time-tabling problems., 681 - 697 (1944)

17. Lassez J., L J.: Constraint logic programming. Proceedings of the 14th ACM POPL. Symposium (1987)

18. Burke E., Jackson K., Kingston J., Weare R.: Automated University Timetabling: The State of the Art. The Computer Journal 40, 565-571 (1997)

19. Azevedo F., Barahona P.: Timetabling in Constraint Logic Programming. The World Congress on Expert Systems (1994)

20. Corne D., Fang H.-L., Mellish C.: Solving the modular exam scheduling problem with genetic algorithms., University of Edinburgh (1993)

21. Erben W., Keppler J.: A Genetic Algorithm Solving a Weekly Course-Timetabling Problem., Fachhochschule Konstanz, Konstanz (1995)

22. Daskalaki S., Birbas T., Housos T.: An integer programming formulation for a case study in university timetabling. European Journal of Operational Research 153(1), 117–135 (2004)

23. Murray K., Müller T., Rudová H.: Modeling and Solution of a Complex University Course Timetabling Problem. Practice and Theory of Automated Timetabling VI, 189-209 (2007)

24. Bakir M., Aksop C.: A 0-1 Integer Programming Approach to a University Tinetabling Problem. Hacettepe Journal of Mathematics and Statistics 37(1), 41-55 (2008)

25. Dimopoulou M., Miliotis P.: Implementation of a university course and examination timetabling system. European Journal of Operational Research 130(1), 202-213 (2001)

26. Burke E., Mareček J., Parkes A., Rudová H.: A branch-and-cut procedure for the Udine Course Timetabling problem. Annals of Operations Research 194(1), 71-87 (2010)

27. Ostermann R., de Werra D.: Some experiments with a timetabling system. OR Spektrum 3, 199–204 (1983)

28. IBM ILOG CPLEX V12.2 User's Manual for CPLEX. (2009)

29. Shafranovich Y.: Common Format and MIME Type for Comma-Separated Values (CSV) Files (RFC 4180). Available at: http://tools.ietf.org/rfc/rfc4180.txt

Prilog 1: Predavači i njihov status

Ime predavača	ST	Ime predavača	ST	Ime predavača	ST
Aleksandar Lipkovski	R	Zoran Lučić	V	Milena Vujošević-Janičić	A
Aleksandar Torgašev	R	Zoran Marković	V	Miljan Knežević	A
Boško Jovanović	R	Zoran Ognjanović	V	Mladen Nikolić	A
