University of East Sarajevo
Mathematical Society of the Republic of Srpska

Third Mathematical Conference of the Republic of Srpska

PROCEEDINGS

Trebinje, 07-08 June 2013

VOLUME II
Trebinje, 2014

ISBN 978-99976-600-1-5

Redakcija

Izdavač

Univerzitet u Istočnom Sarajevu

Fakultet za proizvodnju i menadžment Trebinje

Stepe Stepanovića bb, 89 101 Trebinje, BiH

Telefon: +387 (0)59 490 654

E-mail: fpmtrebinje@gmail.com

University of East Sarajevo
Mathematical Society of the Republic of Srpska

THIRD MATHEMATICAL CONFERENCE OF THE REPUBLIC OF SRPSKA

PROCEEDINGS

Trebinje, 07-08 June 2013

VOLUME II
Trebinje, 2014

TREĆA MATEMATIČKA KONFERENCIJA REPUBLIKE SRPSKE

ZBORNIK RADOVA
TOM II

Izdavač:
Fakultet za proizvodnju i menadžment Trebinje, Univerzitet u Istočnom Sarajevu

Za izdavača:
Doc. dr Dušan Jokanović

Glavni urednik:
Prof. dr Milenko Pikula

Tehnički urednik i kompjuterski slog:
Marina Zirojević

Lektura i korektura:
Maja Kovačević

Štampa:
"Grafokomerc" a. d. Trebinje

Tiraž:
300

Trebinje, 2014.

Sadržaj

1. Radoje Šćepanović – Primjena teoreme o nepokretnoj tački u nastavi matematike **1**

2. Dragan Matić, Vladimir Filipović, Jozef Kratica – Poboljšanje nastavnog plana optimizovanjem broja pokrivenih tematskih oblasti po semestrima **11**

3. Nebojša Elez, Ognjen Papaz, Ivana Čorlija – Spojene familije skupova i povezani topološki prostori **29**

4. Tatjana Bajić – Tabele kontigencije i problem maksimalne uslovne informacije **33**

5. Gordana Jelić – Grupe homologija i mogućnost njihovih izračunavanja **45**

6. Saša Mujović, Slobodan Đukanović – O upotrebi metode najmanjih kvadrata za modelovanje ključnih parametara kvaliteta električne energije **51**

7. Veljko Vranić, Djordje Baralić – Cinderella - način da vidimo apstraktnu matematiku **63**

8. Nebojša Elez, Marko Ćitić – Uopštenje nejednakosti paralelograma **73**

9. Šefket Arslanagić – Jedna interesantna metoda dokazivanja nejednakosti **75**

10. Radoslav Milošević – Logička formulacija elementarne teorije brojeva (FETB) **83**

11. Milan Živanović – Relacije i matematičke operacije na jednoj klasi Pitagorinih trouglova **91**

12. Ivan Budimir – Vjerojatnost kockareve propasti u igri ruleta **97**

13. Vladan Mastilović, Zorana Banjanin – Servisno orijentisana arhitektura informacionog sistema integrisanog univerziteta **101**

TREĆA MATEMATIČKA KONFERENCIJA REPUBLIKE SRPSKE
Trebinje, 7. i 8. juni 2013. godine

Primjena teoreme o nepokretnoj tački u nastavi matematike

Radoje Šćepanović
Univerzitet Crne Gore
radoje@rc.pmf.ac.me

Stručni rad

Apstrakt

U ovom radu navodimo nekoliko primjera iz školske matematike u kojima se koriste teoreme o nepokretnim tačkama.

1 Nepokretne tačke

Neka je f -preslikavanje proizvoljnog skupa A u sebe. Tačku $x \in A$ nazivamo nepokretnom (fiksnom) tačkom preslikavanja f, ako je

$f(x)=x$. (1)

Kada je $A \subset R$, tj. kada je f prosto brojevna funkcija, naći nepokretnu tačku preslikavanja f znači riješiti jednačinu (1). Obratno, ako je data proizvoljna jednačina $g(x)=0$, tada su njena rješenja nepokretne tačke preslikavanja f, gdje je

$f(x)=g(x)+x$.

Primjer 1. Preslikavanje $x \to ax^2+bx+c$, gdje je $a \neq 0$, ima jednu, dvije ili nijednu nepokretnu tačku u zavisnosti od znaka izraza $\Delta=(b-1)^2-4ac$.

Na osnovu broja nepokretnih tačaka može se izvršiti klasifikacija izometrijskih i afinih preslikavanja u ravni i prostoru.

Primjer 2. Izometrija je preslikavanje prostora u samog sebe koje čuva rastojanja među tačkama. Važna karakteristika izometrijskih preslikavanja u prostoru je skup njenih nepokretnih tačaka. Postoji samo pet takvih slučajeva:

1. Izometrija koja nema nepokretnih tačaka je translacija za nenulti vektor,
2. Izometrija koja ima samo jednu nepokretnu tačku je centralna simetrija,
3. Izometrija u kojoj su nepokretne samo tačke jedne prave je rotacija oko te prave,
4. Izometrija u kojoj su nepokretne samo tačake jedne ravni je simetrija u odnosu na tu ravan,
5. Izometija u kojoj su nepokretne sve tačke prostora je identičko preslikavanje.

Drugih izometrijskih preslikavanja u prostoru nema (ovdje su uključene i kompozicije ovih osnovnih izometrijskih preslikavanja). Na sličan način se može izvršiti i klasifikacija izometrijskih preslikavanja u ravni.

Primjer 3. Preslikavanje ravni u sebe nazivamo afinim ako svaku pravu preslikava u pravu i ako patralelne prave preslikava u paralelne prave.

1. Afino preslikavanje koje nema nepokretnih tačaka je translacija,
2. Afino preslikavanje koje ima samo jednu nepokretnu tačku je centralna simetrija,
3. Ako afino preslikavanje ima dvije nepokretne tačke A i B, tada je svaka tačka prave (AB) nepokretna.

Nepokretne tačke nema svako preslikavanje, na primjer, preslikavanje $f(x)=x+2$ skupa R (realnih brojeva) u skup R nema nepokretnih tačaka.

Zadatak 1. Naći preslikavanje ravni u samu sebe koje ima tačno n nepokretnih tačaka ($n \in \mathrm{N}$).

2 Teorema o kontraktivnom preslikavanju. Primjeri iz geometrije

Od posebnog značaja su tzv. kontraktivna (sažimajuća) preslikavanja. Kao što smo uočili, rješenje jednačine oblika $g(x)=0$ se svodi, prostim dodavanjem x njenoj lijevoj strani, na nalaženje nepokretne tačke nekog preslikavanja f, gdje je $f(x)=g(x)+x$. Zato je važno znati uslove pod kojima postoji nepokretna tačka i znati kako naći nepokretnu tačku.

Neka je dato preslikavanje $f\colon A \to A$, gdje je A skup u kojem je definisano rastojanje $d(x,y)$ među tačkama x i y. Na primjer, skup A može biti podskup prave, ravni ili prostora.

Definicija 2.1. Za preslikavanje $f{:}A \to A$ kažemo da je kontraktivno (ili da je kontrakcija, sažimajuće) ako

$$(\exists q \in [0,1))\,(\forall x, y \in A) : d\,(f\,(x)\,, f\,(y)) \leq qd\,(x,y)\,.$$

Drugim riječima, rastojanje među slikama je manje od rastojanja među orginalima kojima odgovaraju te slike, pri čemu se ono umanjuje q "puta". Broj q nazivamo koeficijentom kontrakcije (sažimanja).

Primjer 4.

a) Homotetija ravni sa koeficijentom homotetije k, gdje je $0<k<1$, je kontraktivno preslikavanje.

b) Rotacija ravni nije kontraktivno preslikavanje.

Svako kontraktivno preslikavanje je neprekidno.

Teorema 2.1 (Princip kontraktivnog preslikavanja na pravoj)**.** Svako kontraktivno preslikavanje $f\colon\,]a,b[\to]a,b[$ ima jedinstvenu nepokretnu tačku.

Dokaz 2.1. U osnovi dokaza leži metod iteracije. Izaberimo, na proizvoljan način, $x_0\in]a,b[$ i formirajmo niz

x_0, $x_1=f(x_0)$, $x_2=f(x_1)$, ..., $x_n=f(x_{n-1})$,...

Dokažimo da niz (x_n) konvergira ka traženoj nepokretnoj tački. Za ovo je dovoljno dokazati da je niz (x_n) Košijev (fundamentalan), jer u prostoru realnih brojeva R svaki Košijev niz konvergira. Za ovo je dovoljno dokazati da za svako $\epsilon>0$ postoji prirodan broj n_0 takav da za svako $n,m>n_0$ važi nejednakost

$d(x_n,x_m)<\epsilon$. (2)

Uočimo da je $d(x_k,x_{k+1})<q^{k-1}d(x_1,x_2)$. Dalje je

$d(x_n,x_{n+p})\leq d(x_n,x_{n+1})+ d(x_{n+1},x_{n+2})+....+d(x_{n+p-1},x_{n+p})\leq(q^{n-1}+ q^{n-2}+...+ q^{n+p-2})^1 d(x_1,x_2)\leq (q^{n-1}+ q^{n-2}+...+ q^{n+p-2}+....)d(x_1,x_2) \leq\frac{q^{n-1}}{1-q}d\,(x_1-x_2)$. (3)

Sada, za proizvoljno dato $\epsilon>0$ za broj n_0 možemo uzeti, na primjer, $n_0 = \log_q \frac{\varepsilon(1-q)}{d(x_1,x_2)}$. Ovim smo dokazali da je niz (x_n) Košijev, tj. da je konvergentan. Neka je $\lim\limits_{n\to+\infty} x_n = x^*$. Iz graničnog prelaza u $x_n=f(x_{n-1})$, kada $n\to+\infty$, dobijamo da je $f\,(x^*) = x^*$, tj. x^* je nepokretna tačka preslikavanja f. Jedinstvenost nepokretne tačke je očigledna. Stvarno, ako bi postajale dvije različite nepokretne tačke x_1 i x_2, tada bi saglasno definiciji kontraktivnog preslikavanja imali da je $d(x_1,x_2)<q(x_1,x_2)$, tj. $q>1$, što nije moguće.

Lako se uočava da teorema 1 važi ako interval $]a,b[$ zamijenimo odsječkom, polupravom ili čitavom pravom R. Štaviše, ona važi i u prostoru. Sva ova tvrđenja su specijalni slučajevi sljedeće teoreme.

Teorema 2.2 (teorema o kontraktivnom preslikavanju)**.** Svako kontraktivno preslikavanje kompletnog metričkog prostora u sebe ima jednistvenu nepokretnu tačku.

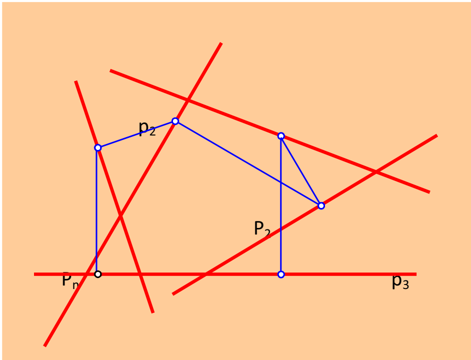
Slijedi nekoliko primjera teoreme o kontraktivnom preslikavanju.Uvedimo oznaku za ortogonalnu projekciju tačke P na pravu ili površ σ. Oznaka je $\mathrm{ort}_\sigma P$.

Primjer 5. U ravni su date prave p_1, p_2, ..., p_n. Neka je $P_1\in p_1$, $P_2 = ort_{p_2}P_1$, ..., $P_n = ort_{p_n}P_{n-1}$, $P_{n+1} = ort_{p_1}P_n$ (slika 1). Dokazati da postoji tačka $P_1\in p_1$ takva da je $P_{n+1}\equiv P_1$.

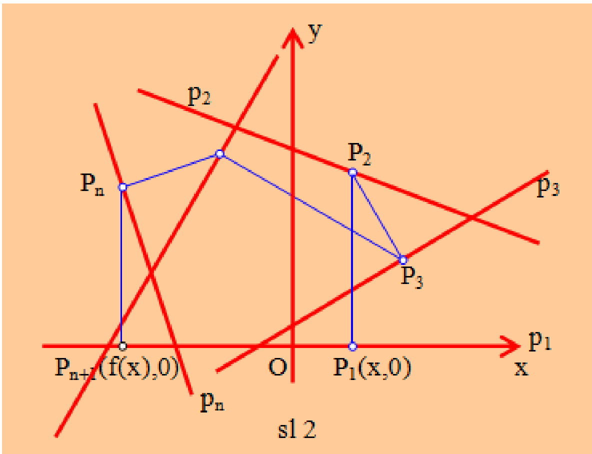
U slučaju da su sve prave p_i, $i=1,2,...,n$ među sobom paralelne, tada za svaku tačku $P_1\in p_1$ važi da je $P_{n+1}\equiv P_1$. Pretpostavimo da među pravama p_1, p_2, ..., p_n postoje bar dvije neparalelne. Neka su to prave p_i i p_{i+1}. Prava p_i $(i=1,2,...,n)$ u Dekartovom pravouglom sistemu Oxy ima jednačinu $a_ix+b_iy+c_i=0$. Ovaj koordinatni sistem izaberimo tako da se prava p_1 poklapa sa osom Ox (sl 2).

Birajući tačku $P_1\in p_1$ mi zapravo biramo apscisu x tačke P_1 na osi Ox. Kada se nađe tačka P_{n+1}, njenu apscisu označimo sa $f(x)$. Na ovaj način je definisano jednoznačno preslikavanje $f{:}R\to R$ (apscisi $x\in$R tačke P_1 pridružena je apscisa $f(x)\in$R tačke P_{n+1}). Za svako $j=1,2,...,n$ imamo:

$$(\exists q_j \in [0,1]) \left(\forall P_j, P_j^{'} \in p_j\right) : \overline{P_{j+1}P_{j+1}^{'}} = q_j\overline{P_jP_j^{'}},$$



Slika 1:



Slika 2:

gdje je sa$\overline{P_j P_j^{'}}$ označena dužina duži $P_j P_j^{'}$. Kako prave p_i i p_{i+1} nijesu paralelne, to je $0 \leq q_j < 1$. Iz prethodnog slijedi,

$$(\exists q \in [0,1)) \left(\forall P_1, P_1^{'} \in p_1\right) : \overline{P_{n+1} P_{n+1}^{'}} \leq q \overline{P_1 P_1^{'}},$$

gdje je $q = q_1 \cdot q_2 \cdots q_n$. Zadnju nejednakost možemo zapisati u obliku

$$(\exists q \in [0,1)) \left(\forall x, x' \in R\right) : \left| f\left(x'\right) - f\left(x\right) \right| \leq q \left| x' - x \right|.$$

(Ovdje je x ($x\prime$) apscisa tačke P_1 ($P_1^{'}$), a $f(x)$ ($f(x\prime)$) apscisa tačke P_{n+1} ($P_{n+1}^{'}$). Slijedi, preslikavanje $f: R \to R$ je kontraktivno, pa saglasno teoremi 1, imamo

$$(\exists! x^* \in R) : f\left(x^*\right) = x^*,$$

tj.

$$(\exists! P_1 \in p_1) : P_{n+1} \equiv P_1$$

(x^* je apscisa tačke P_1).

Napomena 1. Ako postoji takvo i da je $p_i \perp p_{i+1}$, ili da je $p_n \perp p_1$ ($\perp$ - oznaka za normalnost pravih), tada se tačka P_1 iz prethodnog primjera može naći konstruktivno.

Kako naći tačku P_1 (za koju je $P_{n+1} \equiv P_1$)? Kako se tačka P_1 dobija, u opštem slučaju, u beskonačno mnogo koraka, to je konstruktivan prilaz (lenjirom i šestarom) nemoguć. Koristeći matematički aparat linearne algebre, neki od programskih jezika i računar, apscisa tačke P_1 se može izračunati.

Napomena 2. Bez teškoća se primjer 1 može uopštiti na slučaj pravih u prostoru: Date su prave p_1, p_2, ..., p_n u prostoru tako da među njima postoje bar dvije koje nijesu paralelne. Neka je $P_1 \in p_1$, $P_2 = ort_{p_2} P_1$, ..., $P_n = ort_{p_n} P_{n-1}$, $P_{n+1} = ort_{p_1} P_n$. Tada postoji jedinstvena tačka $P_1 \in p_1$ takva da je $P_{n+1} \equiv P_1$.

Posljedica. Ako je n=2 i ako se prave p_1 i p_2 mimoilaze, tada tačke P_1 i P_2, za koje je $P_3 \equiv P_1$, određuju zajedničku normalu (mimoilaznih) pravih p_1 i p_2. Najkraće rastojanje između pravih p_1 i p_2 realizuje se po duži P_1P_2.

Primjer 6. U prostoru su date ravni α_1, α_2, ..., α_n, od kojih bar dvije nijesu paralelne. Neka je P_1 tačka ravni α_1, $P_2 = ort_{\alpha_2} P_1$, ..., $P_n = ort_{\alpha_n} P_{n-1}$, $P_{n+1} = ort_{\alpha_1} P_n$. Dokazati da postoji jedinstvena tačka $P_1 \in \alpha_1$, takva da je $P_{n+1} \equiv P_1$.

Ravan α_i (i=1,2,...,n) u prostoru R^3 sa pravouglim Dekartovim sistemom $Oxyz$ ima jednačinu $A_i x + B_i y + C_i y + D_i = 0$. Koordinatni sistem $Oxyz$ biramo tako da se koordinatna ravan Oxy poklapa sa ravni α_1. Tačka $P_1 \in \alpha_1$ ima koordinate u=(x,y,0). Koordinate tačke $P_{n+1} \in \alpha_1$ označimo sa $f(u) = (\widetilde{x}, \widetilde{y}, 0)$. Na ovaj način je definisano jednoznačno, kontraktivno preslikavanje $f: R^2 \to R^2$. Saglasno uopštenju teoreme 1 imamo da $(\exists! P_1 \in \alpha_1) : P_{n+1} \equiv P_1$.

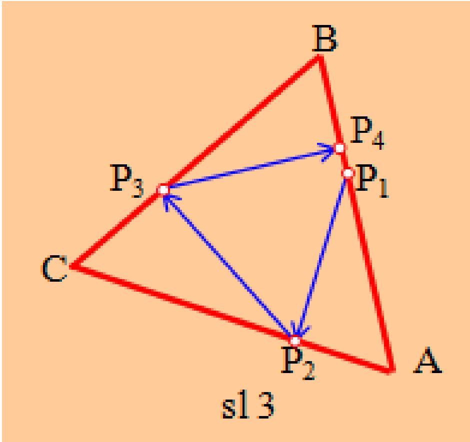
Primjer 7. Dat je oštrougli trougao ABC. Na stranici AB zadata je tačka P_1. Neka je $P_2 = ort_{BC} P_1$, $P_3 = ort_{AC} P_2$ i $P_4 = ort_{AB} P_3$. Dokazati da $(\exists! P_1 \in AB) : P_4 \equiv P_1$ (sl 3).

U ravni u kojoj se nalazi trougao ABC postavimo koordinatni sistem Oxy tako da se stranica AB pripada osi Ox. Stranici AB na osi Ox odgovara neki odsječak, označimo ga sa [a,b]. Birajući tačku $P_1 \in AB$ mi biramo apscisu x te tačke koja pripada osi Ox. Lako se dokazuje da će apscisa f(x) tačke P$_4$ pripadati odsječku [a,b]. Na ovaj način preslikavanje f: [a,b]$\to$[a,b] je kontraktivno, pa, saglasno teoremi 1, imamo $(\exists x_1 \in [a, b]) : f(x_1) = x_1$, odnosno $(\exists! P_1 \in AB) : P_4 \equiv P_1$.

Napomena 3. U vezi sa prethodnim primjerom mogu se postaviti razna pitanja: na primjer, da li tvrđenje važi ako je trougao: a) tupougli, b) pravougli? U kojim slučajevima je moguće konstruisati tačku P_1 za koju je $P_4 \equiv P_1$?

Primjer 8. Data je trostrana piramida $ABCD$ čiji su svi uglovi diedara oštri. Neka $P_1 \in ABC$, $P_2 = ort_{BCD} P_1$, $P_3 = ort_{ACD} P_2$, $P_4 = ort_{ABD} P_3$ i $P_5 = ort_{ABC} P_4$ Dokazati da $(\exists! P_1 \in ABC) : P_5 \equiv P_1$ (sl 4).

Zadatak 2. Neka su A i B dvije mape, pravougaonih oblika, jedne te iste



Slika 3:

oblasti urađene u različtim razmjerama. Neka je mapa A položena na mapu B tako da je, na jeziku skupova skupovno kazano, $A \subset B$. Dokazati da se vrhom olovke može pokazati samo jedno isto mjesto istovremeno na objema tako postavljenim mapama.

3 Brauerova teorema. Primjeri iz analize

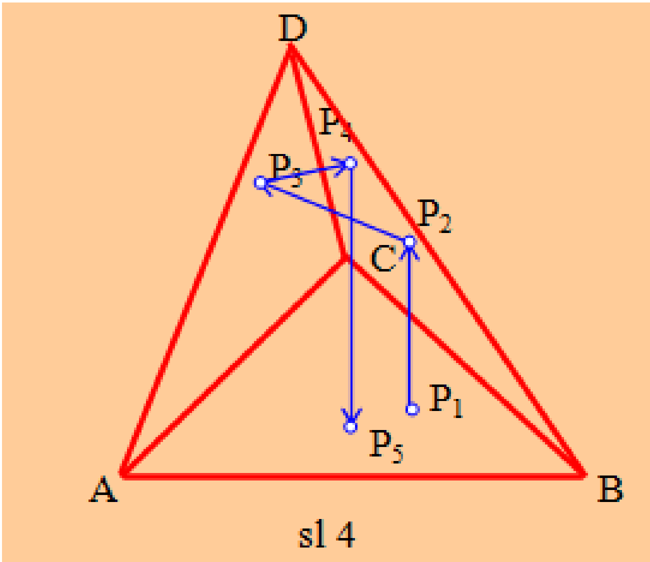
Učenici Brauerovu teoremu koriste, istina bez navođenja njenog imena, u završnim razredima srednje škole. Na primjer, u zadatku: Ako je $f \in C[a,b]$ i $f:[a,b] \to [a,b]$, tada postoji tačka $c \in [a,b]$ takva da je $f(c)=c$. Ovaj zadatak se može formulisati i u terminima nepokretne tačke:

Teorema 3. Ako je $f \in C[a,b]$ i $f:[a,b] \to [a,b]$, tada postoji nepokretna tačka preslikavanja f.

Napomenimo da su ovdje od suštinskog značaja pretpostavke da je f neprekidna funkcija na odsječku $[a,b]$ i da f preslikava odsječak $[a,b]$ u $[a,b]$. Očigledno, nepokretna tačka c ne mora biti jedinstvena.

Primjer 9. Neka dvije osobe šetaju: jedna polazi iz punkta A i ide stalnom brzinom u punkt B , a druga čitavo vrijeme šeta po putu AB. Očigledno ove osobe se moraju sresti, tj. u jednom trenutku se naći na istom mjestu. Matematički modelirajmo susret šetača. Postavimo punktove A i B na koordinatnu osu, označimo je sa Os (sl 5).

Neka je $A(a)$ i $B(b)$. Označimo sa x, $a \leq x \leq b$, koordinate prvog šetača, a su y, $a \leq y \leq b$ koordinate drugog šetača. Na ovaj način, svakom $x \in [a,b]$ pridružujemo po jedno $y=f(x) \in [a,b]$. Ovo znači da $f:[a,b] \to [a,b]$. Rješenje jednačine $f(x)=x$ određuje mjesto susreta. Na primjer, ako je $a=0$ i $b=1000$m, $f(x) = \frac{1}{2}x+200$, tada će do susreta doći u tački $x=250$m. Ukoliko je kretanje drugog šetača "složenije",



Slika 4:

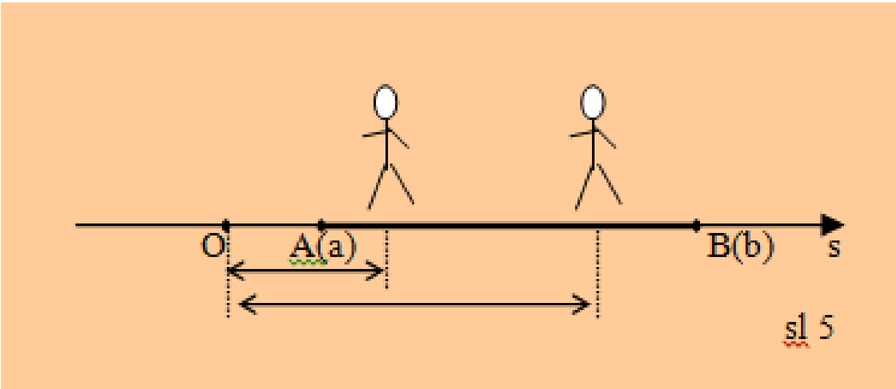
i tada će očigledno doći do njihovog susreta. Međutim, "očigledno" u matematici nije dokaz. Uočimo da će funkcija f u svim slučajevima biti neprekidna na odsječku $[a,b]$ i da će f preslikavati odsječak $[a,b]$ u $[a,b]$. Saglasno teoremi 2 imamo da postoji takvo x za koje je $f(x)=x$, tj. do susreta će doći u mjestu čija je koordinata x.

Primjer 10. U podnožju planine nalaze se monaške ćelije. U nedjelju izjutra, tačno u 7 časova, monah se počeo peti uz planinu i na njen vrh je stigao u 7 časova uveče. Prenoćio je na vrhu planine, a zatim je u 7 časova izjutra počeo da se istim putem vraća u ćeliju u koju je stigao u 7 časova uveče. Dokazati da na tom putu postoji mjesto u kojem je monah bio u isto vrijeme (na časovniku) kada se peo i kada se spuštao sa planine.