Danko Jocić	R	Aleksandar Vučić	D	Sana Stojanović	A
Darko Milinković	R	Dragana Todorić	D	Slavko Moconja	A
Đorde Dugošija	R	Dragoljub Kečkić	D	Sonja Telebaković	A
Dušan Tošić	R	Filip Marić	D	Staša Vujičić	A
Gordana Pavlović-Lažetić	R	Jelena Katić	D	Stefan Miloševic	A
Julka Knežević Miljanović	R	Miroslav Marić	D	Tanja Stojadinović	A
Milutin Dostanić	R	Miroslava Antić	D	Tijana Šukilović	A
Miodrag Mateljević	R	Nebojša Ikodinović	D	Vesna Pavlović	A
Miodrag Živković	R	Nenad Mitić	D	Viktor Obuljen	A
Mirjana Đorić	R	Saša Malkov	D	Zlatko Lazović	A
Miroljub Jevtić	R	Srđan Vukmirović	D	Aleksandar Đenić	S
Miroslav Pavlović	R	Vladica Andrejić	D	Aleksandar Kartelj	S
Nadežda Pejović	R	Vladimir Božin	D	Ana Spasić	S
Nebojša Lažetić	R	Vladimir Filipović	D	Andrijana Savković	S
Neda Bokan	R	Vladimir Grujić	D	Anđelka Zečević	S
Pavle Mladenović	R	Zoran Petrović	D	Danijela Petrović	S
Siniša Vrećica	R	Zoran Stanić	D	Dušan Marčeta	S
Vladimir Janković	R	Zorica Stanimirović	D	Ivan Čukić	S
Zoran Kadelburg	R	Aleksandar Savić	A	Ivan Dimitrijević	S
Zoran Rakić	R	Angelina Ilić-Stepić	A	Ivana Tanasijević	S
Žarko Mijajlović	R	Biljana Stojanović	A	Jelena Dunjić	S
Aleksandar Jovanović	V	Biljana Vujošević	A	Jovana Đuretić	S
Andjelka Kovačevic	V	Bojan Novaković	A	Lenka Živadinović	S
Desanka Radunović	V	Branislav Prvulović	A	Maja Roslavcev	S
Duško Vitas	V	Đorde Krtinić	A	Marija Mikić	S
ENG - Irena Pavlović	V	Goran Đanković	A	Marijana Babić	S
ENG - Zoran Pavlović	V	Ivan Anić	A	Maša Vuković	S
Gojko Kalajdžić	V	Jelena Graovac	A	Milan Perić	S
Milan Božić	V	Jelena Hadži-Purić	A	Miloš Antić	S
Milan Dražić	V	Jelena Škorić	A	Miloš Đorić	S
Milan Tuba	V	Jovan Vukmirović	A	Mirjana Maljković	S
Miloš Arsenović	V	Jovana Kovačević	A	Mirko Spasić	S
Nastavnik PDI	V	Maja Rabrenović	A	Mirko Stojadinović	S
Nastavnik PSI	V	Marek Svetlik	A	Nevena Milojković	S
Predrag Janičić	V	Marija Ivanović	A	Nikola Mutavdžić	S
Predrag Tanović	V	Marija Milanović	A	Nina Radojičić	S
Slobodanka Janković	V	Marko Obradović	A	Sandra Hodžić	S
Stevo Šegan	V	Marko Radovanović	A	Stefan Mišković	S
Veljko Milutinović	V	Milan Banković	A	Vuksan Mijović	S
Vesna Jevremović	V	Milan Jovanović	A	Zorica Dražić	S