Rješavanje ovog zadatka koristi Brauerovu teoremu. Označimo sa [7,19] vremenski interval potreban da se monah popne, odnosno spusti sa planine. Dalje, označimo sa x vrijeme (na časovniku) kada je monah prošao određeno mjesto na putu ka vrhu planine, a sa $f(x)$ vrijeme (na časovniku) kada se monah našao na tom istom mjestu pri spuštanju sa planine. Očigledno, f je neprekidno preslikavanje na odsječku [7,19] i f:[7,19]→[7,19]. Saglasno teoremi 3 postoji takvo x za koje je $f(x)=x$. Ovo znači da postoji mjesto na putu monaha kada je njegov časovnik pokazivao isto vrijeme pri penjanju i pri spuštanju sa planine.

Primjer 11. Dokazati da jednačina: a) $x = \cos x^3$, b) $\cos(\sin x)-x=0$ ima rješenje.

a) Neka je $f(x) = \cos x^3$. Preslikavanje f je neprekidno na R i pri tome $f : \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \to [-1,1]$. Kako je $[-1,1] \subset \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, to je $f : \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \to \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$. Slijedi, postoji tačka $c \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ takva da je $f(c)=c$, odnosno $c = \cos c^3$, tj. c je



Slika 5:

rješenje date jednačine.

Zadatak 2. Navesti primjer neprekidne funkcije f:(0,1)→(0,1) koja nema nepokretnih tačaka na (0,1), tj. ne postoji tačka $c\in(0,1)$ za koju je $f(x)=x$. (Rješenje: Na primjer, $f(x)=x^2$).

Svi koji piju kafu (ili čaj) mogli su primijetiti da pri miješanju kafe u čaši u sredini čaše ostaje jedna nepokretna tačka. Istina, ovo zapažanje pretpostavlja "pravilno" miješanje kafe u trenutaku posmatranja. No i bez toga, u svakom slučaju bar jedna tačka ostaje nepokretna koja ne mora biti u sredini čaše. Ovo je posljedica sljedećeg tvrđenja.

Teorema 4. Svako neprekidno preslikavanje kruga K (kojemu pripada granica) u K ima bar jednu nepokretnu tačku.

Umjesto kruga K može se uzeti trougao, pravougaonik i sl. Teoreme 3 i 4 su specijalni slučajevi tzv. Brauerove teoreme:

Teorema 5. Neka je M konveksan, ograničen i zatvoren skup u konačnodimenzionalnom prostoru. Svako neprekidno preslikavanje iz M u M ima bar jednu nepokretnu tačku.

Strogi dokaz ove teoreme je dao holandski matematičar Brauer 1912. godine. Od tada do danas ova teorema je uopštavana i ima značajnu ulogu u nizu oblasti savremene matematike. Na primjer, koristi se u avioindustriji pri projektovanju raspona krila aviona, za analizu optimalne strategije i stanja ravnoteže u igrama kao i matematičkom modeliranju procesa u ekonomiji itd.

Primjer 12. Dokazati da sistem jednačina $\begin{cases} x = \sin(x+y) \\ y = \frac{1}{2}(x^2+xy^3) \end{cases}$ ima rješenje.

Neka je $D=\{(x,y):|x|\leq 1,\ |y|\leq 1\}$ i $f(x,y) = \begin{pmatrix} \sin(x+y) \\ \frac{1}{2}(x^2+xy^3) \end{pmatrix}$, $(x,y)\in D$. Kako je $|\sin(x+y)|\leq 1$ i $\left|\frac{1}{2}(x^2+xy^3)\right| \leq 1$ za svako $(x,y)\in D$, to je $f:D\to D$. Preslikavanje f je i neprekidno, pa saglasno teoremi 3, postoji u D nepokretna tačka preslikavanja f. Slijedi, dati sistem jednačina ima rješenje.

Evo još nekoliko zanimljivih primjena Brauerove teoreme.

a) Uzmimo prazan paket i list papira koji u potpunosti pokriva dno paketa. Neka svakoj tački lista papira odgovara ta tačka na dnu paketa koja se nalazi ispod nje. Izvadimo pomenuti list papira, od njega (gužvajući ga) napravimo "loptu", a zatim je vratimo u kutiju. Topolozi dokazuju, da nezavisno od toga, kako je od lista napravljena "lopta" i u kojem mjesto kutije je bačena, postoji najmanje jedna tačka na "lopti" ispod koje se nalazi njoj odgovarajuća tačka.

b) U svakom trenutku na Zemlji postoji takvo mjesto u kojem je brzina vjetra jednaka nuli.

c) Na Zemlji uvijek postoje dvije tačke – antipodi (leže na krajevima jednog prečnika Zemlje) u kojima su temperature i vazdušni pritisci jednaki.

Literatura

[1] Gegelija G. Princip kontraktivnih preslikavanja, Kvant, 9 (1980)

[2] Vertgejm B. Metod nepokretne tačke, Kvant, 8 (1980)

[3] Danilov V. T. Lekcije o nepokretnim tačkama, Ruska ekonomska škola, Moskva, 2006

[4] M. Gardner, Aha! Gotcha, W. H. Freeman and Company, San Francisko, 1982

TREĆA MATEMATIČKA KONFERENCIJA REPUBLIKE SRPSKE
Trebinje, 7. i 8. juni 2013. godine

Poboljšanje nastavnog plana optimizovanjem broja pokrivenih tematskih oblasti po semestrima

Dragan Matić
Univerzitet u Banja Luci, Prirodno matematički fakultet
matic.dragan@gmail.com

Vladimi Filipović
Univerzitet u Beogradu, Matematički fakultet
vladaf@matf.bg.ac.rs

Jozef Kratica
Matematički institut Beograd
jkratica@mi.sanu.ac.rs

Stručni rad

Apstrakt

U ovom radu se ispituju mogućnosti unapređenja nastavnih planova matematičkih i računarskih kurseva, primjenom alata kombinatorne optimizacije. Analizirani pristup je zasnovan na metodičkom stavu da osnove (i ne samo osnove) svake tematske cjeline treba periodično obrađivati u što više različitih konteksta, kao i na rastućem nivou složenosti. Ako je cilj organizovati lekcije u dva vremenska perioda, tako da što veći broj tematskih cjelina bude obrađivan u oba perioda, pokazuje se da odgovarajući matematički model posmatranog problema odgovara postavci poznatog kombinatornog problema dijeljenja skupa (Set Splitting Problem). U radu je na primjeru kursa iz Uvoda u računarstvo predložena organizacija datog kursa prema opisanom pristupu. Navedene su i opisane neke metode rješavanja odgovarajućeg matematičkog problema.

1 Uvod

Informatizacijom čitavog društva i povećanjem količine dostupnih informacija, obrazovanje se posljednjih godina susreće sa novim izazovom: da li velika količina obrazovnog materijala koja je dostupna studentima zaista olakšava učenje i podiže nivo studentovog znanja? Usljed nedovoljno dobro organizovanih nastavnih planova, velike količine dostupnih informacija, neadekvatnih udžbenika i metoda podučavanja, studentima je u doba intenzivne primjene naprednih metoda učenja često otežan obrazovni proces, što je suprotno od željenog cilja. Naglašavanjem da se, umjesto na istraživanju pitanja i podsticanju kritične misli, studentima nudi učenje odgovora, ulazi se u problem da studenti nisu podstaknuti da rade i uče zajedno, dijele ideje i informacije, te proširuju intelektualne sposobnosti.

Čest je slučaj da ambiciozni planovi i programi na matematičkim fakultetima uzrokuju nedostatak nastavnog kadra i lošu iskorištenost resursa, što sa jedne strane otežava nastavni proces, a sa druge generiše nepotrebne troškove. U nesređenim planovima i programima, neke teme se obrađuju do manje potrebnih detalja, dok su neke druge, iste ili čak i veće važnosti, preskočene, ili dostupne manjim grupama studenata. Dodatno, svaka tema u nauci, matematici ili tehnologiji koja se uči ili pominje samo u jednoj lekciji, najvjerovatnije neće biti dovoljno shvaćena i naučena od strane studenta. Da bi se koncept koji se prezentuje suštinski shvatio, studentima se on treba prezentovati periodično, u različitim kontekstima i u rastućem nivou složenosti [1].

Uvođenje jednosemestralnih predmeta dovelo je do "podjele" ranijih planova i programa na dva ili više dijelova. Te podjele su u najvećem broju slučajeva podrazumijevale da se lekcije redom uključuju u prvi, odnosno drugi semestar. Sa druge strane, podjela plana i programa kursa na dva dijela može biti zasnovana i na drugačijim principima. Na primjer, jedan mogući pristup zasnovan je na konceptu da se, prije nego što se materijal podijeli na dva dijela, prvobitno odrede težine pojedinačnih lekcija, a da se podjela vodi principom da dijelovi budu ujednačene težine. Kako bi se održao kontinuitet u predavanju lekcija, uvodi se dodatni uslov, koji zahtijeva povezanost lekcija unutar svakog dijela. Principi podjele kursa na ovaj način prikazani su u [2]. Da bi se napravio matematički model koji odgovara ovom pristupu, lekcije kursa se posmatraju kao čvorovi grafa, dok se veze između lekcija (grane grafa) ostvaruju po nekoliko kriterijuma, kao što su sličnost, analogija, uopštenje ili uslovljenost. U [2] je pokazano da podjela lekcija kursa u dvije cjeline uz zahtjeve da svaka cjelina bude povezana i da težine te dvije cjeline budu u što većoj mjeri slične, odgovara matematičkom NP teškom problemu pronalaženja maksimalno balansirane povezane particije u grafu.

U ovom radu se ispituje drugačiji pristup podjele lekcija u jednog kursa u dva dijela, koji podrazumijeva da se što više koncepata (tematskih cjelina) pominje i obrađuje u oba dijela. Za dati kurs se, dakle, identifikuju tematske cjeline koje se obrađuju unutar njega, a podjela lekcija na dva dijela je vođena pristupom da što više tematskih cjelina bude "pokriveno" barem jednom lekcijom u svakom dijelu.

Pokazaće se da određivanje što bolje moguće raspodjele lekcija po opisanom principu odgovara poznatom optimizacionom problemu maksimalnog dijeljenja skupa (eng. Maximum Set Splitting problem - MSSP). Time se otvara mogućnost za rješavanje opisanog problema u nastavi već razvijenim metodama za rješavanje MSSP.

2 Polazne pretpostavke i definicija problema

Da bismo definisali problem i napravili matematički model za njega, polazimo od sljedećih pretpostavki. Materijal za neki kurs se prirodno dijeli po lekcijama. Pretpostavljamo da u datom kursu postoji više tematskih cjelina koje se u nekoj mjeri pokrivaju datim kursom. Jedna lekcija može da spada u različite tematske cjeline, a svaka tematska cjelina sadrži više lekcija. Sa metodičkog aspekta, postavljamo pretpostavku da bi bilo korisno napraviti podjelu lekcija u dvije cjeline (na primjer u zimski i ljetnji semestar), ali tako da što više tematskih cjelina bude prisutno u oba semestra. Time se omogućava da elementi svake tematske cjeline budu ponovljeni i obrađivani, kako je već i zahtijevano, u različitim kontekstima i nivoima složenosti.

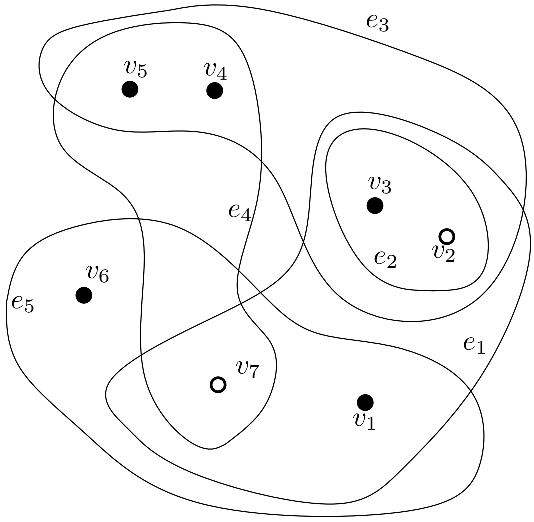
Da bismo konstruisali matematičku formulaciju problema, najprije lekcije $L_1, L_2, ..., L_m$ datog kursa posmatrajmo kao elemente skupa S. Tematske cjeline (oblasti) identifikujemo preko pripadajućih lekcija, tj. tematske cjeline posmatramo kao podskupove skupa lekcija. Neka je $C = \{S_1, S_2, ..., S_n\}$ kolekcija tematskih cjelina, tj. podskupova od S. Problem pronalaženja podjele (particije) skupa lekcija S na dva semestra, tako da je što više temastkih cjelina pokriveno u oba semestra je zapravo pronalaženje particije (P_1, P_2) skupa lekcija, tako da je broj podskupova iz kolekcije C koji imaju neprazan presjek i sa P_1 i sa P_2 najveći mogući.

Matematički model problema iz prethodnog pasusa, u kome su lekcije predstavljene elementima skupa S, a tematske cjeline kolekcijom C, podskupova skupa S, doslovce odgovara definiciji MSSP. MSSP je NP težak problem, što je istražujući neke invarijante hipergrafova, prvi dokazao L. Lovász u [3]. Hipergraf je uopštenje grafa, gdje za dati skup čvorova V, sa jednom granom, može biti incidentan proizvoljan broj čvorova iz V (ovakva grana naziva se hipergrana).

Problem pronalaženja minimalnog broja boja koje su potrebne da se oboje čvorovi grafa na taj način da nijedna grana nema sve čvorove obojene samo jednom bojom se pokazao podjednako teškim kao i određivanje hromatskog broja grafa. Poseban slučaj je bojenje čvorova u tačno dvije boje, a optimizaciona varijanta ovog problema se naziva "maksimalno 2 - bojenje hipergrafa" (eng. Max Hypergraph 2-Coloring - MH2C). Preciznije, 2 - bojenje hipergrafa je određivanje da li se svi čvorovi grafa mogu obojiti sa dvije broje (na primjer u bijelu i crnu), tako da svaka hipergrana ima najmanje po jedan čvor svake boje. Problem maksimalnog 2-bojenja hipergrafa je maksimizovanje broja hipergrana koje sadrže najmanje po jedan bijeli i jedan crni čvor.

Utvrđeno je da su MSSP i MH2C isti problemi. Zaista, za dati skup čvorova V u hipergrafu, svaka hipergrana definiše jedan podskup od V, koji sadrži elemente incidentne sa hipergranom. Tada je skup svih hipergrana zapravo jedna kolekcija C podskupova od V. Tako, određivanje 2-bojenja skupa čvorova kojim se maksimizuje broj hipergrana koje sadrže čvorove u obje boje je isti zadatak kao i pronalaženje particije skupa V, takve da svaki podskup iz C sadrži čvorove iz obje particije.

Primer 2.1. Neka je $S = \{1, 2, 3, 4, 5, 6, 7\}$ i $C = \{S_1, S_2, S_3, S_4, S_5\}$, $S_1 = \{1, 2, 3, 7\}$, $S_2 = \{2, 3\}$, $S_3 = \{2, 3, 4, 5\}$, $S_4 = \{4, 5, 7\}$, $S_5 = \{1, 6, 7\}$. Na slici 1 je prikazan hipergraf, sa čvorovima $\{v_1, v_2, ..., v_7\}$ koji odgovaraju elementima skupa S i hipergranama koje odgovaraju elementima iz C. Čvorovi v_2 i v_7 su obojeni bijelom bojom, a ostali crnom. Lako se vidi da svaka hipergrana ima barem po jedan čvor obojen u svaku boju. Tako je particija $(P_1, P_2) = (\{2, 7\}, \{1, 3, 4, 5, 6\})$ optimalna, jer svaki podskup sadrži bar po jedan element u svakoj particiji.



Slika 1: Primjer maksimalnog 2 - bojenja hipergrafa

3 Analiza upotrebe opisanog pristupa

Da bi se ilustrovao opisani pristup podjele kursa na dva dijela po principu MSSP, odabran je kurs iz Uvoda u računarstvo koji se predaje na prvoj godini studija na Studijskom programu Matematika i informatika Prirodno matematičkog fakulteta u Banjaluci. Ciljevi ovog kursa su:

(a) savladavanje osnovnih znanja iz arhitekture računara;

(b) savladavanje osnovnih znanja iz osnova programiranja, kroz programski jezik Python.

Ukupan fond časova predavanja je 90 (15 sedmica po 3 časa sedmično, u dva semestra).

Da bi se napravio odgovarajući matematički model za dati kurs na koji može da se primijeni neka tehnika za rješavanje MSSP, potrebno je da se urade tri zadatka:

(a) identifikacija lekcija;

(b) identifikacija tematskih cjelina;

(c) smještanje lekcija u odgovarajuće tematske cjeline.

3.1 Identifikacija lekcija

Na osnovu plana i programa, dostupnog i na internet stranicama Studijskog programa za matematiku i informatiku, (http://www.matinf.pmfbl.org), vidi se da je sadržaj čitavog kursa raspoređen u dva semestra od po 15 sedmica. S obzirom na fond od 3 časa sedmično, u zavisnosti od obima lekcija i težine gradiva, u svakoj sedmici se najčešće obrađuju po dvije ili tri lekcije. Na taj način, identifikovano je ukupno **76** lekcija, koje u matematičkom modelu predstavljaju elemente skupa koji se particioniše.

Spisak svih lekcija, prikazan je u tabeli 1. Zbog uštede prostora i bolje preglednosti, nazivi nekih lekcija su skraćeni. Po zvaničnom akreditovanom programu, čitav kurs iz Uvoda u računarstvo je podijeljen i organizovan u dva jednosemestralna kursa, Uvod u računarstvo 1 i Uvod u računarstvo 2. Trenutna podjela lekcija po semestrima odgovara redoslijedu lekcija u tabeli 1: prvih 39 lekcija su lekcije iz zimskog, dok su lekcije od rednog broja 40 do 76 iz ljetnjeg semestra. S obzirom na činjenicu da su lekcije već podijeljene po semestrima, cilj istraživanja prikazan u ovom odjeljku je dvojak:

(a) da se ustanovi koliko trenutna podjela kursa odgovara preporuci da što više tematskih cjelina budu prisutne u oba semestra;

(b) da se predloži preraspodjela lekcija, u odnosu na postojeću, kako bi se postigla bolja podjela, na osnovu formiranog matematičkog modela datog MSSP.

Tabela 1 – Spisak lekcija

R.b.	Naziv lekcije
1	Hardver i softver. CPU i glavna memorija. Input/Output jedinice. Kategorije softvera.
2	Analogno i digitalno. Digitalno predstavljanje podataka. Binarni brojevi.
3	Specifikacija računara. Memorija. Čuvanje podataka u memoriji. Vrste i kapacitet memorije.
4	Struktura CPU. Instrukcioni ciklus.
5	Mreže računara, načini povezivanja. Internet. TCP/IP. Domeni. WWW. HTML
6	Funkcija računarskog sistema. Struktura računarskog sistema. Organizacija i arhitektura računarskog sistema. Fon Nojmanova mašina.
7	Azbuka i kodovi. Brojčani sistemi. Prevođenje cijelih brojeva. Prevođenje razlomljenog dijela. Hijerarhija podataka u računaru. Zapis znakovnih podataka u računaru.
8	Sabiranje i oduzimanje u binarnom i heksadecimalnom sistemu. Prevođenje iz heksadecimalnog u dekadni sistem i obratno.
9	Zapis neoznačenih i označenih brojeva, nepotpuni i potpuni komplement.
10	Konverzija između zapisa različitih dužina, promjena znaka i sabiranje i oduzimanje neoznačenih brojeva, sabiranje i oduzimanje brojeva u nepotpunom i potpunom komplementu.
11	Množenje neoznačenih cijelih brojeva, množenje cijelih brojeva u nepotpunom i potpunom komplementu, dijeljenje cijelih brojeva.
12	Decimalna aritmetika: promjena znaka, sabiranje i oduzimanje. Realni brojevi u nepokretnom i pokretnom zarezu. Zapis sa heksadecimalnom osnovom.
13	IEEE standard 754. Sabiranje i oduzimanje u nepokretnom i poketnom zarezu. Množenje i dijeljenje u nepokretnom i pokretnom zarezu.
14	Zapis teksta, slike i zvuka.
15	Definicija programa, programiranja i računarskih nauka.
16	Instalacija i opis radnog okruženja za programski jezik Python. Proces formiranja programa u radnom okruženju. Izrazi u Python-u.
17	Aritmetički i logički izrazi. Tipovi u Python-u. Tipovi int i float. Prioritet operatora.
18	Varijable i operator dodjele vrijednosti. Složeni operatori i operator dodjele vrijednosti.

Tabela 1 – Spisak lekcija - nastavak

R.b.	Naziv lekcije
19	Greške i mehanizmi otkrivanja grešaka. Pojam funkcije i osnovne osobine funkcija. Lokalne i globalne varijable.
20	Ugrađene funkcije (eng. Build-in) jezika Python. Stilovi pri pisanju programa koji sadrže izraze i operator dodjele. Komentari u programima.
21	Stringovi i operacije nad stringovima. Escape sekvence. Stringovi u više linija. Komanda print(). Formatiranje izlaza. Korisnički ulaz i komanda raw_input().
22	Moduli i primjeri modula. Importovanje modula. Definisanje vlastitih modula. Šta se dešava prilikom importovanja modula. Programiranje help-a u Python-u.
23	Objekti i metode u Python-u.
24	Rad sa slikama. Pikseli i boje.
25	Liste. Liste i indeksi. Modifikovanje listi. Ugrađene funkcije za rad sa lstama. Obrada članova liste. Izdvajanje dijelova liste (eng. slicing). Inverz liste i palindromi. Pseudonimi (aliasi). Metode liste. Ugniježdene liste. Razne vrste nizova. Datoteke kao liste (predstavljanje datoteke listama). Argumenti komandne linije.
26	Komande izbora (grananja) u Python-u. Bulova logika. Bulovi operatori. Relacioni operatori. Primjena Bulovih operatora na int, float i string.
27	Komanda if. Ugnježdene komande grananja, operator dodjele i uslovi grananja.
28	Komande ponavljanja u Python-u. Komanda for. Funkcija range(). Funkcija enumerate().
29	Ugniježdene for petlje. Primjer obilaženja dvodimenzionalne matrice. Komanda while. Komande break i continue. Stilovi pri programiranju komandi grananja i petlji.
30	Datoteke. Otvaranje datoteka sa računara za čitanje, dodavanje ili pisanje. Otvaranje datoteka sa Interneta. Rad sa datotekom koja ima jedan slog u liniji. Slogovi sa više polja u liniji. Slogovi sa pozicioniranim poljima u liniji. Slogovi sa više linija. Look ahead pri čitanju datoteka.
31	Pisanje u datoteke.
32	Skupovi u Python-u. Rječnici. Invertovanje rječnika. Algoritmi. Euklid-ov, Wirth-ov i školski algoritam za NZD.
33	Rekurzija. Rekurzivni i iterativni algoritam za računanje faktorijala i Fibonacci-jevih brojeva.
34	Algoritmi pretraživanja. Poređenje programa po dužini vremena izvođenja. Pretraživanje i sortiranje.
35	Osnovno linearno pretraživanje i pretraživanje sa sentinel-om. Mjerenje vremena linearnog pretraživanja. Binarno pretraživanje.
36	Sort mjehurićima. Sort izborom. Sort umetanjem. Quick sort. Merge sort. Poređenje vremena rada algoritama sortiranja.
37	Konstrukcije. Dodatne osobine funkcija.
38	Izuzeci. Testiranje programa. Debagiranje programa.
39	Uzorci (eng. Patterns)
40	Logičke osnove obrade podataka: Bulova algebra, Pun sistem funkcija, SDNF i SKNF, Metode minimizacije logičkih funkcija, Logički elementi za obradu podataka, Kombinacione mreže, Sekvencijalne mreže, Primjeri.
41	Struktura savremenog računarskog sistema: Fon Nojmanova mašina, Sistem prekida, Brzina obrade podataka
42	Procesori: Organizacija centralnog procesora, Registri, Tehnologije izrade mikroprocesora, Tranzistori, Tehnologije izrade brzih čipova, CISC i RISC arhitektura mikroprocesora, Osobine RISC procesora.
43	Mašinske instrukcije: Karakteristike mašinskih instrukcija, Formati i tipovi instrukcija, Broj adresa u instrukciji. Načini adresiranja u mašinskim instrukcijama
44	Mašinski i asemblerski jezici, Pozivi podprograma, Smještanje podataka u memoriji.
45	Memorije i ulazno/izlazni podsistem: Unutrašnja memorija, Karakteristike, Keš memorija, Spoljašnje memorije, Magnetni i optički mediji,
46	U/I moduli, Tehnike izvršavanja U/I operacija, U/I uređaji i njihove karakteristike
47	Otkrivanje i korekcija grešaka: Metode za otkrivanje i metode za otkrivanje i korekciju grešaka
48	Pretraživanje: Nalaženje minimuma u listi, Prolaženje kroz listu, Mjerenje vremena pretraživanja,
49	Linearno pretraživanje: while i for verzije linearnog pretraživanja, Sentinel, Tajming linearnog pretraživanja.
50	Binarno pretraživanje: Složenost algoritama binarnog pretraživanja, ugrađena funkcija binarnog pretraživanja
51	Algoritmi, Program
52	Složenost sortova u najboljem i najgorem slučaju: Sort mjehurićima (eng. Bubble sort), Sort umetanjem (eng. Insert sort), Sort izborom (eng. Choice sort).

Tabela 1 – Spisak lekcija - nastavak

R.b.	Naziv lekcije
53	Sort spajanjem (eng. Merge sort), Brzi sort (eng. Quick sort)
54	Definnicije funkcija: Podrazumijevani (eng. default) parametri, Liste parametara, Parametri sa nazivima.
55	Izuzeci: tray i except, except objekata. Izuzeci: Funkcije i izuzeci, raise izuzetak, Still izuzetka.
56	Testiranje: Funkcionalni test, unit test, black-box test, glass-box test, Nezavisnost, Ograničenja, Testom vođen razvoj programa. Debagiranje.
57	Obrasci dizajniranja: Fiksne vrijednosti, Steperi i kaunteri, Najpoželjniji nosilac, Zadnji nosilac,
58	Kontejner, Akumulator, Temporalna varijabla, Zastavica. Klase: Atributi, Metode, Konstruktori.
59	Specijalne metode: init ; str ; repr ; add ; sub Metode dir i help.
60	Inkapsulacija u objektno orijentisanom programiranju (eng. Encapsulation)
61	Polimorfizam i nasljeđivanje: ad hock polimorfizam, Parametarski polimorfizam, Hijerarhijski polimorfizam.
62	Nasljeđivanje: Primjeri class Atom i class Molecule.
63	GUI: Osnovni pojmovi, Event-driven programiranje. Modul Tkinter: Widgets: Button, Canvas, Checkbutton, Entry, Frame, Label, Listbox, Menu, Message, Menubutton, Text, TopLevel.
64	Osnovne GUI konstrukcije: Frame, Entry, Laki i teški procesi. Mutable variables: intVar, StringVar, BooleanVar, DoubleVar.
65	Modeli, pogledi, kontroleri: Primjer. Korištenje Lambda: Primjer.
66	Stil pisanja GUI: Font, Color, Layout. Widget-i: Text, Checkbutton, Menu.
67	Objektno-orijentisani GUI: Primjer.
68	Baze podataka: Arhitektura baza podataka , sqlite3, MySQLdb, SQLAlchemy. sqlite3 funkcije: connect(), cursor(), execute(), commit().
69	SQL tipovi podataka: NULL, INTEGER, REAL, TEXT, BLOB. Tabele u SQL: CREATE TABLE TableName(ColumnName Type, ColumnName Type, ...), INSERT INTO VALUES.
70	Nalaženje podataka u Bazama podataka: SELECT FROM, fetchone(), fetchall(), SELECT FROM ORDER BY. Upiti sa uslovima: SELECT FROM WHERE, =; ! =; >; <;>=;<=.
71	Apdejtovanje i brisanje u bazama podataka: UPDATE SET WHERE, DELETE FROM WHERE, DROP TABLE.
72	Transakcije u Bazama podataka. NULL kao zamjena za nepostojeće podatke. Korištenje JOIN za kombinovanje tabela(INNER JOIN).
73	Ključevi i ograničenja u Bazama podataka. Napredne tehnike Baza podataka: Agregacija, Grupisanje, SelfJoins, Ugniježdeni upiti.
74	Mrežni moduli : Socket. Klijent/Server par: Primjer. Mrežni moduli urllib i urllib2: Otvaranje udaljenih datoteka, povraćaj udaljenih datoteka.
75	Opis funkcija nekih mrežnih modula 1: asynchat, asyncore, cgi, Cookie, cookielib, email, ftplib,gopherlib, httplib, imaplib.
76	Opis funkcija nekih mrežnih modula 2: mailbox, mailcap, mhlib, nntplib, poplib, robotparser, impleXMLRPCServer, smtpd, smtplib, telnetlib, urlparse, xmlrpclib.

3.2 Identifikacija tematskih cjelina

Određivanje tematskih cjelina koje pokrivaju čitavu oblast računarskih nauka predstavlja zahtjevan posao. U današnje vrijeme, računarske nauke su podložne raznim uticajima i s obzirom na ogromnu količinu naučnog, obrazovnog, komercijalnog, zabavnog i drugog dostupnog informatičkog sadržaja, srećemo prilično heterogene klasifikacije i ontologije naučnih računarskih oblasti.

Kako bi se osigurao objektivan pristup, sa jedne strane, i kompletnost i koreknost sa druge, kao glavna referenca za određivanje tematskih cjelina u ovom radu korišteni su resursi objavljeni od strane udruženja *Association for Computing Machinery* (*ACM*), jednog od najpoznatijih i najvećih svjetskih obrazovnih i naučnih udruženja iz oblasti računarskih nauka. Jedan od zadataka ove organizacije je kontinuirana isporuka materijala koji sadrži preporuke za kreiranje

Tabela 2: Spisak oblasti znanja

Oblast	Oblast (eng.)	skr.	ob.	iz.
Diskretne strukture	Discrete Structures	DS	6	0
Osnove programiranja	Programming Fundamentals	PF	8	0
Algoritmi i kompleksnost	Algorithms and Complexity	AL	5	6
Arhitektura i organizacija	Architecture and Organization	AR	6	4
Operativni sistemi	Operating Systems	OS	6	8
Mrežno računarstvo	Net-Centric Computing	NC	3	6
Programski jezici	Programming Languages	PL	6	5
Interakcija čovjeka i računara	Human-Computer Interaction	HC	2	8
Grafika i vizuelno računanje	Graphics and Visual Computing	GV	2	11
Inteligentni sistemi	Intelligent Systems	IS	3	8
Upravljanje informacijama	Information Management	IM	3	12
Društveni i profesion. aspekti	Social and Professional Issues	SP	7	4
Softverski inženjering	Software Engineering	SE	8	6
Nauka o izračunavanju	Computational Science	CN	0	3

planova i programa računarskih kurseva, kao i čitavih studijskih programa, kako bi se uspješnije pratili nagli napredak i brze promjene u oblasti računarstva.

U ovom radu se citira posljednji izvještaj [4] iz 2008. godine, u kome je ACM u saradnji sa odjelom za obrazovanje IEEE udruženja (*IEEE Computer Society Education Activities Board*) predstavio novi Vodič za formiranje planova i programa iz oblasti kompjuterskih nauka, zasnovan na prethodnom izvještaju iz 2001. godine i novim saznanjima koja su stečena u međuvremenu. Ovaj izvještaj pruža validnu osnovu za kreiranje planova studijskih programa iz oblasti računarskih nauka, kao i rasporeda pojedinačnih kurseva i koristi se u mnogim visokoškolskim ustanovama širom svijeta. Kao osnovni doprinos prikazan u ovom izvještaju, identifikovano je tzv. "tijelo znanja" (eng. body of knowledge), koje sadrži sve relevantne računarske oblasti, podijeljene u 14 "oblasti znanja" (eng. knowledge areas). U tabeli 2 prikazane su te oblasti, sa nazivima na našem i engleskom jeziku, skraćenim nazivom na engleskom jeziku, zajedno sa brojem ključnih, odnosno izbornih oblasti.

Svaka od ovih oblasti znanja podijeljena je na tematske cjeline, označene kao ključne ili izborne. Ukupno je predloženo 146 tematskih cjelina, od kojih su 65 ključne, a 81 izborne. U zavisnosti od toga koliki se značaj posvećuje posmatranoj cjelini, u izvještaju [4] je predložen i broj časova koje treba posvetiti ključnim tematskim cjelinama. Prirodno, fundamentalnim oblastima je data veća važnost, tj. predložen je veći broj ključnih tematskih cjelina i veći predložen broj časova. To su, prije svih, oblasti: diskretne strukture, osnove programiranja, algoritmi i kompleksnost, arhitektura i organizacija, operativni sistemi, programski jezici, softverski inženjering, te društveni i profesionalni aspekti. Za ostale oblasti znanja predložen je manji broj sati i manji broj ključnih oblasti. Time se kreatorima planova i programa omogućava veća fleksibilnost i prilagođavanje specifičnim potrebama.

Tabela 3: Spisak tematskih cjelina i pripadajućih lekcija

R.b.	Naziv tematske cjeline	Oblast (eng.)	Spisak lekcija
1	Funkcije, relacije i skupovi	DS	18,19,32,33,61,37
2	Osnovna logika	DS	17,26,40,70
3	Tehnike dokazivanja	DS	17,19,33,40
4	Osnove prebrojavanja	DS	7,9,17,24,40,48
5	Grafovi i stabla	DS	33,35,50,74
6	Diskretna vjerovatnoća	DS	36,52,53
7	Osnove konstrukcija	PF	2,17,18,20,27,28,29,30,54
8	Algoritamsko rješavanje problema	PF	15,32,33,34,35,38,47,51,56
9	Strukture podataka	PF	17,21,25
10	Rekurzija	PF	33,35,37,50
11	Programiranje zasnovano na događajima	PF	38,55,63
12	Objektno orijentisana paradigma	PF	23,58,60,61,62,67
13	Sigurno programiranje	PF	19,47,55
14	Osnovna analiza algoritama	AL	32,36,48,52,53
15	Algoritamske strategije	AL	20, 32,34,35,39,50
16	Osnovni algoritmi	AL	32,35,36,48,49,50
17	Digitalna logika i reprezentacija podataka	AR	2,3,7,8,9,10,11,12,13,14
18	Računarska arhitektura i organizacija	AR	1,3,4,6,40,41,42,43,44
19	Interfejsi i U/I strategije	AR	1,30,46
20	Arhitektura memorije	AR	1,3,44,45
21	Funkcionalna organizacija	AR	1,3,4,42,43
22	Uređaji	AR	2,14,24
23	Osnovni pregled operativnih sistema	OS	1,45,46
24	Mrežne komunikacije	NC	5,74,75,76
25	Pregled programskih jezika	PL	15,16,58,59,60,61,62
26	Osnove prevođenja jezika	PL	15,16,19,20,68,69
27	Deklaracije i tipovi	PL	17,18,20,58
28	Mehanizmi apstrakcije	PL	22,57,58,65
29	Objektno orijentisano programiranje	PL	16,58,60,61,62,63,64,65,67
30	Osnove interakcije čovjek-računar	HC	16,63,66
31	Izgradnja GUI interfejsa	HC	63,64,65,66,67
32	Evaluacija korisnički orijentisanih sistema	HC	47,56,57
33	Dizajn GUI	HC	65,66
34	Osnove tehnike grafike i vizuelnog računanja	GV	14,24
35	Informacioni modeli	IM	60,68
36	Jezici upita	IM	68,69,70,71,72,73
37	Istorija računarstva	SP	2,15,41

3.3 Smještanje lekcija u tematske cjeline

Predloženo "tijelo znanja" pokriva čitavu oblast računarskih nauka, dok je za potrebe istraživanja i grupisanja lekcija kursa Uvod u računarstvo u odgovarajuće tematske cjeline broj tematskih oblasti redukovan u značajnoj mjeri. Pored uvodnih fundamentalnih tematskih oblasti, uvrštene su uvodne tematske oblasti koje se odnose na informacione sisteme, interakciju čovjeka i računara, računarske mreže i slično. Preostale oblasti, koje se odnose na napredne kurseve iz računarstva, su izostavljene.

Informacije o razmatranim tematskim cjelinama, kao i pregled odgovarajućih lekcija prikazane su u tabeli 3. Prve tri kolone sadrže redni broj, naziv tematske cjeline i skraćeni naziv oblasti znanja kojoj tematska cjelina pripada, dok je u krajnjoj desnoj koloni naveden spisak rednih brojeva pripadajućih lekcija. Redni brojevi lekcija odgovaraju rednim brojevima iz tabele 1. Kao što se vidi iz

Tabela 4: Raspored broja lekcija po tematskim cjelinama

Broj lekcija	Broj tematskih cjelina sa datim brojem lekcija
2	3
3	10
4	8
5	3
6	7
7	1
8	0
9	4
10	1

tabele 3, lekcije su svrstavane samo u one tematske cjeline za koje se pouzdano može smatrati da se problematika tematske cjeline obrađuje unutar lekcije, a ne samo usputno pominje. Primjećujemo da su sve fundamentalne tematske cjeline pokrivene većim brojem lekcija, dok se neke druge obrađuju samo u manjem broju lekcija. U tabeli 4 sistematizovana je raspodjela lekcija po tematskim cjelinama. Vidimo da je za 21 tematsku cjelinu broj lekcija u kojima se one obrađuju manji ili jednak od 4, za 11 tematskih cjelina važi da se obrađuju u 5, 6 ili 7 lekcija, dok se 5 tematskih cjelina obrađuje u 9 i više lekcija.

3.3.1 Analiza trenutne podjele lekcija i prijedlog poboljšanja

Analizirano je koliko trenutna podjela lekcija odražava polaznu preporuku da što više tematskih cjelina bude obrađeno u oba semestra. Ponovimo da se po trenutnoj raspodjeli, (tabela 1), lekcije sa rednim brojevima 1-39 obrađuju u zimskom semestru, a lekcije 40-76 u ljetnjem. Prema rasporedu lekcija po tematskim cjelinama prikazanom u tabeli 3, od ukupno 37 tematskih cjelina koje su razmatrane, za 30 važi da se one obrađuju u bar po jednoj lekciji u svakom semestru, dok za 7 tematskih cjelina taj uslov ne važi. Primijetimo da algoritam kojim se određuje broj podijeljenih tematskih cjelina odgovara algoritmu verifikacije rješenja odgovarajućeg MSSP. U prve dvije kolone tabele 5 su prikazani redni brojevi i imena tematskih cjelina za koje uslov podijeljenosti ne važi. U narednoj koloni je prikazan skraćeni engleski naziv odgovarajuće oblasti znanja, a u krajnjoj desnoj koloni informacija u kom semestru je cjelina prisutna.

Kao što vidimo iz tabele 5, od sedam "nepodijeljenih cjelina", po jedna se odnosi na arhitekturu računara, te grafiku i vizuelno računanje, dok se tri odnose na interakciju čovjeka i računara i dvije na upravljanje informacijama. Primjećujemo takođe da se oblastima koje se odnose na interakciju čovjeka i računara i upravljanju informacijama pažnja posvećuje u ljetnjem semestru, dok u zimskom semestru ovoj problematici nije posvećena značajnija pažnja. Stoga bi pravac

Tabela 5: Spisak cjelina za koje ne važi traženi uslov

R.b.	Tematska cjelina	Oblast (eng.)	Prisutna samo u
22	Uređaji	AR	zimskom
31	Izgradnja GUI interfejsa	HC	ljetnjem
32	Evaluacija korisnički orijentisanih sistema	HC	ljetnjem
33	Dizajn GUI	HC	ljetnjem
34	Osnove tehnike grafike i vizuelnog računanja	GV	zimskom
35	Informacioni modeli	IM	ljetnjem
36	Jezici upita	IM	ljetnjem

Tabela 6: Sistematizacija predloženih rješenja

Postupak	Efekat na tematske cjeline	Posljedica na ukupno rješenje
Prebacivanje lekcije 24 u ljetnji semestar	Tematske cjeline 22 i 34 će biti podijeljene	Smanjenje broja nepodijeljenih lekcija za 2
Prebacivanje lekcije 56 u ljetnji semestar	Tematska cjelina 32 će biti podijeljena	Smanjenje broja nepodijeljenih lekcija za 1
Prebacivanje lekcije 65 u zimski semestar	Tematske cjeline 31 i 33 će biti podijeljene	Smanjenje broja nepodijeljenih lekcija za 2
Prebacivanje lekcije 68 u zimski semestar	Tematske cjeline 35 i 36 će biti podijeljene	Smanjenje broja nepodijeljenih lekcija za 2

unapređenja Plana i programa kursa mogao da se zasniva na uključivanju po jedne lekcije iz grafičkog korisničkog okruženja, te baza podataka u zimskom semestru. Pažljivom analizom trenutnog rasporeda lekcija, može se primijetiti da bi se prebacivanjem polazne lekcije u vezi sa bazama podataka i jedne lekcije u vezi sa interakcijom čovjeka i računara u zimski semestar, mogao riješiti problem "nepodijeljenosti" cjelina 35 i 36, odnosno 31 i 33, a da se ne naruše drugi metodički aspekti, kao što su kontituitet ili zavisnost lekcija.

Takođe, može se zaključiti da bi se razmjenom lekcija 24 i 56 ("Rad sa slikama. Pikseli i boje." odnosno "Testiranje: Funkcionalni test, *unit test*, *black-box test*, *glass-box test*, Nezavisnost, Ograničenja, Testom vođen razvoj programa. Debagiranje.") postigla situacija da su cjeline 22, 32 i 34 podijeljene, a podijeljenost preostalih tematskih cjelina se ne bi narušila.

Sistematizacija predloženih unapređenja prikazana je u tabeli 6.

Ovom preraspodjelom lekcija dobija se stanje da su sve tematske cjeline podijeljene.

4 Pregled razvijenih metoda za rješavanje MSSP

U ovom odjeljku dat je pregled metoda rješavanje MSSP problema koje se sreću u literaturi. Pošto je problem NP težak, ne postoji deterministrički algoritam koji rješava opšti slučaj problema u polinomnom vremenu. Stoga, egzaktne metode uspijevaju da riješe samo probleme manjih dimenzija, jer vrijeme izvršenja algoritma eksponencijalno raste porastom dimenzije. Za probleme većih dimenzija primjenjuju se heurističke metode. Opisane su dvije formulacije cjelobrojnog programiranja (kvadratna i linearna formulacija), te tri metaheurističke metode, zasnovane na genetskom algoritmu, elektromagnetizmu i metodi promjenljivih okolina.

Prije nego što se navedu metode za rješavanje MSSP, uvedimo sljedeću notaciju. Neka je S konačan skup kardinalnosti $m = |S|$ i neka je $C = \{S_1, S_2, ..., S_n\}$ kolekcija podskupova od S. Bez gubljenja opštosti, možemo pretpostaviti da je $S = \{1, 2, ..., m\}$. Neka je (P_1, P_2) particija od S. Za skup $S_j \in C$ kažemo da je "podijeljen", ako S_j ima neprazan presjek i sa P_1 i sa P_2.

4.1 Kvadratna cjelobrojna formulacija

O ovom pododjeljku prikazana je kvadratna cjelobrojna formulacija iz [5].

Kvadratna cjelobrojna formulacija (1)-(4) uvodi dva skupa promjenljivih:

$$y_i = \begin{cases} 1, & i \in P_1 \\ -1, & i \in P_2 \end{cases}, \quad \text{za } i = 1, ..., m,$$

i

$$z_j = \begin{cases} 1, & S_j \text{ je podijeljen} \\ 0, & S_j \text{ nije podijeljen} \end{cases}, \quad \text{za } j = 1, ..., n.$$

QIP se definiše kao

$$\max \sum_{j=1}^{n} z_j \tag{1}$$

uz ograničenja

$$\frac{1}{|S_j| - 1} \sum_{\substack{i_1, i_2 \in S_j \\ i_1 \neq i_2}} \frac{1 - y_{i_1} \cdot y_{i_2}}{2} \geq z_j, \quad \text{za sve } S_j \in C \tag{2}$$

$$z_j \in \{0, 1\}, \quad \text{za } j = 1, ..., n \tag{3}$$

$$y_i \in \{-1, 1\}, \quad \text{za } i = 1, ..., m \tag{4}$$

4.2 Linearna cjelobrojna formulacija

Ovaj pododjeljak sadrži opis linearne cjelobrojne formulacije, prikazane u [6].

Za ILP uvodimo parametre

$$s_{ij} = \begin{cases} 1 & i \in S_j \\ 0 & i \notin S_j \end{cases} \qquad \text{za } i = 1, ..., m, \ j = 1, ..., n,$$

koji označavaju da li element i iz S pripada podskupu S_j. Promjenljive se definišu kao

$$x_i = \begin{cases} 1 & i \in P_1 \\ 0 & i \in P_2 \end{cases}$$

$$y_j = \begin{cases} 1 & S_j \text{ je podijeljen} \\ 0 & S_j \text{ nije podijeljen} \end{cases}$$

ILP model za MSSP se formuliše kao

$$\max \sum_{j=1}^{n} y_j \tag{5}$$

uz ograničenja

$$y_j \le \sum_{i=1}^{m} s_{ij} \cdot x_i, \qquad \text{za sve } j = 1, ..., n \tag{6}$$

$$y_j + \sum_{i=1}^{m} s_{ij} \cdot x_i \le |S_j| \quad , \qquad \text{za sve} j = 1, ..., n \tag{7}$$

$$y_j \in \{0, 1\} \quad , \qquad \text{za sve } j = 1, ..., n \tag{8}$$

$$x_i \in \{0, 1\} \quad , \qquad \text{za sve } i = 1, ..., m \tag{9}$$

Lako se vidi da ILP formulacija raspolaže sa $m+n$ binarnih promjenljivih i ukupno $2 \cdot n$ ograničenja.

4.2.1 Genetski algoritam za rješavanje MSSP

Genetski algoritam (GA) za rješavanje MSSP prikazan je zajedno sa cjelobrojnom linearnom formulacijom u [6].

GA spadaju u klasu evolucijskih algoritama, čije ponašanje potiče iz procesa evolucije koji se odvija u prirodi. U osnovi se podrazumijeva rad sa populacijom jedinki, gdje svaka jedinka predstavlja potencijalno rješenje, a svaka populacija je podskup ukupnog prostora pretraživanja. Populacija se u iterativnom postupku mijenja tako što se stare jedinke mijenjaju novim, potencijalno boljim.

GA za rješavanje MSSP koristi binarno kodiranje, dok je genetski kôd one dužine koliko ima elemenata skupa. Vrijednost 1 na i-tom mjestu genetskog kôda označava da odgovarajući element i pripada skupu P_1, dok vrijednost 0 označava da dati element pripada skupu P_2. Za datu jedinku, funkcija cilja ima vrijednost broja podijeljenih skupova kolekcije C. Fitnes jedinke f_{ind} se računa skaliranjem vrijednosti funkcije cilja svih jedinki (kojih u populaciji ima N_{pop} u interval [0,1]. Tako najbolja jedinka ind_{max} ima fitnes jednak 1, dok je vrijednost fitnesa najmanje jedinke (ind_{min}) jednak nuli. Preciznije,

$$f_{ind} = \frac{obj_{ind} - obj_{ind_{min}}}{obj_{ind_{max}} - obj_{ind_{min}}}$$

GA implementira standardni genetski operator jednopozicionog ukrštanja, dok za operator selekcije koristi unaprijeđenu turnirsku selekciju, koja omogućava da prosječna veličina turnira bude proizvoljan realan (racionalan) broj. Standardni operator mutacije je unaprijeđen konceptom prepoznavanja tzv. "smrznutih gena", tj. pozicija u genetskom kodu na kojima sve jedinke populacije imaju istu vrijednost. Pojava smrznutih gena uzrokuje drastično smanjenje prostora pretrage, što generalno povećava vjerovatnoću za preuranjenu konvergenciju, dok operatorima selekcije i ukrštanja "smrznuti geni" ne mogu promijeniti vrijednost. U slučaju kodiranja GA za MSSP "smrznuti gen" ukazuje da je dati element (koji odgovara tom genu) u svim jedinkama populacije dodijeljen istom skupu (P_1 ili P_2). Mehanizam kojim se ova pojava eliminiše i koji je implementiran u dati GA, zasnovan je na značajnom povećanju vjerovatnoće za mutaciju "smrznutih gena". Stoga se za "smrznute gene" osnovni nivo mutacije množi odgovarajućim brojem, faktorom mutacije za smrznute gene.

4.3 Elektromagnetizam za rješavanje MSSP

Elektromagnetizam (EM) za rješavanje MSSP prikazan je u [7]. EM je metaheuristička metoda koja koristi mehanizam privlačenja - odbijanja čestica, kako bi se realizovalo kretanje i eventualno dostizanje optimalnog rješenja. Svaka tačka (čestica) se smatra potencijalnim rješenjem, kome je dodijeljen neki nivo naelektrisanja. Bolje čestice imaju jaču silu privlačenja drugih čestica. Tačna vrijednost uticaja jedne čestice na drugu definisana je Kulonovim zakonom. To znači da je sila između dvije tačke proporcionalna proizvodu naelektrisanja i obrnuto proporcionalna rastojanju između njih. Drugim riječima, tačke sa višim naelektrisanjem će više uticati na kretanje drugih čestica. Pored toga, najbolja EM tačka će ostati nepromijenjena. Naelektrisanja tačaka se vezuju za vrijednost funkcije cilja, koja je predmet optimizacije.

EM za rješavanje MSSP koristi hibridni pristup. Kombinuje se osnovni mehanizam "pomjeranja čestica", zasnovan na principu privlačanja i odbijanja čestica, zajedno sa specifično osmišljenom tehnikom skaliranja, koja usmjerava čitav EM u regione pretraživanja za koje se pretpostavljada sadrže kvalitetna rješenja.

Ako je m ukupan broj elemenata skupa, svaka tačka p_i EM algoritma se kodira nizom od m elemenata, gdje su elementi iz intervala $[0, 1]$. Da bi se za svaku tačku p_j odredila funkcija cilja, prvo se odgovarajuća particija (P_1, P_2) odredi zaokruživanjem na sljedeći način: ako je i-ta koordinata tačke p_j veća ili jednaka

0.5, tada se element i pridružuje P_1, a u protivnom P_2. Uvedimo promjenljivu y definisanu na sljedeći način:

$$y_i = \begin{cases} 1, & p_{ji} \geq 0.5 \\ 0, & p_{ji} < 0.5 \end{cases}$$

Ako uvedemo i promjenljivu z definisanu na sljedeći način:

$$z_j = \begin{cases} 1, & S_j \text{ je podijeljen} \\ 0, & S_j \text{ nije podijeljen} \end{cases}$$

tada se vrijednost funkcije cilja definiše kao

$$obj(P_1, P_2) = \sum_{j=1}^{n} z_j. \tag{10}$$

Brzo lokalno pretraživanje dodatno unapređuje cjelokupnu pretragu, omogućavajući pronalaženje boljeg rješenja u nekoj okolini trenutnog. Zasnovano je na zamjeni komponenti parova elemenata skupa i koristi strategiju "primjene prvog poboljšanja", tj. strategiju koja podrazumijeva da se otkriveno unapređenje odmah primjenjuje na trenutno rješenje.