Prilog 2: Predmeti za raspoređivanje

Grupa	Predavač	Naziv predmeta	fond	kap
1I1A	Slavko Moconja	Diskretne strukture 1 (vezbe)	2	v
1I1A	Milos Antic	Linearna algebra i analiticka geometrija (vezbe)	2	v
1I1A	Danijela Petrovic	Programiranje 1 (vezbe)	3	v&r
1I1A	Andelka Zecevic	Uvod u organizaciju racunara (vezbe)	2	v
1I1A&1I1B&1I1C	Aleksandar Jovanovic	Diskretne strukture 1	3	v
1I1A&1I1B&1I1C	Miroslava Antic	Linearna algebra i analiticka geometrija	3	v
1I1A&1I1B&1I1C	Predrag Janicic	Programiranje 1	3	v
1I1A&1I1B&1I1C	Nenad Mitic	Uvod u organizaciju racunara	3	v
1I1A&1I1B&1I1C&1I2A&1I2B&1I2C	ENG - Zoran Pavlovic	Engleski 1	3	v
1I1A&1I1B&1I1C&1I2A&1I2B&1I2C	Zoran Markovic	Kultura komunikacije	2	v
1I1A&1I1B&1I1C&1I2A&1I2B&1I2C	Marko Radovanovic	Kultura komunikacije (vezbe)	1	v
1I1B	Slavko Moconja	Diskretne strukture 1 (vezbe)	2	v
1I1B	Milos Antic	Linearna algebra i analiticka geometrija (vezbe)	2	v
1I1B	Aleksandar Kartelj	Programiranje 1 (vezbe)	3	v&r
1I1B	Danijela Petrovic	Uvod u organizaciju racunara (vezbe)	2	v
1I1C	Slavko Moconja	Diskretne strukture 1 (vezbe)	2	v
1I1C	Milos Antic	Linearna algebra i analiticka geometrija (vezbe)	2	v
1I1C	Jelena Dunjic	Programiranje 1 (vezbe)	3	v&r
1I1C	Andelka Zecevic	Uvod u organizaciju racunara (vezbe)	2	v
…				
3LA	Maja Roslavcev	Metodika nastave matematike A (vezbe)	2	v
3LB	Maja Roslavcev	Metodika nastave matematike A (vezbe)	2	v
3LA&3LB	Sinisa Vrecica	Ocigledna topologija	2	v
3LA&3LB	Branislav Prvulovic	Ocigledna topologija (vezbe)	2	v
3LA&3LB	Dragana Todoric	Teorija brojeva 1	2	v
3LA&3LB	Goran Djankovic	Teorija brojeva 1 (vezbe)	2	v
3LA&3MA&3MB	Milan Jovanovic	Verovatnoca i statistika A (vezbe - stari)	2	v
3LA&3LB&3NA&3NB	Dragana Todoric	Algebra 2	2	v
3LA&3LB	Marko Radovanovic	Algebra 2 (vezbe)	2	v
3LB&3NA&3NB	Milan Jovanovic	Verovatnoca i statistika A (vezbe - stari)	2	v
3MA&3MB	Danko Jocic	Analiza 3A	3	v
3MA&3MB	Djorde Krtinic	Analiza 3A (vezbe)	2	v
3MA&3MB	Marija Mikic	Diferencijalne jednacine A (vezbe)	2	v
3MA&3MB	Miodrag Mateljevic	Kompleksna analiza A	2	v
3MA&3MB	Maja Rabrenovic	Kompleksna analiza A (vezbe)	2	v
…				
5ML	Zoran Kadelburg	Elementarne funkcije	3	v
5ML	Zlatko Lazovic	Elementarne funkcije (vezbe)	2	v&r
5ML	Miroslav Maric	Metodika nastave racunarstva	3	v
5ML	Sana Stojanovic	Metodika nastave racunarstva (vezbe)	2	v&r
5ML	Milan Bozic	Metodologija istrazivanja u nastavi matematike	3	m
5ML	Ivan Anic	Metodologija istrazivanja u nastavi matematike (vezbe)	2	m
5ML	Miodrag Mateljevic	Nacela nastave matematike	2	m
5ML	Ivan Anic	Nacela nastave matematike (vezbe)	2	m
5MM	Mirjana Djoric	Hiperbolicka geometrija	3	m
5MM	Mirjana Djoric	Hiperbolicka geometrija (vezbe)	2	m
5MM	Zoran Rakic	Lijeve grupe i algebre	3	m
5MM	Zoran Rakic	Lijeve grupe i algebre (vezbe)	2	m
5MM	Jelena Katic	Matematicki metodi mehanike	3	v
5MM	Darko Milinkovic	Odabrana poglavlja globalne analize	3	v
5MM	Jovana Djuretic	Odabrana poglavlja globalne analize (vezbe)	2	v
5MM	Miodrag Mateljevic	Odabrana poglavlja kompleksne analize	3	m
5MM	Miljan Knezevic	Odabrana poglavlja kompleksne analize (vezbe)	2	m
5MM&5MN	Danko Jocic	Spektralna teorija operatora	3	v
5MN	Bosko Jovanovic	Metode konacnih elemenata	3	m
5MV	Obuljen Viktor	Teorija informacije (vezbe)	2	m

Prilog 3: Spisak učionica

Učionica	Kapacitet	Opremljenost	Lokacija
704	v		ST
706	v		ST
710	m		ST
718	v	r	ST
809	m		ST
821	v		ST
830	v		ST
840	v		ST
843	v		ST
844	v		ST
BIM	v		ST
DLAB	v	r	ST
JAG1	v	r	JAG
JAG2	v	r	JAG
JAG3	v		JAG
JAG4	v		JAG
KAB 825	m		ST
KABG 838	m		ST
KABN 453	m		ST
RLAB	v	r	ST