Mehanizam privlačenja-odbijanja omogućava kretanje EM tačaka po prostoru pretrage, tako što se pretraga usmjerava ka potencijalno boljim oblastima, gdje se nalaze tačke sa boljim vrijednostima funkcije cilja. U ovom procesu pomjeranja, tačke EM se posmatraju kao naelektrisane čestice. Vrijednost naelektrisanja je u direktnoj vezi sa vrijednošću funkcije cilja i računa se pomoću formule (11).

$$q_i = exp\left(-N_{pop}\frac{f(p_{best}) - f(p^i)}{\sum_{k=1}^{N_{pop}} f(p_{best}) - f(p^k)}\right),\ i = 1, ..., N_{pop} \tag{11}$$

gdje je N_{pop} ukupan broj EM tačaka. Sila između parova tačaka se računa po sličnom principu kao u teoriji elektromagnetizma za naelektrisane čestice. Sila između dvije čestice je obrnuto propocionalna udaljenosti čestica, a direktno proizvodu njihovih naelektrisanja. Tačke sa boljim vrijednostima funkcije cilja privlače druge tačke, dok one sa lošijim odbijaju ostale. Vrijednost sile privlačenja/odbijanja prikazana je formulom (12):

$$F_i = \begin{cases} \sum_{j=1, j\neq i}^{m} (p_j - p_i)\frac{q_i \times q_j}{||p_j - p_i||^2}; & f(p_j) > f(p_i) \\ \sum_{j=1, j\neq i}^{m} (p_i - p_j)\frac{q_i \times q_j}{||p_j - p_i||^2}; & f(p_j) \leq f(p_i) \end{cases} \tag{12}$$

Nakon što se izračunaju sile privlačenja/odbijanja, sve EM tačke osim najbolje se kreću u odgovarajućim smjerovima. Najbolja tačka se ne pomjera. U EM algoritmu, pomjeranjem tačaka se upravlja unutar procedure Pomjeri. Pravac i smjer vektora u kom pravcu se tačka p_i pomjera je određen silom F_i. Sve sile se najprije normalizuju ($F_i = \frac{F_i}{||F_i||}$). Kretanje tačaka je dalje definisano pomoću formule 13, gdje je β slučajan broj iz intervala $[0,1]$.

$$p_i^k = \begin{cases} p_i^k + \beta \frac{F_i^k}{||F_i||}(1 - p_i^k), & F_i^k > 0 \\ p_i^k + \beta \frac{F_i^k}{||F_i||} \cdot p_i^k, & F_i^k < 0 \end{cases} \tag{13}$$

4.4 Metoda promjenljivih okolina za rješavanje MSSP

Metoda promjenljivih okolina (eng. Variable Neighborhood Search - VNS) je metaheuristička metoda koja podrazumijeva sistematsku promjenu okolina, kako bi se izbjegle situacije kada algoritam "upada" u suboptimalna rješenja. Efikasnost VNS algoritma je zasnovana na razmatranju da u mnogim praktičnim optimizacionim problemima postoji veza između lokalnih minimuma. Stoga metod promjenljivih okolina i ne prati unaprijed zadatu putanju, već "skače" na potencijalno bolja rješenja koja se nalaze u nekoj okolini trenutno najboljeg. Na ovaj se način očuvavaju dobre osobine trenutnog rješenja (ako su neke promjenljive već dostigle optimalne vrijednosti), dok se u novim okolinama pokušavaju poboljšati preostale promjenljive. Ovaj sistem je dodatno ojačan sistemom lokalnog pretraživanja, kojim se prelazi iz trenutnog rješenja u najkvalitetnije rješenje u nekoj njegovoj okolini. Metoda promjenljivih okolina za rješavanje MSSP prikazan je u [8].

Kodiranje rješenja je slično kao i kodiranje jedinki GA opisanog u [6]. Rješenje je predstavljeno binarnim nizom dužine koliko ima elemenata skupa. Vrijednost 1 na i-tom mjestu niza označava da odgovarajući element i pripada skupu P_1, dok vrijednost 0 označava da dati element pripada skupu P_2. Polazno rješenje se bira na slučajan način.

Za dato rješenje, funkcija cilja ima vrijednost broja podijeljenih skupova kolekcije C. Na osnovu kodarješenja, lako se određuje kojoj komponenti pripada koji element. Nakon toga se, slično kao u EM algoritmu, za datu particiju (P_1, P_2), i za svaki element (podskup) $S_j \in C$, određuje vrijednost z_j

$$z_j = \begin{cases} 1, & S_j \text{ je podijeljen} \\ 0, & S_j \text{ nije podijeljen} \end{cases}$$

a zatim i funkcija cilja

$$obj(P_1, P_2) = \sum_{j=1}^{n} z_j. \tag{14}$$

Određivanje novih okolina, kao i predlaganje novih potencijalno boljih rješenja iz tih okolina se odvija u proceduri razmrdavanja (eng. shaking). U okviru date procedure razmrdavanja, sistem okolina se formira na sljedeći način. Za definisanje k-te okoline, u algoritmu se na slučajan način bira nekih k elemenata iz S. Svakom izabranom elementu mijenja se pripadajuća komponenta, tj. svi izabrani elementi koji pripadaju P_1 se prebacuju u P_2 i obrnuto. Formalno, k-ta okolina vektora (rješenja) x se može zapisati kao $N_k(x) = \{x' : \{i_1, i_2, ..., i_k\} \subset \{1, 2, ..., |S|\}\ x'_{i_j} = 1 - x_{i_j}\}$. U algoritmu se tako unutar procedure razmrdavanja za svaki cio broj k, $k \in [k_{min}, k_{max}]$ predlaže novo rješenje x'. ($x' \in N_k(x)$), na koje se dalje poziva lokalno pretraživanje.

U svakoj iteraciji lokalne pretrage algoritam pokušava da popravi rješenje tako što mijenja pripadajuće komponente za parove elemenata iz S. Na primjer, ako $a \in P_1$ i $b \in P_2$, tada nakon razmjene imamo $a \in P_2$ i $b \in P_1$. U slučaju da je tako dobijeno novo rješenje x'' bolje od polaznog rješenja x' iz date okoline tačke x, tada x' postaje jednako x''. Lokalna pretraga prestaje sa radom nakon prvog takvog poboljšanja. U drugom slučaju, ako nije došlo do poboljšanja (x'' nije bolje od x'), lokalna pretraga nastavlja sa sljedećim parom elemenata. Po završetku lokalne pretrage, porede se vrijednosti funkcije cilja za x i x' i u zavisnosti koje je bolje, ono se smatra trenutno najboljim rješenjem. U slučaju da su vrijednosti funkcije cilja za x i x' jednake, tada algoritam prelazi u rješenje x' sa vjerovatnoćom 0.4, a u protivnom ostaje u rješenju x.

Eksperimentalni rezultati koji prikazuju kvalitet i upotrebljivost opisanih metoda za rješavanje MSSP pokazuju da date metode uspijevaju da pronađu kvalitetna rješenja u razumnom vremenu. To ukazuje na činjenicu da se opisane metode mogu koristiti i za rješavanje konkretnih problema koji se odnose na podjelu kurseva po opisanom kriterijumu.

5 Zaključak

U ovom radu je prikazana mogućnost unapređenja planova i programa primjenom poznatog matematičkog problema maskimalne podjele skupa. Ako razmatramo metodološki opravdan pristup podjele kursa na dva dijela (na primjer na dva semestra), na način da što više tematskih cjelina bude prisutno u oba dijela, pokazuje se da je matematički model koji odgovara ovom pristupu identičan postavci poznatog matematičkog problema MSSP. Kao ilustraciju upotrebe MSSP u rješavanju problema u nastavi, prikazana je podjela kursa Uvod u računarstvo i analizirana trenutno važeća podjela. Analizom se došlo do saznanja da trenutna podjela kursa u značajnoj mjeri "prati" propisani pristup, dok se minornim razmjenama lekcija između semestara, taj raspored može učiniti i optimalnim. Ova činjenica ukazuje na pretpostavku da su autori Plana i programa, motivišući se ovim ili nekim sličnim pristupom, u značajnoj mjeri vodili računa da se fundamentalne cjeline razmatraju u oba semestra, kako bi se očuvao kontinuitet u izučavanju.

Pošto je odgovarajući matematički problem NP težak, egzaktne metode se mogu koristiti samo pri rješavanju problema raspoređivanja lekcija u semestre koji su manjih dimenzija, dok se za probleme većih dimenzija koriste približne metode. S obzirom na tu činjenicu, prikazani pregled i opis nekih egzaktnih i nekih približnih metoda za rješavanje ovog problema koje mogu pomoći odgovornim osobama prilikom kreiranja planova i programa.

Literatura

[1] Rutherford, F.J. and Ahlgren, A. "Science for All Americans". Oxford University Press, USA. 1991.

[2] D. Matić and M. Božić. "Maximally Balanced Connected Partition Problem in Graphs: Application in Education", The Teaching of Mathematics, 15(2), p. 121–132, 2012.

[3] L. Lovász. "Coverings and colorings of hypergraphs", Proc. 4th Southeastern Conf. on Comb. Springer Berlin Heidelberg. p. 3–12, 1973.

[4] "Computer Science Curriculum 2008: An Interim Revision of CS 2001 Report from the Interim Review Task Force". 2008. http://www.acm.org/education/curricula/ComputerScience2008.pdf

[5] G. Andersson and L. Engebretsen. "Better approximation algorithms for Set Splitting and Not-All-Equal Sat", Information Processing Letters, 65(6), p. 305 – 311, 1998.

[6] B. Lazović, M. Marić,V. Filipović, A. Savić. "An integer linear programming formulation and genetic algorithm for the maximum set splitting problem", Publications de l'Institut Mathematique, 92(106) p. 25–34, 2012.

[7] J. Kratica. "An Electromagnetism-like method for the maximum set splitting problem", Yugoslav Journal of Operations Research, 23(1), p. 31–41, 2013.

[8] Matić, D. "A Variable Neighborhood Search Approach for Solving the Maximum Set Splitting Problem", Serdica Journal of Computing, 6(4) p. 369–384, 2012.

TREĆA MATEMATIČKA KONFERENCIJA REPUBLIKE SRPSKE
Trebinje, 7. i 8. juni 2013. godine

Spojene familije skupova i povezani topološki prostori

Nebojša Elez
Univerzitet u Istočnom Sarajevu
nelez@ffuis.edu.ba

Ognjen Papaz
Univerzitet u Istočnom Sarajevu
opapaz@ffuis.edu.ba

Ivana Čorlija
Univerzitet u Istočnom Sarajevu
ivana.corlija@gmail.com

Stručni rad

Apstrakt

U ovom radu proširen je pojam razdvojenosti dva skupa na familiju podskupova topološkog prostora X. Definisano je kad je familija $(A_i)_{i\in I}$ podskupova topološkog prostora X razdvojena, i kad je spojena. Zatim su dokazane odredjenje osobine ovih relacija, i dokazano je da je unija spojene familije povezanih prostora povezana.

1 Uvod

Definicija 1.1. Za familiju skupova $(A_i)_{i\in I}$ kažemo da je vezana ako za svaka dva indeksa $i, j \in I$ postoji konačan niz indeksa $i = i_0, \ldots, i_n = j$ takav da je $A_{i_{k-1}} \cap A_{i_k} \neq \emptyset$ za $k = 1, \ldots, n$.

Teorema 1.1. Familija skupova $(A_i)_{i\in I}$ je vezana ako i samo ako za svaki $J \subset I$, takav da je $\emptyset \neq J \neq I$, postoji $j \in J$ i $i \in I \setminus J$, da važi $A_i \cap A_j \neq \emptyset$.

Dokaz 1.1. Neka je $(A_i)_{i\in I}$ vezana familija i neka je $J \subset I$ takav da je $\emptyset \neq J \neq I$. Za $j \in J$ i $i \in I \setminus J$ postoji konačan niz indeksa $i = i_0, \ldots, i_n = j$ takav da je $A_{i_{k-1}} \cap A_{i_k} = \emptyset$ za $k = 1, \ldots, n$. Neka je $m = \max\{k | k < n,\ i_k \in J\}$. Tada je $i_m \in J$ i $i_{m+1} \in I \setminus J$. Dakle, važi uslov tvrdjenja.
Obrnto, neka familija $(A_i)_{i\in I}$ nije vezana. Neka je $i_0 \in I$ i neka je J skup svih indeksa $j \in I$ takvih da postoji konačan niz indeksa $i_n \ldots, i_n = j$ takav da je $A_{i_{k-1}} \cap A_{i_k} = \emptyset$ za $k = 1, \ldots, n$. Jasno je da je $\emptyset \neq J \neq I$ i da za sve $j \in J$ i $i \in I \setminus J$ važi $A_j \cap A_i = \emptyset$. Dakle, ne važi uslov tvrdjenja.

Definicija 1.2. Za dva skupa A i B u topološkom prostoru kažemo da su razdvojena ako važi $\overline{A} \cap B = \emptyset = A \cap \overline{B}$ i pišemo $A \leftrightarrow B$. Ako skupovi A i B nisu razdvojeni kažemo da su oni spojeni i pišemo $A \not\leftrightarrow B$.

Teorema 1.2. U svakom topološkom prostoru sledeći uslovi su ekvivalentni:

1. $A \leftrightarrow B$.
2. $\overline{A} \cap \overline{B} \cap (A \cup B) = \emptyset$.
3. A i B su disjunktni i otvoreni u $A \cup B$.
4. A i B su disjunktni i zatvoreni u $A \cup B$.

Dokaz 1.2. Uslovi 3. i 4. su ekvivalentni jer su disjunktni skupovi A i B komplementarni u $A \cup B$.
1.$\Leftrightarrow$ 2. jer je $\overline{A} \cap \overline{B} \cap (A \cup B) = (\overline{A} \cap B) \cup (A \cap \overline{B})$.
(1. $\Rightarrow$ 4.) $\mathrm{Cl}_{A \cup B}(A) = \overline{A} \cap (A \cup B) = A \cup (\overline{A} \cap B) = A \cup \emptyset = A$. Dakle, A je zatvoren u $A \cup B$. Analogno, B je zatvoren u $A \cup B$. Jasno je da je $A \cap B = \emptyset$.
(4. $\Rightarrow$ 2.) $\emptyset = A \cap B = \mathrm{Cl}_{A \cup B}(A) \cap \mathrm{Cl}_{A \cup B}(B) = \overline{A} \cap (A \cup B) \cap \overline{B}$.

Posledica 1.2.1. U svakom topološkom prostoru X važi:

1. Ako je $x \in \overline{A}$ onda je $\{x\} \not\leftrightarrow A$.
2. Ako je $\overline{A} \cap \overline{B} = \emptyset$ onda je $A \leftrightarrow B$.
3. Ako su disjunktni skupovi A i B otvoreni ili zatvoreni ili je jedan od njih otvoreno-zatvoren tada su oni razdvojeni.
4. $A \leftrightarrow X \setminus A$ ako i samo ako je A otvoreno zatvoren.

Teorema 1.3. U svakom topološkom prostoru X važe osobine:

1. $A \leftrightarrow \emptyset$ za sve $A \subset X$.
2. Ako je $A \leftrightarrow B$ tada je $B \leftrightarrow A$.
3. Ako je $A \leftrightarrow B$ tada je $A \cap B = \emptyset$.
4. Ako je $A' \subset A,\ B' \subset B$ i $A \leftrightarrow B$, tada je $A' \leftrightarrow B'$.
5. Ako je $A \leftrightarrow (B \cup C)$ tada je $A \leftrightarrow B$ i $A \leftrightarrow C$.

Dokaz 1.3. Dokaz ovog tvrdjenja izostavljamo.

2 Rezultati

Teorema 2.1. Ako u topološkom prostoru važi $A \leftrightarrow B$, tada je:

1. $A^+ \cup B^+ = (A \cup B)^+$.
2. $(A \cup B)^\circ = A^\circ \cup B^\circ$.
3. $\mathrm{Fr}A \cup \mathrm{Fr}B = \mathrm{Fr}(A \cup B)$.

Dokaz 2.1. 1. $(A \cup B)^+ = \overline{A \cup B} \setminus (A \cup B) = (\overline{A} \setminus (B \cup A)) \cup (\overline{B} \setminus (A \cup B)) = ((\overline{A} \setminus B) \setminus A) \cup ((\overline{B} \setminus A) \setminus B) = (\overline{A} \setminus A) \cup (\overline{B} \setminus B) = A^+ \cup B^+$.
2. Koristeći formulu $(A \cup B)^\circ \subset A^\circ \cup \overline{B}$, imamo

$$(A \cup B)^\circ \subset (A^\circ \cup \overline{B}) \cap (\overline{A} \cup B^\circ) = A^\circ \cup B^\circ \cup (\overline{A} \cap \overline{B}) = A^\circ \cup B^\circ.$$

3. Dovoljno je provjeriti da je $\mathrm{Fr}A \cup \mathrm{Fr}B \subset \mathrm{Fr}(A \cup B)$. Pokažimo da je $\mathrm{Fr}A \subset \mathrm{Fr}(A \cup B)$.

$$\mathrm{Fr}A = \overline{A} \setminus A^\circ = (\overline{A} \setminus B) \setminus A^\circ = (\overline{A} \setminus B^\circ) \setminus A^\circ = \overline{A} \setminus (A^\circ \cup B^\circ) =$$

$$\overline{A} \setminus (A \cup B)^\circ \subset \overline{A \cup B} \setminus (A \cup B) = \mathrm{Fr}(A \cup B).$$

U diskretnom prostoru dva skupa su spojena (razdvojena) ako i samo ako se sijeku (su disjunktni).

U indiskretnom prostoru dva skupa su spojena ako i samo ako su neprazna.

Disjunktni intervali $[0, 1)$ i $[1, 2)$ su spojeni u $\mathbb{R}$.

Neka je topologija $\mathcal{U}$ slabija od topologije $\mathcal{V}$ na skupu X. Ako je $A \leftrightarrow B$ u prostoru $(X, \mathcal{U})$ tada je $A \leftrightarrow B$ u prostoru $(X, \mathcal{V})$.

Ako su u prostoru X skupovi A i B razdvojeni i $A \cup B \subset Y \subset X$, tada su skupovi A i B razdvojeni u poprostoru Y.

Ako je $A \leftrightarrow B$ u prostoru X tada je $A \times Y \leftrightarrow B \times Y$.

Definicija 2.1. Za familiju skupova $(A_i)_{i \in I}$ u topološkom prostoru kažemo da je razdvojena ako za svaka dva različita indeksa $i, j \in I$ vrijedi $A_i \leftrightarrow A_j$.

Definicija 2.2. Za familiju skupova $(A_i)_{i \in I}$ u topološkom prostoru kažemo da je spojena ako za svako $J \subset I$, tako da je $J \neq \emptyset \neq I \setminus J$, važi $\bigcup_{j \in J} A_j \not\leftrightarrow \bigcup_{i \in I \setminus J} A_i$.

Familija skupova $(A_i)_{i \in I}$, u topološkom prostoru, je spojena ako postoji indekst $i_0 \in I$ takav da je $A_{i_0} \not\leftrightarrow A_i$ za svako $i \in I$.

Svaka vezana familija skupova u topološkom prostoru je spojena.

Familija skupova $(A_i)_{i \in I}$ u prostoru je vezana ako i samo ako je spojena u diskretnoj topologiji prostora.

Svaka spojena familija otvorenih skupova je vezana.

Teorema 2.2. Sledeći uslovi za topološki postor X su ekvivalentni:

1. X je povezan.
2. Svako pokriće prostora je spojeno.
3. Svako otvoreno pokriće prostora je vezano.

Dokaz 2.2. (1. $\Rightarrow$ 2.) Neka je povezan prosotor X unija familije nepraznih skupova $(A_i)_{i\in I}$. Neka je $J \subset I$ i $J \neq \emptyset \neq I \setminus J$. Tada neprazni skupovi $\bigcup_{j\in J} A_j$ i $\bigcup_{i\in I\setminus J} A_i$ nisu razdvojeni, što znači da je familija $(A_i)_{i\in I}$ spojena.
(2. $\Rightarrow$ 3.) Svako spojeno pokriče je vezano. (3. $\Rightarrow$ 1.) Svako dvoelementno otvoreno pokriće prostora X je vezna pa je X povezan.

Teorema 2.3. Unija spojene familije povezanih prostora je povezana.

Dokaz 2.3. Neka je $(C_i)_{i\in I}$ spojena familija povezanih skupova. Neka je $C = \bigcup_{i\in I} C_i = A \cup B$, gdje su A i B razdvojeni. Pokažimo da je C povezan tako što ćemo pokazati da je jedan od skupova A ili B prazan. Za svako $i \in I$ važi $C_i \subset A$ ili $C_i \subset B$. Neka je $J = \{i \in I | C_i \subset A\}$. Imamo da je $\bigcup_{i\in J} C_i \subset A$ i $\bigcup_{j\in I\setminus J} C_j \subset B$. Dakle, $\bigcup_{i\in J} C_i \leftrightarrow \bigcup_{j\in I\setminus J} C_j$, pa je $J = I$ ili $J = \emptyset$, tj. $C \subset A$ ili $C \subset B$. Što znači da je $A = \emptyset$ ili $B = \emptyset$.

Posledica 2.3.1. U svakom topološkom prostoru važi:

1. Unija familije povezanih skupova, koja ima neprazan presjek, je povezana.
2. Ako svake dvije tačke prostora pripadaju nekom povezanom skupu, tada je prostor povezan.
3. Ako je A povezan skup i ako je $A \subset B \subset \overline{A}$, tada je i B povezan.
4. Prostor je povezan ako ima gust povezan potprostor.

Dokaz 2.4. 1. Data familija je spojena.
2. Familija svih povezanih skupova prostora koji sadrže fiksiranu tašku zadovoljava uslov 1., a njena je unija čitav prostor.
3. Ako $x \in B$ tada $x \in \overline{A}$ i skup $A \cup \{x\}$ je spojena unija povezanih skupova, pa je povezan. Familija $(A \cup \{x\})_{x\in B}$ zadovoljava uslov 1. i njena unija je B.
4. Tvrdjenje slijedi iz 3.

Literatura

[1] Ryszard Engelking, General Topology, PWN Warszava, 1977.

TREĆA MATEMATIČKA KONFERENCIJA REPUBLIKE SRPSKE
Trebinje, 7. i 8. juni 2013. godine

Tabele kontigencije i problem maksimalne uslovne informacije

Tatjana Bajić
Visoka škola strukovnih studija za vaspitače, Šabac
ttanja.bajic@gmail.com

Stručni rad

Apstrakt

U primenama se često susrećemo sa trodimenzionalnim tabelama kontigencije gde je jedno obeležje kontrolna promenljiva na čijim nivoima se razmatraju povezanosti ishoda druga dva obeležja. U radu se razmatra pod kojim uslovima se na osnovu podataka iz trodimenzionalne tabele kontigencije mogu dobiti maksimalne uslovne informacije na svakom nivou kontrolnog obeležja, a odatle i maksimalna informacija koju dva obeležja nose o trećem. Dobijeni rezultati se mogu primeniti prilikom planiranja trodimenzionalne tabele kontigencije.

1 Uvod

Prilikom izvođenja statističkih zaključaka često se susrećemo sa potrebom da frekvencije ishoda dva ili više obeležja unakrsno predstavimo pomoću tabele kontigencije kako bi verifikovali povezanost između ishoda posmatranih obeležja. U tom smeru razvijeni su odgovarajući matematički modeli (loglinearni modeli) kojima se opisuje veza između obeležja na osnovu (logaritma) očekivanih ćelijskih frekvencija. Za realizovan uzorak se odgovarajućim statističkim testovima utvrđuje sa kojim pragom značajnosti se može smatrati da očekivane frekvencije zadovoljavaju pretpostavljeni loglinearni model (Agresti 1996).

Međutim, sa druge strane, imajući u vidu da se ishodi obeležja registruju slučajnim eksperimentima i da informacija u smislu Shannon-a kvantitativno meri koliko ishodi jednog obeležja iscrpljuju *neodređenost* ishoda drugog obeležja, tabele kontigencije se mogu analizirati i pomoću pojmova entropije i informacije u smislu Shannon-a (Bajić 2012). Šta više, pojam Shannon-ove informacije može biti od značaja prilikom planiranja tabele kontigencije u cilju dobijanja maksimalne informativnosti na osnovu podataka iz formirane tabele.

U ovom radu razmatramo trodimenzionalne tabele kontigencije gde jedno obeležje predstavlja kontrolnu promenljivu Z za čije fiksirane ishode (nivoe) ispitujemo međusobnu povezanost ishoda druga dva obeležja X i Y, pri čemu promenljiva X u određenom smislu "objašnjava" promenljivu Y. U cilju formiranja trodimenzionalne tabele kontigencije sa podacima koji pružaju maksimalnu informativnost, razmatramo pod kojim uslovima obeležje X sadrži optimalnu (maksimalnu) informaciju o obeležju Y na svakom nivou kontrolne promenljive Z.

2 Struktura trodimenzionalne tabele kontigencije

Neka su X, Y i Z slučajne promenljive definisane nad istim prostorom verovatnoća $(\Omega, \mathcal{F}, P)$ koje redom indukuju *konačna* razbijanja A_i, $i = 1,2,...,I$, B_j, $j = 1,2,...,J$ i C_k, $k = 1,2,...,K$ skupa svih mogućih elementarnih ishoda. Ako sa α, β i γ označimo slučajne eksperimente u kojima se redom registruju ishodi promenljivih X, Y i Z, tada za događaje $\{A_i\}$, $\{B_j\}$ i $\{C_k\}$ kažemo da su *skupovi mogućih ishoda* eksperimenata α, β i γ, respektivno.

Pretpostavimo da su eksperimenti α, β i γ realizovani nad uzorkom obima N. Neka su $\{N_{ijk}\}$ registrovane frekvencije događaja $\{A_iB_jC_k\}$, a $\{N_{ij\bullet}\}$, $\{N_{i\bullet k}\}$, $\{N_{\bullet jk}\}$, $\{N_{i\bullet\bullet}\}$, $\{N_{\bullet j\bullet}\}$ i $\{N_{\bullet\bullet k}\}$ odgovarajuće marginalne frekvencije redom događaja $\{A_iB_j\}$, $\{A_iC_k\}$, $\{B_jC_k\}$, $\{A_i\}$, $\{B_j\}$ i $\{C_k\}$. Ukoliko su zadovoljene jednakosti

$$N_{ij\bullet} = \sum_{k=1}^{K} N_{ijk}, \quad N_{i\bullet k} = \sum_{j=1}^{J} N_{ijk}, \quad N_{\bullet jk} = \sum_{i=1}^{I} N_{ijk} \tag{1}$$

$$N_{i\bullet\bullet} = \sum_{j=1}^{J}\sum_{k=1}^{K} N_{ijk}, \quad N_{\bullet j\bullet} = \sum_{k=1}^{K}\sum_{i=1}^{I} N_{ijk}, \quad N_{\bullet\bullet k} = \sum_{i=1}^{I}\sum_{j=1}^{J} N_{ijk} \tag{2}$$

i

$$N = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K} N_{ijk} = \sum_{i=1}^{I} N_{i\bullet\bullet} = \sum_{j=1}^{J} N_{\bullet j\bullet} = \sum_{j=1}^{J} N_{\bullet\bullet k}, \tag{3}$$

tada frekvencije kombinacija ishoda posmatranih eksperimenata možemo predstaviti pomoću trodimenzionalne tabele kontigencije.

U osnovi trodimenzionalne tabele kontigencije su dvodimenzionalne tabele koje nazivamo *parcijalnim tabelama kontigencije* i koje se dobijaju fiksiranjem jednog od ishoda C_k, B_j ili A_i. Ako sa (α_k, β_k), (α_j, γ_j) i (β_i, γ_i) označimo redom parcijalne tabele kontigencije za fiksirane ishode C_k, B_j i A_i, tada frekvencije ishoda u parcijalnim tabelama možemo predstaviti na sledeći način:

$$\underset{(\alpha_k,\beta_k)}{\begin{array}{c|c|c} X\setminus Y & B_jC_k & \sum \\ \hline A_iC_k & N_{ijk} & N_{i\bullet k} \\ \hline \sum & N_{\bullet jk} & N_{\bullet\bullet k} \end{array}}, \quad \underset{(\alpha_j,\gamma_j)}{\begin{array}{c|c|c} X\setminus Z & C_kB_j & \sum \\ \hline A_iB_j & N_{ijk} & N_{ij\bullet} \\ \hline \sum & N_{\bullet jk} & N_{\bullet j\bullet} \end{array}}, \quad \underset{(\beta_i,\gamma_i)}{\begin{array}{c|c|c} Y\setminus Z & C_kA_i & \sum \\ \hline B_jA_i & N_{ijk} & N_{ij\bullet} \\ \hline \sum & N_{i\bullet k} & N_{i\bullet\bullet} \end{array}}. \tag{4}$$

Združivanjem parcijalnih tabela (α_k, β_k) po ishodima C_k, $k = 1,2,\ldots,K$, odnosno tabela (α_j, γ_j) po ishodima B_j, $j = 1,2,...,J$, odnosno tabela (β_i, γ_i) po ishodima A_i, $i = 1,2,...,I$, dobijamo takođe dvodimenzionalne tabele kontigencije ali one predstavljaju *marginalne tabele kontigencije* koje redom odgovaraju eksperimentima α i β, u oznaci (α, β), odnosno α i γ, u oznaci (α, γ), odnosno β i γ, u oznaci (β, γ), tj.

$$\underset{(\alpha,\beta)}{\begin{array}{c|c|c} X \setminus Y & B_j & \sum \\ \hline A_i & N_{ij\bullet} & N_{i\bullet\bullet} \\ \hline \sum & N_{\bullet j\bullet} & N \end{array}}, \quad \underset{(\alpha,\gamma)}{\begin{array}{c|c|c} X \setminus Z & C_k & \sum \\ \hline A_i & N_{i\bullet k} & N_{i\bullet\bullet} \\ \hline \sum & N_{\bullet\bullet k} & N \end{array}}, \quad \underset{(\beta,\gamma)}{\begin{array}{c|c|c} Y \setminus Z & C_k & \sum \\ \hline B_j & N_{\bullet jk} & N_{\bullet j\bullet} \\ \hline \sum & N_{\bullet\bullet k} & N \end{array}}. \qquad (5)$$

U marginalnoj tabeli kontigencije, frekvencija u određenoj ćeliji je jednaka sumi frekvencija iz ćelija koje su na istoj lokaciji u parcijalnim tabelama. Marginalne tabele ne daju nikakvu informaciju o trećem eksperimentu po čijim ishodima je izvršeno združivanje parcijalnih tabela u marginalnu, odnosno to su prosto dvodimenzionalne tabele kontigencije dva posmatrana eksperimenta.

Statistička zavisnost između X i Y predstavljena parcijalnom tabelom kontigencije (α_k, β_k) naziva se *uslovna statistička zavisnost*, jer se analizira pod uslovom realizacije ishoda C_k promenljive Z. Uslovne statističke zavisnosti u parcijalnim tabelama u opštem slučaju ne povlače statističku zavisnost u marginalnoj tabeli (Agresti 1996: 53-57).

Budući da su X, Y i Z slučajne promenljive određene odgovarajućim raspodelama verovatnoća, umesto ćelijskih i marginalnih frekvencija prvenstveno se razmatraju verovatnoće realizacija događaja $\{A_i\}$, $\{B_j\}$ i $\{C_k\}$, koje redom označavamo sa $\{P(A_i)\}$, $\{P(B_j)\}$ i $\{P(C_k)\}$, zatim uslovne verovatnoće događaja $\{A_i\}$, $\{B_j\}$ i $\{A_iB_j\}$ pod uslovom realizacije događaja C_k, $k = 1, 2, ..., K$, koje redom označavamo sa $\{P_{C_k}(A_i)\}$, $\{P_{C_k}(B_j)\}$ i $\{P_{C_k}(A_iB_j)\}$, i uslovne verovatnoće realizacija događaja $\{A_i\}$ i $\{B_j\}$ pod uslovom realizacije događaja B_jC_k odnosno događaja A_iC_k, $i = 1, 2, ..., I$, $j = 1, 2, ..., J$, $k = 1, 2, ..., K$, označene sa $\{P_{B_jC_k}(A_i)\}$ i $\{P_{A_iC_k}(B_j)\}$ respektivno.

3 Uslovna informacija i problem optimalne uslovne informacije

Neka su X, Y i Z obeležja čije su frekvencije ishoda predstavljene tabelama kontigencije (1-5) tako da je Z kontrolna promenljiva na čijim nivoima se razmatra povezanost između X i Y, pri čemu promenljiva X u određenom smislu "objašnjava" promenljivu Y. Budući da se ishodi obeležja Z, X i Y predviđaju redom eksperimentima γ, α i β, interesuje nas pod kojim uslovima eksperiment α sadrži optimalnu informaciju o eksperimentu β za dati eksperiment γ. Odatle, prvo razmatramo pojmove uslovne entropije i uslovne informacije.

3.1 Uslovna entropija i uslovna informacija

Definicija uslovne entropije. Neka su α, β i γ eksperimenti sa konačnim skupovima mogućih ishoda $\{A_i\}$, $\{B_j\}$ i $\{C_k\}$, respektivno. *Uslovna entropija eksperimenta* α, pod pretpostavkom realizacije ishoda C_k eksperimenta γ, u oznaci $H_{C_k}(\alpha)$, se definiše sa

$$H_{C_k}(\alpha) = -\sum_{i=1}^{I} P_{C_k}(A_i) \log P_{C_k}(A_i), \;\; P(C_k) > 0. \tag{6}$$

Za $P_{C_k}(A_i) = 0$ uzima se da je $\log P_{C_k}(A_i) = 0$. Analogno se definišu i uslovne entropije $H_{C_k}(\beta)$ i $H_{C_k}(\alpha\beta)$. Odatle, pod pretpostavkom $\{P(C_k) > 0\}$, uslovna entropija eksperimenta α za realizovan eksperiment γ je data sa

$$H_\gamma(\alpha) = \sum_{k=1}^{K} P(C_k) H_{C_k}(\alpha) = -\sum_{k=1}^{K}\sum_{i=1}^{I} P(C_k) P_{C_k}(A_i) \log P_{C_k}(A_i) \tag{7}$$

odnosno, uslovna entropija eksperimenta β za realizovan eksperiment γ sa

$$H_\gamma(\beta) = \sum_{k=1}^{K} P(C_k) H_{C_k}(\beta) = -\sum_{k=1}^{K}\sum_{j=1}^{J} P(C_k) P_{C_k}(B_j) \log P_{C_k}(B_j), \tag{8}$$

dok je uslovna entropija složenog eksperimenta $\alpha\beta$ za realizovan eksperiment γ određena sa

$$H_\gamma(\alpha\beta) = \sum_{k=1}^{K} P(C_k) H_{C_k}(\alpha\beta) = -\sum_{k=1}^{K}\sum_{j=1}^{J}\sum_{i=1}^{I} P(C_k) P_{C_k}(A_i B_j) \log P_{C_k}(A_i B_j), \tag{9}$$

(Yaglom 1973: 59-73).

Teorema 3.1. *Neka su α, β i γ eksperimenti čiji ishodi su predstavljeni tabelama kontigencije (1-5) tako da $\{P(C_k) > 0\}$ i $\{P(A_i C_k) > 0\}$. Tada, uslovna entropija eksperimenta β za realizovane eksperimente γ i α je određena sa*

$$H_{\alpha\gamma}(\beta) = \sum_{k=1}^{K}\sum_{i=1}^{I} P(A_i C_k) H_{A_i C_k}(\beta) \tag{10}$$

gde je

$$H_{A_i C_k}(\beta) = -\sum_{j=1}^{J} P_{A_i C_k}(B_j) \log P_{A_i C_k}(B_j) \tag{11}$$

uslovna entropija eksperimenta β za realizovan događaj $A_i C_k$. Analogno se određuje i uslovna entropija eksperimenta α za realizovane eksperimente γ i β, $H_{\beta\gamma}(\alpha)$, uz dodatni uslov da je $\{P(B_j C_k) > 0\}$.

Dokaz*:* Budući da su α, β i γ slučajni eksperimenti sa skupom mogućih ishoda $\{A_i\}$, $\{B_j\}$ i $\{C_k\}$, respektivno, zamenom eksperimenta γ sa složenim eksperimentom $\alpha\gamma$ u (8) i dodatnim sumiranjem po indeksu $i = 1, 2, ..., I$, neposredno se dobija (10) odnosno (11). Analogno važi i za $H_{\beta\gamma}(\alpha)$■

Definicija uslovne informacije (u Shannon-ovom smislu). Neka su α i β dva proizvoljna slučajna eksperimenta definisana nad istim prostorom verovatnoća.

Tada, količina informacije o eksperimentu β koja je sadržana u eksperimentu α definiše se sa

$$I(\alpha,\beta) = H(\beta) - H_\alpha(\beta), \tag{12}$$

(Yaglom 1973: 74), pri čemu važi

$$I(\alpha,\beta) = H(\beta) - H_\alpha(\beta) = H(\alpha) - H_\beta(\alpha), \tag{13}$$

odnosno $I(\alpha,\beta)$ je uzajamna informacija jednog eksperimenta o drugom (Yaglom 1973: 85). Ukoliko su eksperimenti statistički nezavisni (i samo u tom slučaju) tada je $I(\alpha,\beta)=0$, odnosno $H(\beta)=H_\alpha(\beta)$ i $H(\alpha)=H_\beta(\alpha)$ (Yaglom 1973: 88). Ako je γ treći slučajni eksperiment definisan nad istim prostorom verovatnoća kao α i β, tada *uslovnu informaciju* koju eksperiment α sadrži o eksperimentu β pod uslovom realizacije eksperimenta γ, definišemo sa

$$I_\gamma(\alpha,\beta) = H_\gamma(\beta) - H_{\alpha\gamma}(\beta) \tag{14}$$

(Yaglom 1973: 91). Imajući u vidu jednakost

$$H(\alpha\gamma\beta) - H(\gamma) = H_\gamma(\alpha) + H_{\alpha\gamma}(\beta) = H_\gamma(\beta) + H_{\beta\gamma}(\alpha)$$

dobijamo

$$I_\gamma(\alpha,\beta) = H_\gamma(\beta) - H_{\alpha\gamma}(\beta) = H_\gamma(\alpha) - H_{\beta\gamma}(\alpha). \tag{15}$$

(Yaglom 1973: 92).

Teorema 3.2. *Neka su α, β i γ eksperimenti čiji ishodi su predstavljeni tabelama kontigencije (1-5) tako da $\{P(C_k)>0\}$. Tada, uslovna informacija eksperimenata α i β pod uslovom realizacije eksperimenta γ je određena sa*

$$I_\gamma(\alpha,\beta) = \sum_{k=1}^{K} P(C_k) I_{C_k}(\alpha,\beta) \tag{16}$$

gde je $I_{C_k}(\alpha,\beta)$ uslovna informacija eksperimenata α i β za realizovan događaj C_k eksperimenta γ data sa

$$I_{C_k}(\alpha,\beta) = \sum_{i=1}^{I}\sum_{j=1}^{J} P_{C_k}(A_iB_j)\log\frac{P_{C_k}(A_iB_j)}{P_{C_k}(A_i)P_{C_k}(B_j)}, k=1,2,...,K. \tag{17}$$

Dokaz*:* Imajući u vidu da su uslovne verovatnoće $\{P_{C_k}(A_i)\}$, $\{P_{C_k}(B_j)\}$ i $\{P_{C_k}(A_iB_j)\}$ realizacije redom ishoda $\{A_i\}$, $\{B_j\}$ i $\{A_iB_j\}$ pod uslovom realizacije događaja C_k, $k=1,2,...,K$, zamenom (8), (10) i (11) u (14) dobija se (16) i (17)$\blacksquare$

Sa uslovnom informacijom $I_\gamma(\alpha,\beta)$ u direktnoj vezi je informacija koju složen eksperiment $\alpha\gamma$ nosi o eksperimentu β i koja je definisana sa

$$I(\alpha\gamma,\beta) = H(\beta) - H_{\alpha\gamma}(\beta), \tag{18}$$

jer

$$\begin{aligned} I(\alpha\gamma,\beta) &= H(\beta)-H_{\alpha\gamma}(\beta)=H(\beta)-H_{\gamma}(\beta)+H_{\gamma}(\beta)-H_{\alpha\gamma}(\beta) \\ &= I(\gamma,\beta)+I_{\gamma}(\alpha,\beta) \end{aligned} \tag{19}$$

i odatle važi nejednakost $I_{\gamma}(\alpha,\beta)\leq I(\alpha\gamma,\beta)$ pri čemu se jednakost dostiže kada je $I(\gamma,\beta)=0$, odnosno kada su eksperimenti β i γ statistički nezavisni.

3.2 Problem optimalne uslovne informacije

U cilju planiranja trodimenzionalne tabele kontigencije sa podacima koji bi pružali maksimalnu informativnost, razmatramo pod kojim uslovima se dobija optimalna uslovna informacija koju eksperiment α nosi o eksperimentu β pod pretpostavkom realizacije eksperimenta γ. Imajući u vidu da je uslovna informacija $I_{\gamma}(\alpha,\beta)$ određena sa (14), odnosno sa (16) i (17), nadalje pretpostavljamo da su verovatnoće događaja $\{C_k\}$ eksperimenta γ unapred date i da važi

$$\{P(C_k)>0\}\ \text{ i }\ \{P(A_iC_k)>0\}\,. \tag{20}$$

Označimo sa α_k i β_k redom eksperimente α i β za svaki fiksirani ishod C_k eksperimenta γ, $k=1,...,K$. Odatle, uslovne entropije $H_{\alpha C_k}(\beta)$ i $H_{C_k}(\beta)$ možemo redom označiti sa $H_{\alpha_k}(\beta_k)$ i $H(\beta_k)$. Budući da je uslovna informacija eksperimenata α i β za realizovan eksperiment γ određena sa (16), za date verovatnoće $\{P(C_k)>0\}$, optimalna $I_{\gamma}(\alpha,\beta)$ se dobija ukoliko su za svako fiksirano C_k, uslovne informacije

$$I_{C_k}(\alpha,\beta)=I(\alpha_k,\beta_k)=H(\beta_k)-H_{\alpha_k}(\beta_k),\ k=1,...,K, \tag{21}$$

optimalne. S obzirom da za uslovnu informaciju (14) važi nejednakost $0\leq I_{\gamma}(\alpha,\beta)\leq H_{\gamma}(\beta)$ (Yaglom 1973: 91), prvo razmatramo pod kojim uslovima se dobijaju jednakosti $I_{\gamma}(\alpha,\beta)=0$ i $I_{\gamma}(\alpha,\beta)=H_{\gamma}(\beta)$.

Lema 3.1. *Neka su α, β i γ eksperimenti čije se frekvencije ishoda mogu predstaviti tabelama kontigencije (1-5) i neka su date verovatnoće realizacija ishoda $\{C_k\}$ eksperimenta γ takve da važi (20). Ako su na svakom nivou eksperimenta γ, eksperimenti α i β uslovno statistički nezavisni tada je $I_{\gamma}(\alpha,\beta)=0$ i eksperiment α ne nosi nikakvu dodatnu informaciju o β u odnosu na eksperiment γ, odnosno $I(\alpha\gamma,\beta)=I(\gamma,\beta)$.*

Dokaz: Ako su na svakom nivou eksperimenta γ eksperimenti α i β uslovno statistički nezavisni to znači da je za svako fiksirano C_k, $P_{C_k}(A_iB_j)=P_{C_k}(A_i)P_{C_k}(B_j)$, $i=1,...,I$, $j=1,...,J$, te iz (17) sledi da je $I(\alpha_k,\beta_k)=0$, a odatle iz (16) dobijamo $I_{\gamma}(\alpha,\beta)=0$.

Sa druge strane, iz definicije uslovne informacije (14), ako je $I_{\gamma}(\alpha,\beta)=0$ tada je $H_{\gamma}(\beta)=H_{\alpha\gamma}(\beta)$, a odatle iz (18) sledi

$$I(\alpha\gamma,\beta)=H(\beta)-H_{\alpha\gamma}(\beta)=H(\beta)-H_{\gamma}(\beta)=I(\gamma,\beta),$$

odnosno eksperiment α ne nosi nikakvu dodatnu informaciju o β u odnosu na eksperiment γ■

Definicija potpune informacije. Eksperiment α sadrži *potpunu informaciju* o eksperimentu β ako α iscrpljuje svu neodređenost eksperimenta β, odnosno ako je $H_\alpha(\beta) = 0$. Budući da je u tom slučaju $I(\alpha, \beta) = H(\beta)$, potpuna informacija je inače maksimalna moguća informacija koju jedan eksperiment može nositi o drugom (Yaglom 1973: 91). Analogno, složen eksperiment $\alpha\gamma$ sadrži *potpunu informaciju* o eksperimentu β ako $\alpha\gamma$ iscrpljuje svu neodređenost eksperimenta β, odnosno ako je $H_{\alpha\gamma}(\beta) = 0$. U tom slučaju je $I(\alpha\gamma, \beta) = H(\beta)$ i to je takođe maksimalna moguća informacija koju složen eksperiment može nositi o trećem eksperimentu.

Vrednost entropije $H_{\alpha\gamma}(\beta) = 0$ zapravo znači da na svakom nivou C_k, $k = 1, 2, ..., K$, za sve $i = 1, 2, ..., I$, važi $H_{A_iC_k}(\beta) = 0$, odnosno na svakom nivou eksperimenta γ za svako $i = 1, 2, ..., I$ sledi da je za neko $j = 1, ..., J$, $P_{A_iC_k}(B_j) = 1$, dok za sve ostale $l \neq j$, $P_{A_iC_k}(B_l) = 0$.

Definicija potpune uslovne informacije. Ako na nivou C_k eksperimenta γ, $k = 1, 2, ..., K$, eksperiment α_k sadrži *potpunu informaciju* o eksperimentu β_k, odnosno ako je $H_{\alpha_k}(\beta_k) = 0$, tada kažemo da eksperiment α sadrži *potpunu uslovnu informaciju* o eksperimentu β na nivou C_k eksperimenta γ. Analogno, ako za realizovan eksperiment γ, eksperiment $\alpha\gamma$ sadrži *potpunu informaciju* o eksperimentu β odnosno ako je $H_{\alpha\gamma}(\beta) = 0$, tada kažemo da eksperiment α sadrži *potpunu uslovnu informaciju* o eksperimentu β pod uslovom realizacije eksperimenta γ. U tom slučaju važi $I_\gamma(\alpha, \beta) = H_\gamma(\beta)$ i to je za realizovan eksperiment γ, maksimalna moguća uslovna informacija koju eksperiment α može nositi o eksperimentu β.

Lema 3.2. *Neka su α, β i γ eksperimenti čije se frekvencije ishoda mogu predstaviti tabelama kontigencije (1-5) i neka su date verovatnoće realizacija ishoda $\{C_k\}$ eksperimenta γ takve da važi (20). Eksperiment α sadrži potpunu uslovnu informaciju o eksperimentu β na svakom nivou eksperimenta γ ako i samo ako eksperiment α sadrži potpunu uslovnu informaciju o eksperimentu β pod uslovom realizacije eksperimenta γ. Šta više, ako eksperiment $\alpha\gamma$ sadrži potpunu informaciju o eksperimentu β i ako su eksperimenti β i γ statistički nezavisni tada je*

$$I_\gamma(\alpha, \beta) = I(\alpha\gamma, \beta) = H(\beta). \tag{22}$$

Dokaz: Ako je $H_{\alpha_k}(\beta_k) = 0$ na svakom nivou C_k eksperimenta γ, $k = 1, 2, ..., K$, iz (21) neposredno sledi da je $I(\alpha_k, \beta_k) = H(\beta_k)$ za svako $k = 1, 2, ..., K$, odnosno zbog uslova $\{P(C_k) > 0\}$ dobijamo da je

$$I_\gamma(\alpha, \beta) = \sum_{k=1}^{K} P(C_k) I(\alpha_k, \beta_k) = \sum_{k=1}^{K} P(C_k) H(\beta_k) = H_\gamma(\beta) \tag{23}$$

što znači da eksperiment α sadrži potpunu uslovnu informaciju o eksperimentu β pod uslovom realizacije eksperimenta γ i obratno. Šta više, iz $H_{\alpha\gamma}(\beta) = 0$ na osnovu definicije sledi da je $I(\alpha\gamma, \beta) = H(\beta)$ i $I_\gamma(\alpha, \beta) = H_\gamma(\beta)$. Odatle, ako su uz to eksperimenti β i γ statistički nezavisni tada je $I(\gamma, \beta) = 0$ odnosno $H(\beta) = H_\gamma(\beta)$ te neposredno dobijamo (22)■

Kako su $I_\gamma(\alpha, \beta) = H_\gamma(\beta)$ i $I_\gamma(\alpha, \beta) = 0$ optimalne vrednosti uslovne informacije $I_\gamma(\alpha, \beta)$ koje se dobijaju kada na svakom nivou C_k eksperimenta γ, $H_{\alpha_k}(\beta_k) = 0$, odnosno $H_{\alpha_k}(\beta_k) = H(\beta_k)$, respektivno, zanima nas pod kojim uslovima će na svakom nivou C_k, eksperimenta γ, eksperiment α dati maksimalnu uslovnu informaciju o eksperimentu β ako je zadovoljena stroga nejednakost $0 < H_{\alpha_k}(\beta_k) < H(\beta_k)$. Matematički model kojim se opisuje postavljen problem može se formulisati sa

$$\underset{I(\alpha_k,\beta_k)}{(\max)}\ I_\gamma(\alpha, \beta) = \sum_{k=1}^{K} P(C_k) I(\alpha_k, \beta_k) \tag{24}$$

pod uslovom da je za svaki ishod C_k, $k = 1, ..., K$, eksperimenta γ data njegova verovatnoća realizacije $P(C_k) > 0$ i eksperimenti α_k i β_k zadovoljavaju strogu nejednakost $0 < H_{\alpha_k}(\beta_k) < H(\beta_k)$, odnosno na svakom nivou C_k, eksperiment α_k ne iscrpljuje u potpunosti neodređenost eksperimenta β_k i eksperimenti α_k i β_k nisu statistički nezavisni.

Imajući u vidu da za svako fiksirano C_k, α i β se mogu razmatrati kao eksperimenti α_k i β_k predstavljeni dvodimenzionalnom parcijalnom tabelom kontigencije (α_k, β_k), u dokazu leme kojom se daje rešenje problema (24) koristi se sledeća teorema koju navodimo bez dokaza (Bajić 2012: 257).

Teorema 3.3. *Neka su α i β statistički zavisni eksperimenti koji su predstavljeni dvodimenzionalnom tabelom kontigencije sa entropijom $H(\alpha\beta)$. Ako eksperiment α ne iscrpljuje u potpunosti neodređenost eksperimenta β, tada se maksimalna informacija koju eksperiment α nosi o eksperimentu β dostiže kada su svi ishodi eksperimenta α jednakoverovatni.*

Na osnovu principa maksimalne entropije slučajnog eksperimenta (Yaglom 1973: 48), eksperiment koji ima jednakoverovatne ishode ima maksimalnu entropiju. Odatle, prethodna teorema pod pretpostavljenim uslovima ukazuje da maksimalna entropija eksperimenta α povlači dostizanje maksimalne informacije koju eksperiment α nosi o eksperimentu β.

Lema 3.3. *Neka su α, β i γ eksperimenti čije se frekvencije ishoda mogu predstaviti tabelama kontigencije (1-5) i neka su date verovatnoće realizacija ishoda $\{C_k\}$ eksperimenta γ takve da važi (20). Pretpostavimo da za svako fiksirano C_k eksperimenta γ, $k = 1, 2, \ldots, K$, eksperimenti α i β su uslovno statistički zavisni sa datom entropijom $H(\alpha_k\beta_k)$ i eksperiment α ne sadrži potpunu uslovnu informaciju o eksperimentu β. Tada, eksperiment α će sadržati maksimalnu uslovnu informaciju o eksperimentu β pod uslovom realizacije eksperimenta γ ako i samo ako su eksperimenti α i γ statistički nezavisni i ishodi eksperimenta α su jednakoverovatni.*

Dokaz: Budući da su verovatnoće $\{P(C_k) > 0\}$ realizacija ishoda $\{C_k\}$ eksperimenta γ date, iz (24) sledi da se maksimalna uslovna informacija $I_\gamma(\alpha, \beta)$ dobija kada su za svako fiksirano C_k uslovne informacije (17) maksimalne.