Prilog 4: Primer generisanog rasporeda za grupu 1I1A

	Ponedeljak	Utorak	Sreda	Četvrtak	Petak
08	1I1A&1I1B&1I1C&1I2A&1I2B&1I2C Marko Radovanovic Kultura komunikacije (vezbe)	1I1A&1I1B&1I1C&1I2A&1I2B&1I2C Zoran Markovic Kultura komunikacije	1I1A&1I1B&1I1C Nenad Mitic Uvod u organizaciju racunara	1I1A&1I1B&1I1C Predrag Janicic Programiranje 1	1I1A&1I1B&1I1C Miroslava Antic Linearna algebra i analiticka geometrija
09	1I1A&1I1B&1I1C Aleksandar Jovanovic Diskretne strukture 14	1I1A&1I1B&1I1C&1I2A&1I2B&1I2C Zoran Markovic Kultura komunikacije	1I1A&1I1B&1I1C Nenad Mitic Uvod u organizaciju racunara	1I1A&1I1B&1I1C Predrag Janicic Programiranje 1	1I1A&1I1B&1I1C Miroslava Antic Linearna algebra i analiticka geometrija
10	1I1A&1I1B&1I1C Aleksandar Jovanovic Diskretne strukture 1	1I1A Milos Antic Linearna algebra i analiticka geometrija (vezbe)	1I1A&1I1B&1I1C Nenad Mitic Uvod u organizaciju racunara	1I1A&1I1B&1I1C Predrag Janicic Programiranje 1	1I1A&1I1B&1I1C Miroslava Antic Linearna algebra i analiticka geometrija
11	1I1A&1I1B&1I1C Aleksandar Jovanovic Diskretne strukture 1	1I1A Milos Antic Linearna algebra i analiticka geometrija (vezbe)	1I1A Slavko Moconja Diskretne strukture 1 (vezbe)	1I1A&1I1B&1I1C&1I2A&1I2B&1I2C ENG - Zoran Pavlovic Engleski 1	1I1A Danijela Petrovic Programiranje 1 (vezbe)
12	1I1A Andelka Zecevic Uvod u organizaciju racunara (vezbe)		1I1A Slavko Moconja Diskretne strukture 1 (vezbe)	1I1A&1I1B&1I1C&1I2A&1I2B&1I2C ENG - Zoran Pavlovic Engleski 1	1I1A Danijela Petrovic Programiranje 1 (vezbe)
13	1I1A Andelka Zecevic Uvod u organizaciju racunara (vezbe)			1I1A&1I1B&1I1C&1I2A&1I2B&1I2C ENG - Zoran Pavlovic Engleski 1	1I1A Danijela Petrovic Programiranje 1 (vezbe)
14					
15					
16					
17					
18					
19					
20					

Prilog 5: Primer generisanog rasporeda za predavača Žarka Mijajlovića

	Ponedeljak	Utorak	Sreda	Četvrtak	Petak
08					2MNVA&2MNVB Zarko Mijajlovic Algebra 1
09	1I2A&1I2B&1I2C Zarko Mijajlovic Diskretne strukture 1				2MNVA&2MNVB Zarko Mijajlovic Algebra 1
10	1I2A&1I2B&1I2C Zarko Mijajlovic Diskretne strukture 1			1O1A&1O1B Zarko Mijajlovic Linearna algebra A	2MNVA&2MNVB Zarko Mijajlovic Algebra 1
11	1I2A&1I2B&1I2C Zarko Mijajlovic Diskretne strukture 1			1O1A&1O1B Zarko Mijajlovic Linearna algebra A	
12				1O1A&1O1B Zarko Mijajlovic Linearna algebra A	
13					
14					
15					
16					
17					
18					
19					
20					

Prilog 6: Sadržaj CD-ROM-a

Direktorijum	Sadržaj
[bin]	Folder sa kompajliranim programom
[data-1800]	Datoteke za test 1 iz poglavlja 6.3
[data-3600]	Datoteke za test 2 iz poglavlja 6.3
[data-7200]	Datoteke za test 3 iz poglavlja 6.3
[doc]	Elektronska verzija dokumenta
[src]	Izvorni kod aplikacije

Sadržaj datoteka u folderima [data]

Datoteka	Sadržaj
mstr.csv	Podaci o časovima
mstr.res.dat	Podaci o resursima
testK.html	Generisani rasporedi po grupama
testL.html	Generisani rasporedi po predavačima
node.log	Izlaz CPLEX rešavača
test.lp*	Generisani model
testObj.lp*	Generisani model sa uključenom funkcijom cilja
test.mst*	Dostignuta moguća rešenja pri optimizaciji u XML formatu
test.prm*	Parametri rešavača za fazu traženja početnog rešenja
testObj.prm*	Parametri rešavača za fazu traženja optimalnog rešenja
test.sol*	Rešenje u XML formatu

* sintaksa datoteka opisana u priručniku „File formats supported by CPLEX”