Pretpostavimo da pod uslovom realizacije eksperimenta γ, eksperiment α sadrži maksimalnu uslovnu informaciju o eksperimentu β, odnosno za svako fiksirano C_k, eksperiment α_k sadrži maksimalnu informaciju o eksperimentu β_k. Imajući u vidu da na svakom nivou C_k eksperimenta γ, eksperimenti α_k i β_k zadovoljavaju nejednakost $0 < H_{\alpha_k}(\beta_k) < H(\beta_k)$ sa datom $H(\alpha_k\beta_k)$, na osnovu Teoreme 3.3 sledi da su svi ishodi od α_k jednakoverovatni, odnosno da je

$$P_{C_k}(A_i) = \frac{1}{I}, i = 1, 2, ..., I. \tag{25}$$

Zamenom (25) u (6) dobijamo

$$H_{C_k}(\alpha) = -\sum_{i=1}^{I} P_{C_k}(A_i) \log P_{C_k}(A_i) = -\sum_{i=1}^{I} \frac{1}{I} \log \frac{1}{I} = \log I. \tag{26}$$

Kako (26) važi za svako C_k, $k = 1, 2, ..., K$, iz (7) i (26) neposredno sledi da je

$$H_\gamma(\alpha) = \sum_{k=1}^{K} P(C_k) H_{C_k}(\alpha) = \log I \sum_{k=1}^{K} P(C_k) = \log I. \tag{27}$$

Razmatrajući α i γ kao eksperimente predstavljene marginalnom tabelom kontigencije (α, γ), na osnovu Snannon-ove nejednakosti $0 \leq H_\gamma(\alpha) \leq H(\alpha)$ i principa maksimalne entropije, iz (27) sledi da je $H(\alpha) = H_\gamma(\alpha) = \log I$, a odatle iz $I(\gamma, \alpha) = H(\alpha) - H_\gamma(\alpha)$ dobijamo da je $I(\gamma, \alpha) = 0$. Budući da $I(\gamma, \alpha) = 0$ važi samo u slučaju statističke nezavisnosti dva posmatrana eksperimenta (Yaglom 1973: 88), neposredno sledi da su eksperimenti α i γ statistički nezavisni.

Obratno, neka su α i γ statistički nezavisni eksperimenti i neka su ishodi eksperimenta α jednakoverovatni, odnosno

$$P(A_i) = \frac{1}{I}, i = 1, 2, ..., I.$$

Iz statističke nezavisnosti ekperimenata α i γ dobijamo da je za uslovne verovatnoće $\{P_{C_k}(A_i)\}$ ispunjeno $P_{C_k}(A_i) = P(A_i)$, $i = 1, 2, ..., I$, $k = 1, ..., K$, što povlači da je za svaki fiksirani ishod C_k eksperimenta γ, $H(\alpha_k) = \log I$, odnosno svi ishodi eksperimenta α_k su jednakoverovatni. Imajući u vidu pretpostavku da je za svako C_k eksperimenta γ zadovoljen uslov $0 < H_{\alpha_k}(\beta_k) < H(\beta_k)$ sa datom $H(\alpha_k\beta_k)$, na osnovu Teoreme 3.3, iz jednakosti $H(\alpha_k) = \log I$ sledi da eksperiment α_k sadrži maksimalnu informaciju o eksperimentu β_k. Kako to važi za svako C_k, $k = 1, 2, ..., K$, to eksperiment α sadrži maksimalnu uslovnu informaciju o eksperimentu β pod uslovom realizacije eksperimenta γ■

Kako je $I_\gamma(\alpha, \beta) \leq H_\gamma(\beta)$, a jednakost $I_\gamma(\alpha, \beta) = H_\gamma(\beta)$ ukazuje na dostizanje potpune uslovne informacije, u postavci problema (24), nejednakost $0 <$

$H_{\alpha_k}(\beta_k) < H(\beta_k)$ se može zameniti sa slabijim uslovom $0 \leq H_{\alpha_k}(\beta_k) < H(\beta_k)$, $k = 1, 2, \ldots, K$. Sa druge strane, vrednost uslovne informacije $I_\gamma(\alpha, \beta)$ zavisi i od uslovne statističke zavisnosti eksperimenata α i β na svakom nivou eksperimenta γ koju ne možemo unapred da predvideti jer su α i β slučajni eksperimenti. Međutim, budući da nas za dato γ zanimaju samo uslovi pod kojima bi $I_\gamma(\alpha, \beta)$ dostizala svoju maksimalnu vrednost za realizovane α i β, to u opštem slučaju razmatramo problem (24) pod uslovom da važi $0 \leq H_{\alpha_k}(\beta_k) \leq H(\beta_k)$, $k = 1, 2, \ldots, K$.

Teorema 3.4. *Neka su α, β i γ eksperimenti čije se frekvencije ishoda mogu predstaviti tabelama kontigencije (1-5) i neka su date verovatnoće realizacija ishoda $\{C_k\}$ eksperimenta γ takve da važi (20) i da je za svako fiksirano C_k eksperimenta γ, $k = 1, 2, \ldots, K$, data entropija $H(\alpha_k\beta_k)$ eksperimenata α i β. Tada, eksperiment α sadrži maksimalnu uslovnu informaciju o eksperimentu β, pod uslovom realizacije eksperimenta γ, ako su eksperimenti α i γ su statistički nezavisni i ishodi eksperimenta α su jednakoverovatni.*

Dokaz: Neposredno sledi iz drugog dela dokaza Leme 3.3 sa tom razlikom što ukoliko je na nekom nivou C_k eksperimenta γ, $H_{\alpha_k}(\beta_k) = 0$ tada je na tom nivou $I(\alpha_k, \beta_k) = H(\beta_k)$, odnosno eksperiment α_k nosi potpunu informaciju o eksperimentu β_k, a ako je na nekom nivou C_k eksperimenta γ, $H_{\alpha_k}(\beta_k) = H(\beta_k)$ tada je na tom nivou $I(\alpha_k, \beta_k) = 0$, odnosno eksperiment α_k ne nosi nikakvu dodatnu informaciju o eksperimentu β_k u odnosu na ishod C_k eksperimenta γ. Ukoliko $I(\alpha_k, \beta_k) = H(\beta_k)$ važi na svakom nivou C_k, $k = 1, 2, \ldots, K$ eksperimenta γ, tada na osnovu Leme 3.2 eksperiment α nosi potpunu uslovnu informaciju o eksperimentu β, odnosno $I_\gamma(\alpha, \beta) = H_\gamma(\beta)$. Sa druge strane, ako $H_{\alpha_k}(\beta_k) = H(\beta_k)$ važi na svakom nivou C_k, $k = 1, 2, \ldots, K$ eksperimenta γ, tada na osnovu Leme 3.1 eksperiment α ne nosi nikakvu dodatnu informaciju o β u odnosu na eksperiment γ, odnosno $I_\gamma(\alpha, \beta) = 0$ i u tom slučaju to je maksimalna uslovna informacija koju eksperiment α nosi o eksperimentu β, pod uslovom realizacije eksperimenta γ jer su α i β uslovno statistički nezavisni na svakom nivou eksperimenta γ■

U slučaju kada je $I_\gamma(\alpha, \beta) = 0$, tada je $I(\alpha\gamma, \beta) = I(\gamma, \beta)$ i može se razmatrati problem optimalne informacije u dvodimenzionalnoj marginalnoj tabeli kontigencije (γ, β). Šta više, uzimajući u obzir (19), odnosno da za informaciju koju složen eksperiment $\alpha\gamma$ nosi o eksperimentu β važi $I(\alpha\gamma, \beta) = I(\gamma, \beta) + I_\gamma(\alpha, \beta)$, maksimalna uslovna informacija $I_\gamma(\alpha, \beta)$ pod određenim uslovima povlači dobijanje maksimalne informacije koju složen eksperiment $\alpha\gamma$ nosi o eksperimentu β.

Posledica 3.4.1. *Neka su α, β i γ eksperimenti čije se frekvencije ishoda mogu predstaviti tabelama kontigencije (1-5) i neka su date verovatnoće $\left\{P(C_k) = \frac{1}{K}\right\}$ realizacija ishoda $\{C_k\}$ eksperimenta γ. Ako eksperiment α sadrži maksimalnu uslovnu informaciju o eksperimentu β za realizovan eksperiment γ i eksperimenti γ i β nisu statistički nezavisni, tada složen eksperiment $\alpha\gamma$ sadrži maksimalnu informaciju o eksperimentu β.*

Dokaz: Razlikujemo dve mogućnosti: $H_\gamma(\beta) = 0$ i $H_\gamma(\beta) > 0$. Ako je $H_\gamma(\beta) = 0$ tada je $I(\gamma, \beta) = H(\beta)$ i odatle $I_\gamma(\alpha, \beta) = 0$, odnosno $I(\alpha\gamma, \beta) = H(\beta)$. U drugom slučaju, ako je $H_\gamma(\beta) > 0$, tada pošto eksperimenti γ i β nisu statistički nezavisni i ishodi eksperimenta γ su jednakoraspodeljeni, na osnovu Teoreme 3.3 sledi da eksperiment γ nosi maksimalnu informaciju o eksperimentu β. S obzirom na (19), iz maksimalnih vrednosti informacija $I(\gamma, \beta)$ i $I_\gamma(\alpha, \beta)$ neposredno sledi da će $I(\alpha\gamma, \beta)$ dostići svoju maksimalnu vrednost■

3.3 Zaključna razmatranja

Neka su γ, α i β slučajni eksperimenti koji se redom sprovode nad uzorkom obima N i čije frekvencije ishoda treba unakrsno da predstavimo pomoću trodimenzionalne tabele kontigencije. Ukoliko je poznata raspodela verovatnoća eksperimenta γ uz ispunjen uslov (20), tada na osnovu Teoreme 3.4 sledi da će eksperiment α sadrži maksimalnu uslovnu informaciju o eksperimentu β na svakom nivou eksperimenta γ, ako su eksperimenti α i γ statistički nezavisni i ishodi eksperimenta α su jednakoverovatni. Ocenjujući verovatnoće $\{P(C_k)\}$, $\{P(A_i)\}$, i $\{P(A_iC_k)\}$ redom sa relativnim frekvencijama

$$P(C_k) = \frac{N_{\bullet\bullet k}}{N},\ P(A_i) = \frac{N_{i\bullet\bullet}}{N} \text{ i } P(A_iC_k) = \frac{N_{i\bullet k}}{N}, \tag{28}$$

$i = 1, 2, ..., I, k = 1, ..., K$, iz uslova statističke nezavisnosti eksperimenata α i γ, $P(A_iC_k) = P(A_i)P(C_k)$, $i = 1, 2, ..., I$, $k = 1, ..., K$, dobijamo da za relativne frekvencije (28) treba da važi

$$\frac{N_{i\bullet k}}{N} = \frac{N_{i\bullet\bullet}}{N}\frac{N_{\bullet\bullet k}}{N},\ i = 1, 2, ..., I, k = 1, ..., K,$$

odnosno

$$\frac{N_{i\bullet k}}{N_{\bullet\bullet k}} = \frac{N_{i\bullet\bullet}}{N}, i = 1, 2, ..., I, k = 1, ..., K. \tag{29}$$

S obzirom da jednakoraspodeljenost ishoda eksperimenta α povlači da je

$$\frac{N_{i\bullet\bullet}}{N} = \frac{1}{I}, i = 1, 2, ..., I, \tag{30}$$

iz (29) i (30) sledi da na svakom nivou C_k eksperimenta γ, $k = 1, ..., K$, za frekvencije ishoda $\{A_iC_k\}$ treba da bude ispunjeno

$$N_{1\bullet k} = N_{2\bullet k} = ... = N_{I\bullet k} = \frac{N_{\bullet\bullet k}}{I}. \tag{31}$$

Odatle, ukoliko želimo da formiramo trodimenzionalnu tabelu kontigencije koja bi dala maksimalnu uslovnu informaciju $I_\gamma(\alpha, \beta)$, tada u marginalnoj tabeli kontigencije (α, γ), za svaki fiksiran ishod C_k eksperimenta γ, $k = 1, ..., K$, raspodela frekvencija ishoda $\{A_iC_k\}$ eksperimenta α treba da zadovoljava (31). To je moguće učiniti ako su ishodi eksperimenta α definisani tako da dopuštaju podjednako raspoređivanje frekvencija na svakom nivou eksperimenta γ.

Pored toga, ocenjujući verovatnoće $\{P_{C_k}(A_i)\}$, $\{P_{C_k}(B_j)\}$ i $\{P_{A_iC_k}(B_j)\}$ sa realizovanim relativnim frekvencija

$$P_{C_k}(A_i) = \frac{n_{i\bullet k}}{n_{\bullet\bullet k}},\ P_{C_k}(B_j) = \frac{n_{\bullet jk}}{n_{\bullet\bullet k}} \text{ i } P_{A_iC_k}(B_j) = \frac{n_{ijk}}{n_{i\bullet k}},$$

iz (16) i (17) možemo odrediti vrednost uslovne informacije $I_\gamma(\alpha, \beta)$

$$i_\gamma(\alpha,\beta) = \sum_{k=1}^{K} \frac{n_{\bullet\bullet k}}{n} i_{C_k}(\alpha,\beta) = \frac{1}{n}\sum_{k=1}^{K}\sum_{j=1}^{J}\sum_{i=1}^{I} n_{ijk} \log\left(\frac{n_{ijk}}{n_{i\bullet k}n_{\bullet jk}} n_{\bullet\bullet k}\right), \tag{32}$$

kao i vrednost potpune uslovne informacije $I_\gamma(\alpha,\beta) = H_\gamma(\beta)$

$$h_\gamma(\beta) = -\frac{1}{n}\sum_{k=1}^{K}\sum_{j=1}^{J} n_{\bullet jk} \log \frac{n_{\bullet jk}}{n_{\bullet\bullet k}}, \tag{33}$$

a odatle i $h_{\alpha\gamma}(\beta) = h_\gamma(\beta) - i_\gamma(\alpha,\beta)$ čime se kvantitativno ukazuje za koliko bita realizovan eksperiment α iscrpljuje neodređenost eksperimenta β za dati eksperiment γ.

Literatura

[1] Agresti 1996: A. Agresti, *An Introduction to Categorical Data Analysis*, John Wiley & Sons, INC, New York

[2] Bajić 2012: T. Bajić, *Analiza tabele kontigencije pomoću pojmova entropije i informacije*, Zbornik radova sa II matematičke konferencije Republike Srpske, 8-9. jun 2012. Trebinje, 249-260

[3] Good 1963: I. J. Good, *Maximum Entropy for Hypothesis Formulation, Especialy for Multidimensional Contigency Tables*, The Annals of Mathematical Statistics, Volume 34, Number 3, 911-934.

[4] Ivković 1980: Z. Ivković, *Matematička statistika*, Naučna knjiga, Beograd

[5] Stojanović 1980: S. Stojanović, *Matematička statistika*, Naučna knjiga, Beograd

[6] Yaglom 1973: A.M. Yaglom, I. M. Yaglom, *Probability and Information,* Third Edition, English Edition: D. Reidel Publishing Company, Dordrecht, Holland in co-edition with Hindustan Publishing Corporation, India

TREĆA MATEMATIČKA KONFERENCIJA REPUBLIKE SRPSKE
Trebinje, 7. i 8. juni 2013. godine

Grupe homologija i mogućnost njihovih izračunavanja

Gordana Jelić
Fakultet tehničkih nauka u Kosovskoj Mitrovici
gordana.jelic@pr.ac.rs

Stručni rad

Apstakt

Jedna od važnijih geometrijskih invarijanti mnogostrukosti je njena grupa homologija. Postoji nekoliko načina njenih definisanja. U ovom radu prikazan je jedan od načina definisanja grupe homologija sa realnim koeficijentima $H^k(M^n; R)$ na glatkoj mnogostrukosti M^n. Opisano je na nekoliko primera i efektno izračunavanje grupa homologija i neke njihove primene.

1 Definicija grupe homologija

Jedna od važnijih geometrijskih invarijanti mnogostrukosti je njena grupa homologija. Postoji nekoliko načina definisanja grupa homologija. Jedan od tih načina je sledeći.

Posmatrajmo zatvorene diferencijalne forme stepena k na mnogostrukosti M^n, tj. imamo

$$\omega = \sum_{i_1 < \cdots < i_k} a_{i_1 \ldots i_k} dx_{i_1} \wedge \cdots \wedge dx_{i_k}, \quad d\omega = 0.$$

Zatvorena diferencijalna forma naziva se *tačnom* (ili *homologičnom nuli*), ako je $\omega = d\omega'$, gde je ω' forma stepena $k-1$.

Definition 1.1. Grupom kohomologija $H^k(M^n; R)$ zove se faktorgrupa grupa svih zatvorenih formi stepena k po podgrupi tačnih formi, ili grupa klasa ekvivalencija zatvorenih formi sa tačnošću do tačnih formi ($\omega_1 \sim \omega_2$, ako je $\omega_1 - \omega_2 = d\omega'$).

2 Neka izračunavanja u grupi kohomologija

Proposition 2.1. *Za ma koju mnogostrukost M^n grupa $H^0(M^n; R)$ je linearni prostor one dimenzije q, iz koliko povezanih delova (komponenti) se sastoji mnogostrukost.*

Proof. Forma stepena 0 je skalarna funkcija $f(x)$ na mnogostrukosti. Ako je forma stepena 0 zatvorena, tada je $df(x) = 0$. To znači da je funkcija $f(x)$ lokalno konstantna, tj. konstantna na svakom povezanom delu mnogostrukosti. Zatvorene forme stepena 0 jesu nizovi od q konstanti, gde je q broj delova. Tvrđenje je dokazano, jer tačnih formi ovde nema. $\dashv$

Ako postoji glatko preslikavanje mnogostrukosti $f : M_1 \to M_2$, tada je određeno preslikavanje formi $\omega \to f^*(\omega)$, tako da je $d(f^*\omega) = f^*(d\omega)$. Prema tome, određeno je preslikavanje grupa kohomologija $f^* : H^k(M_2; R) \to H^k(M_1; R)$, pošto klase ekvivalencije prelaze jedna u drugu (pri preslikavanju f^* zatvorene forme ostaju zatvorene, a tačne ostaju tačne).

Theorem 2.1. *Ako su data dva glatka preslikavanja $f_1 : M_1 \to M_2$ i $f_2 : M_1 \to M_2$, i ta preslikavanja su homotopna, tada se preslikavanja grupa kohomologija f_1^* i f_2^* poklapaju, tj. $f_1^* = f_2^* : H^k(M_2; R) \to H^k(M_1; R)$.*

Proof. Neka je data glatka homotopija $F : M_1 \times I \to M_2$, gde je I odsečak, $1 \leq t \leq 2$ i $F(x, 1) = f_1(x)$, $F(x, 2) = f_2(x)$. Bilo koja forma Ω stepena k na $M_1 \times I$ ima oblik

$$\Omega = \omega_1 + \omega_2 \wedge dt, \ \Omega/_{t=t_0} = \omega_1(t_0),$$

gde je ω_1 forma stepena k i ω_2 forma stepena $k-1$ (lokalne koordinate u $M_1 \times I$ biraju se uvek u obliku $(x_1, ..., x_n, t)$ gde su $(x_1, ..., x_n)$ lokalne koordinate na M_1). Neka je ω ma koja forma stepena k na mnogostrukosti M_2. Tada je forma $F^*(\omega) = \Omega = \omega_1 + \omega_2 \wedge dt$, gde imamo da je

$$\begin{aligned} \omega_2 &= \sum_{i_1 < \cdots < i_k} a_{i_1 \ldots i_{k-1}}(x, t) dx_{i_1} \wedge \cdots \wedge dx_{i_{k-1}}, \\ \omega_1 &= \sum_{j_1 < \cdots < j_k} b_{j_1 \ldots j_k} dx_{j_1} \wedge \cdots \wedge dx_{j_k}. \end{aligned}$$

Odredimo formu $D(\Omega)$ stepena $k-1$ sledećom formulom:

$$D(\Omega) = \sum_{i_1 < \cdots < i_k} \left(\int_1^2 a_{i_1 \ldots i_{k-1}}(x, t) \right) dx_{i_1} \wedge \cdots \wedge dx_{i_{k-1}} = \int_1^2 \omega_2 dt$$

Očigledno je $D(\Omega)$ forma stepena $k-1$ na $M_1 \times I$.

Lemma 2.1. *Tačna je formula $dD(F^*(\omega)) \pm D(dF^*(\omega)) = f_2^*(\omega) - f_1^*(\omega)$ ili, ako je $\Omega = f^*(\omega)$ forma na M_2 i $F/_{t=2} = f_2$, $F/_{t=1} = f_1$, tačna je formula $dD(\Omega) \pm D(d\Omega) = \Omega/_{t=2} - \Omega/_{t=1}$.*

Proof. Izračunajmo $dD(\Omega)$ za bilo koju formu Ω na $M_1 \times I$, gde je $\Omega = \omega_1 + \omega_2 \wedge dt$. Imamo

$$dD(\Omega) = \sum_{i_1 < \cdots < i_k} \sum_j \left(\int_1^2 \frac{\partial a_{i_1 \ldots i_{k-1}}}{\partial x_j} dt \right) dx_j \wedge dx_{i_1} \wedge \cdots \wedge dx_{i_{k-1}},$$

$D(d\Omega) = D(d\omega_1) + D(d\omega_2 \wedge dt) =$

$$= D\left(\sum_{j_1<\cdots<j_k}\sum_q \frac{\partial b_{j_1\ldots j_k}}{\partial x_q} dx_q \wedge dx_{j_1} \wedge \cdots \wedge dx_{j_k} + \sum_{j_1<\cdots<j_k} \frac{\partial b_{j_1\ldots j_k}}{\partial t} dt \wedge dx_{j_1} \wedge \cdots \wedge dx_{j_k}\right) \neq$$

$$\neq D\left(\sum_{i_1<\cdots<i_k}\sum_p \frac{\partial a_{i_1\ldots i_k}}{\partial x_p} dx_p \wedge dx_{i_1} \wedge \cdots \wedge dx_{i_k}\right).$$

Odatle se vidi da je

$$dD(\Omega) + (-1)^{k+1}D(d\Omega) = \pm \sum_{j_1<\cdots<j_k} (b_{j_1\ldots j_k}(x,2) - b_{j_1\ldots j_k}(x,1))dx_{j_1} \wedge \cdots \wedge dx_{j_k}.$$

Kako je $\Omega/_{t=t_0} = (\omega_1 + \omega_2 \wedge dt)|_{t=t_0}$, dobijamo $dD(\Omega) + (-1)^{k+1}D(d\Omega) = \Omega/_{t=2} - \Omega/_{t=1}$. Prvi deo leme je dokazan. Ako je sada $\Omega = f^*(\omega)$, tada je $\Omega/_{t=t_0} = f^*_{t_0}(\omega)$, gde je $F(x, t_0) = f_{t_0} : M_1 \to M_2$. Posebno, za $t_0 = 1, 2$ dobijamo potreban rezultat. Lema je dokazana. ⊣

[Produžetak dokaza teoreme 2.1.] Neka je data zatvorena forma ω na M_2 (tj. $d\omega = 0$). Tada je forma $d(f_2^*(\omega) - f_1^*(\omega)) = dD(F^*(\omega)) \pm D(dF^*(\omega))$. Međutim, $dF^*(\omega) = F^*(d\omega) = 0$. Prema tome imamo $f_2^*(\omega) - f_1^*(\omega) = dDF^*(\omega)$, tj. razlika formi je tačna (ili su one homotopne). To i znači, po definiciji da homomorfizmi

$$f_1^* : H^k(M_2; R) \to H^k(M_1; R),$$

$$f_2^* : H^k(M_2; R) \to H^k(M_1; R)$$

se podudaraju na klasama ekvivalencije (kohomologija). Teorema je dokazana. ⊣

Definition 2.1. Mnogostrukosti M_1, M_2 nazivaju se homotopijski ekvivalentnim, ako se mogu naći takva (glatka) preslikavanja $f : M_1 \to M_2$, $g : M_2 \to M_1$, tako da su obe kompozicije $f \circ g : M_2 \to M_2$ i $g \circ f : M_1 \to M_1$ homotopne identičnim preslikavanjima

$$M_1 \to M_1(x \to x),\ M_2 \to M_2(y \to y).$$

Corollary 2.1. *Homotopijski ekvivalentne mnogostrukosti* $M_1 \overset{f}{\underset{g}{\rightleftarrows}} M_2$ *imaju iste grupe kohomologija.*

Proof. Posmatrajmo preslikavanja $f^* : H^k(M_2; R) \to H^k(M_1; R)$ i $g^* : H^k(M_1; R) \to H^k(M_2; R)$. Pošto su preslikavanja $f \circ g$ i $g \circ f$ homotopna, to su homomorfizmi $(f \circ g)^* = g^* \circ f^*$ i $(g \circ f)^* = f^* \circ g^*$ identični sa homomorfizmima grupa kohomologija po Teoremi 2.1:

$$1 = g^* \circ f^* : H^k(M_2) \to H^k(M_2),$$

$$1 = f^* \circ g^* : H^k(M_1) \to H^k(M_1).$$

Odatle, očigledno, sledi da su homomorfizmi f^* i g^* izomorfizmi, pri čemu su uzajamno inverzni: $f^* = (g^*)^{-1}$. Posledica je dokazana. ⊣

Corollary 2.2. *Grupe kohomologija euklidske ravni bez tačke* $R^2 \setminus Q$ *(ili prstena) iste su kao za sferu* S^1 *i imaju oblik*

$$\begin{aligned} H^k(S^1) &= H^k(R^2 \setminus Q) = 0, \quad k > 1, \\ H^1(S^1) &= H^1(R^2 \setminus Q) = R, \quad k = 1, \\ H^0(S^1) &= H^0(R^2 \setminus Q) = R, \quad k = 0. \end{aligned}$$

Proof. Razmotrimo grupe $H^k(S^1)$. Očigledno, one su trivijalne (pišemo $= 0$) ako je $k > 1$. Dalje, $H^0(S^1) = R$, pošto je sfera povezana, na osnovu Tvrđenja 1. Za ispitivanje grupe $H^1(S^1)$ uvedimo koordinatu φ, tako da φ i $\varphi + 2\pi n$ predstavljaju jednu tačku za cele brojeve n. Forma stepena 1 je forma oblika $a(\varphi + 2\pi) = a(\varphi)$. Uvek je $d\omega = 0$, pošto je dimenzija sfere S^1 jednaka 1. Postavlja se pitanje kada je forma $a(\varphi)d\varphi$ tačna.

To znači da je $a(\varphi)d\varphi = dF$, gde je $F(\varphi)$ periodična funkcija. Očigledno je

$$F(\varphi) = \int_0^{\varphi} a(\psi)d\psi + const.$$

Prema tome, funkcija $F(\varphi)$ je periodična tada i samo tada, kada je ispunjen uslov

$$\int_0^{2\pi} a(\psi)d\psi = 0 \quad \text{ili} \quad \int_{S^1} \omega = 0.$$

Dakle, forma stepena 1, $\omega = a(\varphi)d\varphi$ na sferi je tačna, ako i samo ako je ispunjen uslov

$$\int_{S^1} \omega = 0.$$

Odatle sledi da dve forme $\omega_1 = a(\varphi)d\varphi$ i $\omega_2 = a(\varphi)d\varphi$ određuju jednu i samo jednu klasu kohomologija, tada i samo tada, kada je

$$\int_{S^1} \omega_1 = \int_{S^1} \omega_2.$$

Prema tome, dobijamo da je $H^1(S^1, R) = R$. Time je posledica dokazana. $\dashv$

Corollary 2.3. *U orijentisanoj zatvorenoj rimanovoj mnogostrukosti* M^n *(neka je i povezana) grupa kohomologija* $H^n(M^n, R)$ *je netrivijalna*

Proof. Posmatrajmo element oblika Ω, gde imamo da je $\Omega = \sqrt{|g|}dx_1 \wedge \cdots \wedge dx_n$. Ako je niz oblasti lokalnih koordinata izabran u odgovarajućoj orijentaciji (tj. svi jakobijani su pozitivni), tada je Ω diferencijalna forma stepena n pri čemu je

$$\int_{M^n} \Omega > 0$$

(to je obim mnogostrukosti M^n). Očigledno je $d\Omega = 0$ jer je stepen forme Ω jednak n. Ako bi bilo $\Omega = d\omega$, tada bi po Stoksovoj formuli imali

$$\int\limits_{\partial M^n} \omega = \int\limits_{M^n} \Omega = \int\limits_{M^n} d\omega = 0$$

(jer je M^n zatvorena i nema granicu). Dobijamo protivurečnost. Posledica je dokazana. $\dashv$

Remark 2.1. Ako zatvorenu mnogostrukost M^n ne orijentišemo (npr. RP^2), tada je grupa $H^n(M^n, R)$ trivijalna. U stvari, element obima $d\sigma = \sqrt{|g|}d^n x$ pri zameni sa negativnim jakobijanom nije diferencijalna forma.

Može se pojasniti geometrijski smisao grupa kohomologija. Ako je M^n proizvoljna mnogostrukost i ω zatvorena forma stepena k, tada su određeni njeni "integrali po ciklovima". Neka je M^k zatvorena, orijentisana, k-dimenzionalna mnogostrukost. Pod "ciklom" u mnogostrukosti M^n podrazumevamo glatko preslikavanje $f : M^k \to M^n$.

Definition 2.2. *Periodom forme ω po ciklu (M^k, f)* nazivamo integral

$$\int\limits_{M^k} f^*(\omega).$$

Neka je sada N^{k+1} proizvoljno orijentisana mnogostrukost sa granicom $G = M^k$. Granica te zatvorene orijentisane mnogostrukosti može biti sastavljena od nekoliko delova. Pod "opnom" podrazumevamo preslikavanje $F : N^{k+1} \to M^n$ koje se poklapa sa f na M^k.

Theorem 2.2. *a) Za ma koji cikl (M^k, f) period bilo koje tačne forme $\omega = d\omega'$ jednak je nuli.*

b) Ako je cikl (M^k, f) granica opne (N^{k+1}, R), gde je M^k granica N^{k+1} i $F|_{M^k} = f$, tada je period ma koje zatvorene forme po ciklu (M^k, f) jednak nuli.

Proof. a) Ako je $\omega = d\omega'$, tada po Stoksovoj formuli imamo

$$\int\limits_{M^k} f^*(\omega) = \int\limits_{M^k} f^*(d\omega') = \int\limits_{M^k} d(f^*\omega) = \int\limits_{\partial M^k} f^*(\omega') = 0,$$

jer mnogostrukost M^k nema granicu.

b) Ako je M^k granica N^{k+1} (s obzirom na orijentaciju) i $F|_{M^k} = f$, tada po Stoksovoj formuli imamo

$$\int\limits_{M^k} F^*(\omega) = \int\limits_{N^{k+1}} dF^*\omega) = \int\limits_{N^{k+1}} F^*(d\omega) = 0.$$

Teorema je dokazana. $\dashv$

Literatura

[1] Бурбаки Н.: Общая топология. "Наука", Москва, 1975.

[2] Новиков П.С., Фоменко Т.А.: Елементи дифференциалной геометрии и топологии. "Наука", Москва, 1987.

TREĆA MATEMATIČKA KONFERENCIJA REPUBLIKE SRPSKE
Trebinje, 7. i 8. juni 2013. godine

O upotrebi metode najmanjih kvadrata za modelovanje ključnih parametara kvaliteta električne energije

Saša Mujović
Univerzitet Crne Gore, Elektrotehnički fakultet Podgorica
sasam@ac.me

Slobodan Đukanović
Univerzitet Crne Gore, Elektrotehnički fakultet Podgorica
slobdj@ac.me

Stručni rad

Apstakt

Nelinearni potrošači malih snaga su veoma zastupljeni u strukturi savremenih elektrodistributivnih sistema. Karakteriše ih naglašena mogućnost generisanja viših harmonika, čime utiču na degradaciju naponskih prilika u sistemu, odnosno ispravno funkcionisanje ostalih potrošača. Razvoj pametnih mreža (smart grids) dodatno aktuelizira problematiku kvaliteta električne energije, pa je veoma važno predvidjeti i kvantifikovati stepen negativnog uticaja nelinearnih potrošača malih snaga na naponske prilike u sistemu. S tim u vezi, u ovom radu je predstvaljen matematički model faktora strujne i naponske distorzije, uzrokovanih jednovremenim radom grupe računara, kao najrasprostranjenijih predstavnika pomenute grupe nelinearnih potrošača. Matematički modeli su razvijeni na bazi sprovednih mjerenja i simulacija, korišćenjem metode najmanjih kvadrata. Adekvatnost predloženih modela je provjerena dodatnim mjerenjima.

1 Uvod

Razvoj novih tehnologija (komunikacionih, energeske elektronike, obnovljivih izvora električne energije) rezultirao je razvojem koncepta pametnih mreže, kod kojih pitanje kvaliteta električne energije postaje izuzetno važno [1]. Nekoliko razloga treba navesti kao ključne u profilisanju važnosti adekvatnog kvaliteta električne energije:

- Veoma intenzivna upotreba elektronskih uređaja, dominantno nelinearnog karaktera, u strukturi elektrodistributivnih konzuma;
- Izuzetno naglašen negativan uticaj lošeg kvaliteta mrežnog napona (električne energije) na ispravno funkcionisanje pomenute kategorije potrošača;
- Pojava distribuiranih izvora (generatora) i težnja za što većom upotrebom obnovljivih izvora električne energije;

- Intermitentnost distribuiranih generator, proistekla iz nemogućnosti konstantog korišćenja određenih izvora energije (npr. sunčeva energija nije raspoloživa 24 sata tokom dana), negativno utiče na naponske prilike u sistemu;

- Poremećaji parametara kvaliteta elektrčne energije se opažaju i na strani proizvodnje i na strani potrošnje električne energije.

U klasi nelinearnih potrošača malih snaga ($I \leq 16$ A), personalni računari (PC) su posebno važni, imajući na umu njihovu široku rasprostranjenost i naglašen potencijal u smislu generisanja viših harmonika. Kao posljedica prekidačkog karaktera jedinice za napajanje računara, njegov harmonijski spektar struje je bogat neparnim harmonicima. Nivo prisustva viših (neparnih) harmonika se može kvantifikovati faktorom ukupne harmonijske distorzije struje (THD_I):

$$\mathrm{THD}_I(\%) = \frac{\sqrt{\sum\limits_{h=2}^{\infty} I_h^2}}{I_1} 100, \tag{1}$$

gdje h predstavlja red harmonika, a I_1 i I_h efektivne vrijednosti struja osnovnog i h-tog harmonika. U zavisnosti od konfiguracije računara, vrijednost THD_I faktora varira između 100% i 120% [2]. Jednovremena (simultana) upotreba veće grupe računara, negativno utiče na naponske prilike u mreži [3, 4]. Mjera ovog uticaja se može predstaviti faktorom ukupne harmonijske distorzije napona (THD_U)

$$\mathrm{THD}_U(\%) = \frac{\sqrt{\sum\limits_{h=2}^{\infty} U_h^2}}{U_1} 100, \tag{2}$$

gdje U_1 i U_h predstavljaju efektivne vrijednosti napona osnovnog i h-tog harmonika, respektivno. Upoređujući izmjerene vrijednosti faktora THD_I i THD_U sa limitima definisanim standardima [5] i [6] mogu se dobiti informacije o nivou uticaja potrošača na kvalitet napona napajanja.

Najprecizniji način utvrđivanja vrijednosti faktora THD_I i THD_U su direktna mjerenja, korišćenjem specijalizovane mjerne aparature (analizatora kvaliteta električne energije). Ipak, ovo je i najskuplji način, pa će u ovom radu biti prezentovan alternativni pristup, koji podrazumijeva matematičko modelovanje pomenutih faktora. Kao osnova za modelovanje poslužiće rezultati mjerenja, koja su sproveli autori rada, i rezultati simulacija dobijeni korišćenjem razvijenog Matlab Simulink modela.

Rad je podijeljen u šest poglavlja. U poglavlju 2 su prikazani rezultati mjerenja i simulacija. Poglavlja 3 i 4 su ključna i sadrže procedure izvođenja matematičkih modela faktora THD_I i THD_U. U poglavlju 5 je provjerena validnost predloženih modela. Konačno, u poglavlju 6 je dat zaključak rada.

2 Rezultati mjerenja i simulacija

2.1 Mjerenja u računarskom centru i rezultati simulacija

Mjerenja parametara kvaliteta električne energije su sprovedena u računarskom centru Fakulteta tehničkih nauka u Novom Sadu (RCFTN). U ovom Centru se nalaze 163 personalna računara (57, 50, 56 po fazama 1, 2, 3, respektivno). Detaljan opis mjerne procedure, sprovedene prema standardu IEC 61000-4-7 [7], i rezultata mjerenja je dat u [3]. U ovom radu izdvajamo određene rezultate koji će biti od značaja za dobijanje modela:

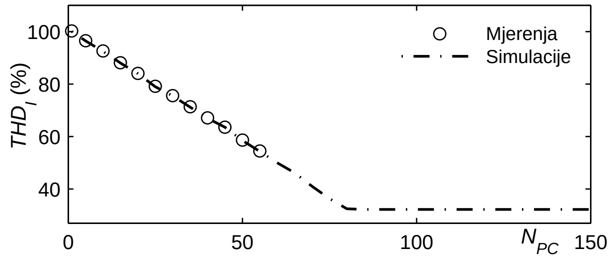
- Proračunata vrijednost krutosti mreže (S_{SC}) u tački priključka grupe računara Centra iznosi 4000 kVA;
- Kapacitivnost kondenzatora jedinica za napajanje svih računara u Centru iznosi 235 μF;
- Maksimalne registrovane vrijednosti faktora THD_I (po fazama) su iznosile 55.8%, 61.2% i 61.3%;
- Maksimalne registrovane vrijednosti faktora THD_U (po fazama) su iznosile 5.18%, 5.26% i 5.3%.

Simulacioni model RCFTN je razvijen u Matlab Simulink-u i detaljno je predstavljen u [3].

2.2 Mjerenja u banci

Dodatna mjerenja su sprovedena u poslovnoj zgradi NLB Montenegro banke u Podgorici. Ova mjerenja su imala za cilj da se verifikuju predloženi matematički modeli i provjeri njihova tačnost. Banka je opremljena sa 244 računara (83, 80, 81 računar po fazama 1, 2, 3, respektivno), a posebno važni rezultati mjerenja su:

- Proračunata vrijednost krutosti mreže u tački priključka grupe računara iznosi 5800 kVA;
- Kapacitivnost kondenzatora jedinica za napajanje 222 računara iznosi 235μF, dok 22 računara imaju kondenzatore kapacitivnosti 500 μF;
- Maksimalne registrovane vrijednosti faktora THD_I (po fazama) su iznosile 47.9%, 52.4% i 49.2%;
- Maksimalne registrovane vrijednosti faktora THD_U (po fazama) su iznosile 4.92%, 5.01% i 5.04%;
- Osim računara, tokom mjerenja u banci je bilo uključeno i linearno opterećenje (fluorescentne svetiljke, bojleri i sl.) aktivne snage 1350 W. Ovo je važno zbog analize tačnosti predloženih modela.



Slika 1: Rezultati mjerenja i simulacija THD_I faktora. Pošto je vrijednost THD_I fiksna za $N_{PC} > 150$, te vrijednosti nisu prikazane.

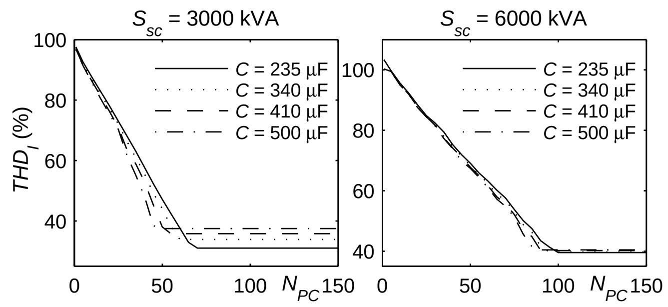
3 Modelovanje faktora THD_I

Istraživanjima predstavljenim u [2] i [3] zaključeno je da faktor THD_I grupe računara zavisi od broja računara u grupi (N_{PC}), kao i prethodno pomentih parametara S_{SC} i C. Treba naglasiti da je uticaj N_{PC} i S_{SC} dominantan. Nedostatak adekvatnog THD_I modela koji bi uvažio uticaj svih navedenih parametara i koji bi bio jednostavan za upotrebu, bio je glavni motiv ovog istraživanja. Vrijednosti faktora THD_I registrovanih u RCFTN, kao i vrijednosti dobijene simulacijama, predstavljene su na slici 1. Grupa od 200 računara (po fazi) je razmatrana u simulacijama. Takođe, vrijednosti S_{SC} i C parametara u simulacionom modelu su podešene na vrijednosti koje korespondiraju sa vrijednostima registrovanim u Centru (4000 kVA i $235\mu F$).

Sa slike 1 je uočljivo da faktor THD_I opada sa porastom broja N_{PC}. Ipak, ovaj rast nije kontinuiran, već pri određenom broju N_{PC} faktor THD_I ulazi u zonu zasićenja i svako dalje povećanje broja priključenih računara neće imati uticaj na vrijednost faktora THD_I. Da bi se ispitalo da li je ovakav zaključak validan i generalan, sprovedene su simulacije za različite vrijednosti parametara S_{SC} (3000, 4000, 5000 i 6000 kVA) i C (340, 410 i $500\mu F$). Ove vrijednosti su odabrane, kao najčešće u praksi. Na slici 2 su predstavljene THD_I krive dobijene za $S_{SC} = 3000$ kVA i 6000 kVA i sve navedene C vrijednosti. Ove dvije S_{SC} vrijednosti predstavljaju granične slučajeve niske i visoke krutosti mreže.

Analizirajući sliku 2 jasno je da je prethodno postavljeni zaključak o karakteru THD_I faktora ispravan. Takođe, uočljivo je da THD_I opada sa povećanjem S_{SC}. Kapacitivnost C utiče na nagib THD_I krive, na način što veći nagib korespondira manjim vrijednostima kapacitivnosti i obrnuto. Konačno, sa slike 2 se zaključuje da tačka saturacije (broj priključenih računara pri kojoj funkcija THD_I mijenja karakter i od linearno opadajuće funkcije postaje konstanta) zavisi i od C i od S_{SC}. Uzimajući u obzir navedene zaključke, faktor THD_I se može modelovati sledećom funkcijom:

$$\mathrm{THD}_I(N_{PC}, S_{SC}, C) = \begin{cases} f(S_{SC}, C)N_{PC} + g(S_{SC}), & \text{for } N_{PC} < N_{PC}^{sat} \\ \mathrm{THD}_I^{sat}(N_{PC}^{sat}, S_{SC}, C), & \text{for } N_{PC} \geq N_{PC}^{sat}. \end{cases} \quad (3)$$



Slika 2: Vrijednosti THD_I simulacija za $S_{SC} = 3000$ kVA (lijevo) and 6000 kVA (desno).

$$ \tag{4} $$

Dakle, zadatak je da se odrede funkcije $f(S_{SC}, C)$ i $g(S_{SC})$, kao i tačka saturacije N_{PC}^{sat}. Funkcija $\mathrm{THD}_I^{sat}\left(N_{PC}^{sat}, S_{SC}, C\right)$ je vrijednost THD_I funkcije za $N_{PC} = N_{PC}^{sat}$. Funkcija f, koja predstavlja funkciju nagiba se može aproksimirati polinomom drugog reda

$$P(S_{SC}, C) = \sum_{m=0}^{M} \sum_{n=0}^{M} a_{mn} S_{SC}^m C^n, \tag{5}$$

gdje je M red polinoma, dok su a_{mn}, $0 \leq m, n \leq M$. Koeficijenti a_{mn} će biti određeni metodom najmanjih kvadrata. Optimalne vrijednosti koeficijenata se dobijaju minimizacijom funkcije greške $E\left(S_{SC}, C\right)$:

$$E(S_{SC}, C) = \sum_{S_{SC}} \sum_{C} \left(f(S_{SC}, C) - P(S_{SC}, C)\right)^2. \tag{6}$$

Funkciju $E\left(S_{SC}, C\right)$ je moguće minimizovati rješavanjem sledećeg sistema jednačina:

$$\frac{\partial E(S_{SC}, C)}{\partial a_{pq}} = 0, \quad 0 \leq p, q \leq M. \tag{7}$$

Kombinujući (5), (6) i (7), dobija se

$$\begin{aligned} \sum_{m=0}^{M} \sum_{n=0}^{M} &\left(\sum_{S_{SC}} \sum_{C} S_{SC}^{m+p} C^{n+q} \right) a_{mn} \\ &= \sum_{S_{SC}} \sum_{C} f(S_{SC}, C) {S_{SC}}^p C^q, \quad 0 \leq p, q \leq M, \end{aligned} \tag{8}$$

Za očekivati je da polinomi većeg reda bolje aproksimiraju funkciju nagiba. Ipak, jedna od ključnih premisa ovog rada je težnja za postizanjem što jednostavnijeg modela. Upravo je to razlog da se aproksimacija izvrši polinomom prvog reda:

$$\begin{aligned} P^{(1)}(S_{SC}, C) &= \sum_{m=0}^{1}\sum_{n=0}^{1} a_{mn} S_{SC}^{m} C^{n} \\ &= a_{00} + a_{01}C + a_{10}S_{SC} + a_{11}S_{SC}C. \end{aligned} \tag{9}$$

Rješavajući sistem jednačina (8) polinomom $P^{(1)}(S_{SC}, C)$, dobijaju se optimalni koeficijenti:

$$\begin{aligned} a_{00} &= -0.7644 \\ a_{01} &= -1.966 \cdot 10^{3} \\ a_{10} &= 1.893 \cdot 10^{-5} \\ a_{11} &= 3.124 \cdot 10^{-1}. \end{aligned} \tag{10}$$

Funkcija $g(S_{SC})$ će biti modelovana polinomom prvog reda, u cilju zadovoljenja održavanja jednostavnosti modela

$$g(S_{SC}) = b_0 + b_1 S_{SC}. \tag{11}$$

Koeficijenti b_0 i b_1 se dobijaju rješavanjem sledećeg sistema linearnih jednačina:

$$\underbrace{\begin{bmatrix} 1 & S_{SC}^{(1)} \\ 1 & S_{SC}^{(2)} \\ \vdots & \vdots \\ 1 & S_{SC}^{(L)} \end{bmatrix}}_{\mathbf{X}} \underbrace{\begin{bmatrix} b_0 \\ b_1 \end{bmatrix}}_{\mathbf{b}} = \underbrace{\begin{bmatrix} \mathrm{THD}_I^{(1)} - f\left(S_{SC}^{(1)}, C^{(1)}\right) N_{PC}^{(1)} \\ \mathrm{THD}_I^{(2)} - f\left(S_{SC}^{(2)}, C^{(2)}\right) N_{PC}^{(2)} \\ \vdots \\ \mathrm{THD}_I^{(L)} - f\left(S_{SC}^{(L)}, C^{(L)}\right) N_{PC}^{(L)} \end{bmatrix}}_{\mathbf{y}}. \tag{12}$$

U relaciji (12), sa L su označene tačke odabrane sa simulacionih THD_I krivih. Svaka tačka je definisana sa tri parametra $(N_{PC}^{(l)}, S_{SC}^{(l)}, C(l))$, $l = 1, 2, \ldots, L$. Iako je sistem jednačina (12) sistem sa dvije nepoznate, za čije rješavanje bi bilo dovoljno $L = 2$, u ovom radu je korišten veći broj tačaka ($L > 2$), pa sistem (12) rješavamo metodom najmanjih kvadrata:

$$\mathbf{b} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}. \tag{13}$$

Za $L = 4$, dobijaju se koeficijenti:

$$\begin{aligned} b_0 &= 94.151 \\ b_1 &= 1.539 \cdot 10^{-3}. \end{aligned} \tag{14}$$

Estimacija tačke saturacije N_{PC}^{sat} se može sprovesti identičnom procedurom polinomijalne aproksimacije primijenjenom u slučaju funkcije $f(S_{SC}, C)$. Ponovo je, u cilju jednostavnosti dobijenog rješenja, odabran polinom prvog reda:

$$N_{PC}^{sat}(S_{SC}, C) \approx c_{00} + c_{01}C + c_{10}S_{SC} + c_{11}S_{SC}C. \tag{15}$$

Rješavanjem sistema jednačina (12) sa numeričkim vrijednostima $N_{PC}^{sat}(S_{SC}, C)$, umjesto $f(S_{SC}, C)$, dobijaju se koeficijenti:

$$\begin{aligned} c_{00} &= 73.606 \\ c_{01} &= -1.232 \cdot 10^5 \\ c_{10} &= 5.342 \cdot 10^{-3} \\ c_{11} &= 14.234. \end{aligned} \tag{16}$$

Konačni oblik THD_I modela dat je relacijom (16). Detaljnija procedura izvođenja data je u [8].

$$\begin{aligned} &\mathrm{THD}_I(N_{PC}, S_{SC}, C) \\ &= \begin{cases} (-0.7644 - 1.966 \cdot 10^3 C + 1.893 \cdot 10^{-5} S_{SC} + 3.124 \cdot 10^{-1} S_{SC} C) N_{PC} & \\ +1.539 \cdot 10^{-3} S_{SC} + 94.151, & \text{for } N_{PC} < N_{PC}^{sat}(S_{SC}, C) \\ \mathrm{THD}_I^{sat}(N_{PC}, S_{SC}, C), & \text{for } N_{PC} \geq N_{PC}^{sat}(S_{SC}, C), \end{cases} \end{aligned} \tag{17}$$

gdje je

$$\begin{aligned} \mathrm{THD}_I^{sat}(N, S_{SC}, C) = 37.887 &- 5.054 \cdot 10^4 C - 1.151 \cdot 10^{-3} S_{SC} \\ &-7.175 \cdot 10^{-1} S_{SC} C + 2.422 \cdot 10^8 C^2 + 1.011 \cdot 10^{-7} S_{SC}^2 \\ &-6.646 \cdot 10^4 S_{SC} C^2 + \ 1.938 \cdot 10^{-3} S_{SC}^2 C + 4.446 S_{SC}^2 C^2, \end{aligned}$$

i

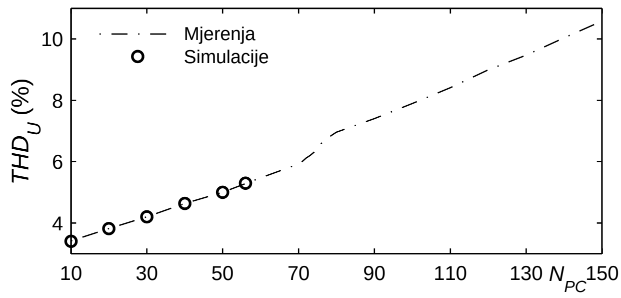
$$N_{PC}^{sat}(S_{SC}, C) \approx 73.606 - 1.232 \cdot 10^5 C + 5.342 \cdot 10^{-3} S_{SC} + 14.234 S_{SC} C.$$

4 Modelovanje faktora THD_U

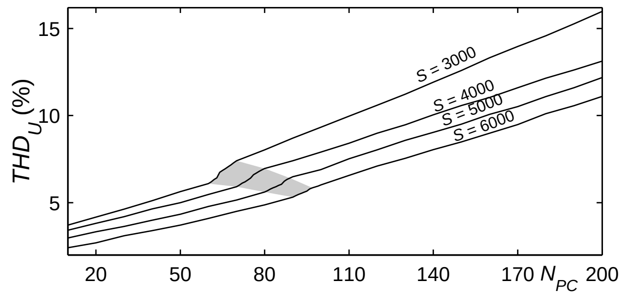
Motivacija za razvoj adekvatnog THD_U modela koji će kvantifikovati negativan uticaj simultanog rada grupe računara na kvalitet mrežnog napona je bila identična kao i u slučaju razvoja THD_I modela. N_{PC} i S_{SC} dominantno utiču na karakter THD_U faktora. Ostali parametri, kao što su poprečni presjek voda koji napaja grupu računara i početna naponska distorzija, imaju znatno manji uticaj i mogu se zanemariti. Mjerene vrijednosti THD_U faktora registrovane u RCFTN i vrijednosti dobijene simulacijama, predstavljene su na slici 3.

Analizirajući sliku 3, može se zaključiti da faktor THD_U raste sa povećanjem broja N_{PC}. Ipak, ovaj porast je specifičan i može se razmatrati podjelom THD_U krive u tri regiona (oblasti). U cilju verifikacije karaktera THD_U faktora pri različitim vrijednostima krutosti mreže, kao i utvrđivanja zavisnosti faktora THD_U od S_{SC}, sprovedene su dodatne simuacije prikazane na slici 4.

Uočljivo je da THD_U opada sa porastom S_{SC} parametra, što je i očekivano, jer veća krutost mreže znači i veću otpornost mreže na negativan uticaj priključenih potrošača (u konkretnom slučaju računara). Takođe, potvrđeno je postojanje tri



Slika 3: Vrijednosti THD_U simulacija i mjerenja za $S_{SC} = 4000$ kVA.



Slika 4: Vrijednosti THD_U simulacija za različite S_{SC} vrijednosti.

regiona THD_U krive. Sa porastom krutosti mreže, srednja regija se sve više "pomjera" ka većem broju N_{PC}, što će za velike vrijednosti S_{SC} dovesti do вЂњnestan ove regije i pojave uniformnosti THD_U funkcije na ukupnom N_{PC} interval. Unutar svake od uočenih oblasti THD_U funkcija se može aproksimirati linearnom funkcijom od N_{PC}. Uzimajući u obzira navedeno, THD_U se može modelovati na sledeći način

$$\mathrm{THD}_U(N_{PC}, S_{SC}) = f(S_{SC})N_{PC} + g(S_{SC}). \tag{18}$$

Slično kao i u slučaju modelovanja THD_I faktora, funkcije $f(S_{SC})$ i $g(S_{SC})$ se mogu predstaviti polinomima:

$$\mathrm{THD}_U(N_{PC}, S_{SC}) \approx \left(\sum_{i=0}^{P} k_i S_{SC}^i \right) N_{PC} + \sum_{j=0}^{Q} c_j S_{SC}^j, \tag{19}$$

gdje P i Q predstavljaju redove polinoma, dok k_i i c_j predstavljaju koeficijente polinoma, koji se mogu estimirati rješavanjem sledećeg sistema linearnih jednačina:

$$\mathbf{Ax} = \mathbf{b} \tag{20}$$

gdje je

$$\mathbf{A} = \begin{bmatrix} S_{(1)}^{P} N_{(1)} & ... & S_{(1)} N_{(1)} & N_{(1)} & S_{(1)}^{Q} & ... & S_{(1)} & 1 \\ S_{(2)}^{P} N_{(2)} & ... & S_{(2)} N_{(2)} & N_{(2)} & S_{(2)}^{Q} & ... & S_{(2)} & 1 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots \\ S_{(L-1)}^{P} N_{(L-1)} & ... & S_{(L-1)} N_{(L-1)} & N_{(L-1)} & S_{(L-1)}^{Q} & ... & S_{(L-1)} & 1 \\ S_{(L)}^{P} N_{(L)} & ... & S_{(L)} N_{(L)} & N_{(L)} & S_{(L)}^{Q} & ... & S_{(L)} & 1 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} k_P & ... & k_1 & k_0 & c_Q & ... & c_1 & c_0 \end{bmatrix}^T$$

$$\mathbf{b} = \begin{bmatrix} \mathrm{THD}_U^{(1)} & \mathrm{THD}_U^{(2)} & ... & \mathrm{THD}_U^{(L-1)} & \mathrm{THD}_U^{(L)} \end{bmatrix}^T.$$

U cilju pojednostavljenja zapisa u prethodnoj relaciji, umjesto N_{PC} i S_{SC} stoje oznake N i S. Takođe, matrica $\mathbf{A}$ je dimenzija $L \times (P + Q + 2)$, x i b su $(P+Q+2)\times 1$ i $L\times 1$ vektori, respektivno. L predstavlja broj razmatranih tačaka THD_U krive. Jedan red matrice $\mathbf{A}$ korespondira sa jednom THD_U tačkom. Kako je za rješavanje sistema jednačina (20) pretpostavljeno da je $L > P+Q+2$, vektor nepoznatih koeficijenata $\mathbf{x}$ ćemo dobiti metodom najmanjih kvadrata:

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}. \tag{21}$$

Estimacija (21) će biti sprovedena za svaki THD_U region pojedinačno. Takođe, redovi polinoma P i Q se biraju tako da finalni THD_U model bude što tačniji i jednostavniji. To znači da se počinje sa polinomima $P = 1$ i $Q = 1$ i provjerava tačnost THD_U modela. Zadovoljavajućom tačnošću smatra se tačnost od 0.1% izražena srednjom kvadratnom greškom. Finalni THD_U model dat je relacijom (22). Detaljnija procedura izvođenja modela data je u [9].

$$\begin{aligned} &\mathrm{THD}_U(N_{PC}, S_{SC}) = \\ &\begin{cases} (-3.043 \cdot 10^{-6} S_{SC} + 5.44 \cdot 10^{-2}) N_{PC} & \\ -4.527 \cdot 10^{-4} S_{SC} + 4.714, & \text{for } N_{PC} \leq \mathrm{LB}(S_{SC}), \\ (-2.931 \cdot 10^{-7} S_{SC} + 1.059 \cdot 10^{-1}) N_{PC} & \\ -1.348 \cdot 10^{-3} S_{SC} + 3.916, & \text{for } \mathrm{LB}(S_{SC}) < N_{PC} < \mathrm{RB}(S_{SC}), \\ (-7.647 \cdot 10^{-13} S_{SC}^3 + 1.219 \cdot 10^{-8} S_{SC}^2 & \\ -6.701 \cdot 10^{-5} S_{SC} + 1.767 \cdot 10^{-1}) N_{PC} & \\ -5.264 \cdot 10^{-4} S_{SC} + 4.466, & \text{for } N_{PC} \geq \mathrm{RB}(S_{SC}) \end{cases} \\ &\mathrm{LB}(S_{SC}) = 1 \cdot 10^{-2} S_{SC} + 30 \\ &\mathrm{RB}(S_{SC}) = -5 \cdot 10^{-7} S_{SC}^2 + 1.35 \cdot 10^{-2} S_{SC} + 34. \end{aligned} \tag{22}$$

5 Verifikacija tačnosti predloženih modela

U cilju utvrđivanja tačnosti THD_I modela (17), proračunate su vrijednosti srednje kvadratne greške (RMSE) za različite vrijednosti krutosti mreže i kapacitivnosti

Tabela 7: RMSE u zavisnosti od S_{SC} and C

S_{SC}[kVA] \ C [μF]	235	340	410	500
3000	2.87	1.88	2.19	2.49
4000	0.94	1.79	2.11	2.77
5000	1.55	2.36	2.47	3.12
6000	1.52	1.06	1.69	3.16

jedinice za napajanje računara (Tabela 7). Maksimalna vrijednost greške iznosi 3.16%, što je potpuno prihvatljivo.

Za vrijednosti S_{SC} i C registrovanim mjerenjima u banci (poglavlje 2.2), modelom (17) se dobijaju vrijednosti 45.35%, 47.51% i 46.97%, što je blisko mjerenim vrijednostima. Slična procedura je sprovedena i za provjeru tačnosti predloženog THD$_U$ modela (22). Izračunate vrijednosti srednje kvadratne greške iznose 0.09%, 0.09%, 0.071%, 0.08%, 0.057% za S_{SC} =3000, 4000, 5000 i 6000 kVA, respektivno. Dodatno, pri vrijednostima S_{SC} i N_{PC}, karakterističnim za najopterećeniji fazni provodnik voda koji napaja grupu računara u banci, modelom (22) se dobija vrijednost 5.14%, što potvrđuje njegovu validnost. Mala razlika između mjerenih i vrijednosti dobijenih primjenom modela (17) i (22), javlja se kao posljedica priključenja dodatnog linearnog opterećenja (vidjeti poglavlje 2.2).

6 Zaključak

Strujna i naponska distorzija uzrokovana jednovremenim radom grupe računara može se uspješno modelovati primjenom metode najmanjih kvadrata. Predloženi modeli, dobijeni na bazi rezultata mjerenja i simulacija, su adekvatne tačnosti, što je potvrđeno dodatnim setom mjerenja. Budući rad u ovoj oblasti će biti usmjeren na proširenju ovih modela kroz uvažavanje uticaja ostalih tipova nelinearnih potrošača malih snaga na kvalitet napona napajanja.

Literatura

[1] S.-H. Jo, S. E. Son, J-W. Park, “On Improving Distortion Power Quality Index in Distributed Power Grids,”*IEEE Trans. On Smart Grid*, vol. 2, issue: 1, pp. 586–595, March 2013.

[2] V. A. Katić, B. Dumnić, S. Mujović, J. Radović, “Effects of Low Power Electronics & Computer Equipment on Power Quality at Distribution Grid - Measurements and Forecast,”In *Proc. Int. IEEE Conf. on Industrial Technology*, Hammamet, 2004, pp.585–589.

[3] S. Mujović, V. A. Katić, J. Radović, “Improved Analytical Expression for Calculating Total Harmonic Distortion of PC Clusters,”*Electric Power Systems Research*, vol. 81, no. 7, pp. 1317–1324, 2011.

[4] V. A. Katić, S. V. Mujović, V. M. Radulović, J. S. Radović, “The Impact of the Load Side Parameters on PC Cluster’s Harmonics Emission,”*Advances in Electrical and Computer Engineering*, vol. 11, no. 1, pp. 103–110, 2011.

[5] EN50160 Standard, “Voltage Characteristics of Electricity Supplied by Public Distribution Systems,”*CENELEC*, 1994.

[6] IEEE Standard 519-1992, “IEEE Recommended Practices and Requirements for Harmonic Control in Electric Power Systems,”*IEEE Press*, 1993.

[7] IEC Standard 61000-4-7, “General Guide on Harmonics and Inter-harmonics Measurements and Instrumentation for Power Supply Systems and Connected Equipment, ”*IEC*, 2002.

[8] S. Mujović, S. Đukanović, V. Radulović, V. A. Katić, “Determination of PC Cluster Current Harmonic Distortion through Mathematical Modeling of Measured and Simulated Data,”*International Transactions on Electrical Energy Systems*, submitted for publication.

[9] S. Mujović, S. Đukanović, V. Radulović, V. A. Katić, M. Rašović, “Least Squares Modeling of Voltage Harmonic Distortion Due to PC Cluster Operation,”*Advances in Electrical and Computer Engineering*, accepted for publication.

TREĆA MATEMATIČKA KONFERENCIJA REPUBLIKE SRPSKE
Trebinje, 7. i 8. juni 2013. godine

Cinderella - način da vidimo apstraktnu matematiku

Veljko Vranić
University of Belgrade, Faculty of Mathematics
byblos94@gmail.com

Djordje Baralić
Mathematical Institute SASA
djbaralic@mi.sanu.ac.rs

Stručni rad

Apstrakt

Softver Cinderella pruža brojne mogućnosti za rad u brojnim disciplinama matematike: različitim vrstama geometrije, analizi, fraktalima, kompleksnoj i konveksnoj analizi, kao i u mehanici i fizici. Izložićemo neke od njegovih alatki i predstaviti naš dosadašnji rad na razvijanju njegovih opcija za rad sa algebarskim krivama višeg reda. Problem vizuelizacije apstraktnih objekata u matematici je jedan od prvih problema sa kojima se suočavamo, a pokazaćemo da nam dobar sotver uz jaku matematičku osnovu može pomoći da ga uspešno prevazidjemo.

1 Uvod

Od kada su se pojavili personalni računari i postali naširoko popularni, počele su da se javljaju ideje za stvaranje najrazličitijih softvera koji bi olakšali čivot savremenom čoveku. Računari su se u naučne svrhe najpre koristili za numeričke proračune, kasnije za čisto numeričke simulacije, dok je danas bavljenje naukom gotovo nezamislivo bez njih. Oni nam služe za skladištenje i lagodan pristup podacima, jednostavnu i besplatnu komunikaciju sa svetom, brzo širenje ideja. Druga stvar koja je tekla paralelno sa razvojem softvera za gorepomenute olakšice, tekao je i razvoj računarske grafike, sa ciljem jednostavnije upotrebe računara ali i u svrhu zabave korisnika.
Danas postoje hiljade i hiljade raznoraznih programa koji povezuju ove dve "oblasti" računarstva.

Cinderella pripada disciplini u računarskoj naukama poznatijom kao "Dinamička geometrija". Koristeći miša i uređaje koji imaju ekrane velike rezolucije, možemo da crtamo prave i krugove, koristimo njihove preseke i štampamo naše crteže. Prva korist od upotrebe je uvećana preciznost crteža - bićete u mogućnosti da koristite pravi presek dve duži iako niste vešti u crtanju. Bitno poboljšanje u odnosu na vreme lenjira i šestara jeste da računar čuva vašu konstrukciju, i

lako može da vrati par koraka unazad. Ključ svega ovoga je naravno u interaktivnosti: Odaberete tačku, pomerate je i trenutno vidite kako se konstrukcija pomera. Tvorci ovog softvera su *Jurgen Richter-Gebert* i *Ulrich Kortenkamp*.
Međutim, neispravno je reći da je *Cinderella* samo program čije su mogućnosti

ograničene samo na geometriju. Njegov spektar mogućnosti obuhvata sve od jednostavnijih planimetrijskih konstrukcija, preko mogućnosti rada sa hiperboličkom geometrijom, do stvaranja fraktala i baratanja algebarskim krivama drugog i trećeg reda. Pored ovih mogućnosti *Cinderella* ima dodatke koji omogućuju naprednije fizičke simulacije, vizuelizaciju grafovskih algoritama, kao i manipulaciju nad slikama i zvukom.
U ovom radu prikazaćemo najznačajnije alate koje nam ovaj softver pruža.

2 Glavne karakteristike

Jedna od najvažnijih osobina *Cinderella*-e je to što je u potpunosti zasnovana na kompleksnom projektivnom prostoru. Upravo zahvaljujući ovakvoj osnovi, *Cinderella* može da sagledava pojedinačnu konstrukciju iz više uglova u isto vreme.

2.1 Euklidska ravan

Uobičajeno gledište koje se koristi jeste *Euklidska ravan*. Naravno, ovo nije prava matematička Euklidska ravan, već računarski prikaz jedne njene verzije.
Možete odabrati deo Euklidske ravni koji želite da vidite uveličavajući i translirajući okvir ekrana. Zamisli proizvoljan pravougaonik koji možete postaviti na proizvoljni deo Euklidske ravni, i sve što je unutar tog pravougaonika će biti prikazano unutar ekrana.

2.2 Sferna projekcija

Unutar *Cinderella* Euklidska ravan je proširena "linijom u beskonačnosti", koja pretvara ravan u prostor poznat kao projektivna ravan. Ova projektivna ravan može biti zamisljena kao jedinična sfera u trodimenzionalnom prostoru : Tačke se preslikavaju u par antipodnih tačaka na sferi, prave se preslikavaju u velike krugove oko sfere. Preslikavanje je dato centralnom projekcijom ravni locirane na $z = 1$ u $3D$ prostoru kroz koordinatni početak na jediničnu sferu. "Severni pol" sfere dodiruje ravan u koordinatnom početku. Njen ekvator ne odgovara ni jednoj tački Euklidove ravni, pa ih stoga zovemo "tačke u beskonačnosti."
Možemo koristiti pogled sferne projekcije da bi bolje razumeli pojam beskonačnosti. Mnogi ljudi su čuli da se paralelne linije sreću u beskonačnosti, ali nisu mogli to da potvrdi iskustvom, ili da zamisle tako nešto. Sa *Cinderella*-om možemo otvoriti

sferični pogled i namestiti dva velika kruga da se seku tačno na ekvatoru, dok će u Euklidskom pogledu te dve prave biti paralelne.

2.3 Polarne koordinate

Uloge tačke i prave u Projektivnoj Geometriji mogu se medjusobno menjati. Omesto prave koja spaja dve tačke, imamo presečnu tačku dve prave, itd. Ovo se zove dualnost. *Cinderella* može automatski da prikazuje konstrukciju u polarnom Euklidskom ili polarnom sferičnom pogledu. Ovo je veoma korisno, pogotovu ako pokušavate da dokažete teoremu: Nekada je polarna verzija mnogo jednostavnija za razumevanje i dokazivanje. Takođe polarni pogled nije ograničen samo na tačke i prave.

2.4 Poenkareov disk model hiperboličke geometrije

Hiperbolička geometrija je verovatno najpoznatija ne-Euklidska geometrija. Fundamentalni objekat Poenkareovog disk modela, koji igra ulogu beskonačnosti, jeste spoljni krug.
Hiperbolička ravan je sjajno igralište za geometrijska istraživanja. Ovde možete proveriti kako izgledaju vaše omiljene teoreme u hiperboličkoj ravni.

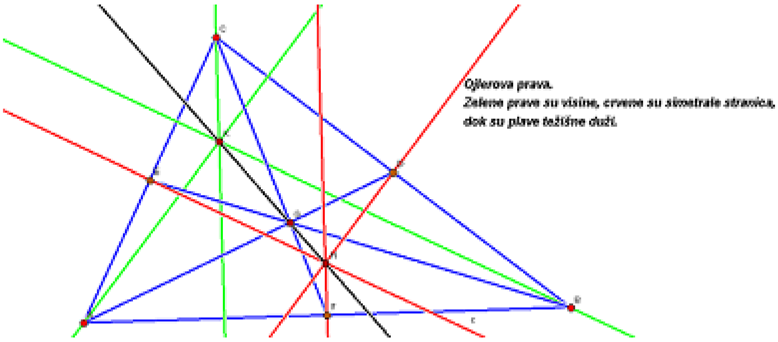
2.5 Ostale korisne osobine

Što se tiče ostalih osobina *Cinderella*-e ona ima ugradjen kompajler za LaTeX, "izvoženje" slike visoke rezolucije iz konstrukcije, ali i mogućnost jednostavne integracije bilo koje *Cinderella*-ine konstrukcije na internet stranice.

3 Osnovni alati

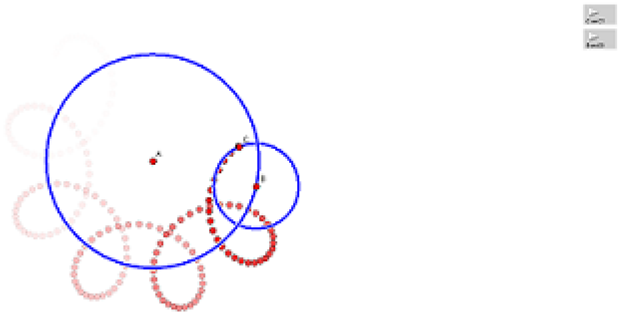
U ovom delu rada prikazaćemo neke od najzanimljivih konstrukcija koje se mogu kreirati uz vrlo malo poznavanja programa i vrlo intuitivno.
Jedna od najpoznatijih pravih elementarne geometrije je Ojlerova prava, definisana za sve trouglove osim jednakostraničnog. Ona prolazi kroz ortocentar, težište i centar opisane kružnice datog trougla. Pored toga, njoj pripada i centar kružnice devet tačaka i neke skorije ustanovljene tačke trougla. Ono što nam *Cinderella* pruža jeste ta dinamika konstrukcije, naime jednostavnim pomeranjem jednog od temena trougla cela konstrukcija se pokreće ali ono što ostaje kostantno jeste pripadnost pomenutih tačaka Ojlerovoj pravoj.

Pored ovih elementarno geometrijskih alata, *Cinderella* može napraviti periodičnu animaciju kretanja odredjene tačke po pravoj, krugu ili nekoj drugoj krivoj drugog reda, a samim tim i transformaciju koja nastaje pomeranjem te tačke. Još u staroj Grčkoj, posmatranje neba je navelo najveće umove tog vremena da



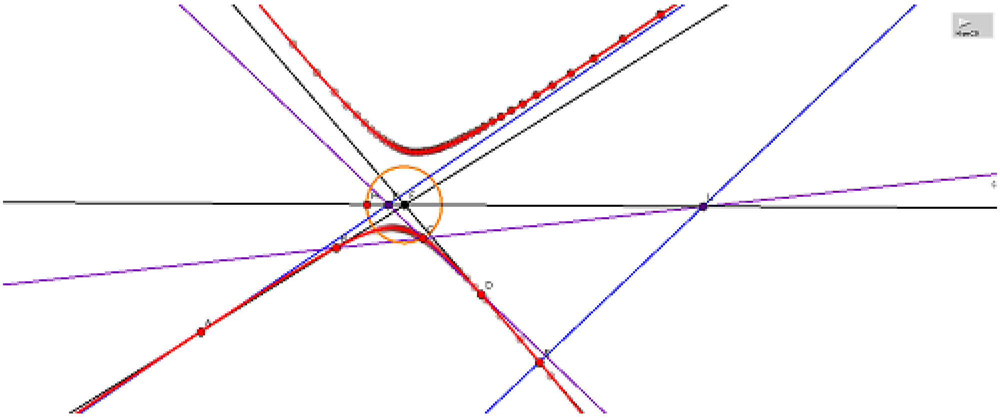
Slika 1: Ojlerova prava

razmišljaju o kretanju nebeskih tela. Kao zaostavštine toga perioda ostale su nam najrazličitije zanimljive krive. Jedna od njih je epickloida. Koristeći alat za pravljenje animacija u *Cinderella*-i, možemo stvoriti ovako lepe slike kao što je



Slika 2: Epicikla

Koristeći se animacijama u *Cinderella*-i možemo konstruisati i poznate konike. Konike su krive drugog redam koje zapravo znamo pod drugim imenom. Sve konike možemo podeliti na hiperbole, parabole i elipse. Načini za njihove konstrukcije postojali su još u Njutnovo vreme, a jedna od najpoznatijih metoda za konstrukciju konika jeste *Braikenridge-Maclaurin* metoda. U osnovi ove metoda nalazi se parnjak Paskalovove teoreme, jedna od prvih teorema projektivne geometrije. Koristeći *Cinderella*-u vrlo jednostavno možemo konstruisati bilo koju koniku kojoj zadamo pet tačaka. Ovo ćemo uraditi animacijom.

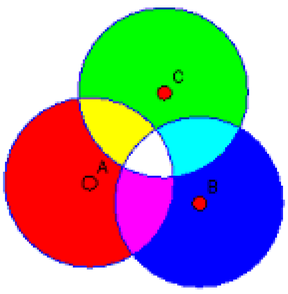


Slika 3: Braikenridge-Maclaurin konstrukcija

4 *CindyScript*

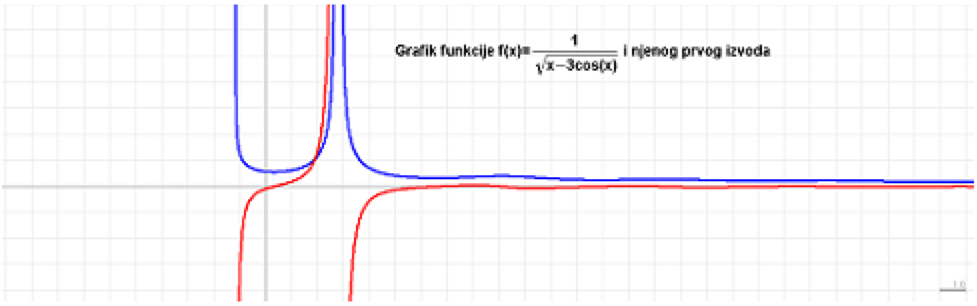
Nastankom ovog softvera rodila se i želja da se kreira programski jezik koji bi živeo u simbiozi sa dosadasnjim funkcijama i mogućnostima *Cinderella*-e. To je autorima pošlo za rukom u verziji 2.0 . Njegova sintaksa je vrlo intuitivna svima koji iole poznaju programiranje i vrlo jednostavna. Koristeći se ovim dodatkom, *Cinderella* postaje sasvim ozbiljan alat za sve vrste simulacija i proračuna. U poslednjim verzijama programa pojavljuju se neke veoma zanimljive opcije, kao što je pogadjanje koji je to iracionalan broj čija je realna vrednost poznata.

Krenućemo od najjednostavnijih primera. *CindyScript*, a sa njom i *Cinderella* imaju mogućnost baratanja skupovima, to jest, skupovnim operacijama. U sledećem primeru, skup će predstavljati unutrašnost i obod pojedinačno svakog od tri kruga. Svakom od skupova će biti pridružena po jedna od RGB boja, a njih presek će zapravo predstavljati kombinaciju te dve boje.



Slika 4: Venov dijagram

Kada su u pitanju funkcije, *CindyScript* je u mogućnosti da iscrtava grafike polinomskih, racionalnih, iracionalnih i trigonometrijskih funkcija, kao i bilo koju složenu funkciju nastalu kompozicijom navedenih elementarnih funkcija. Pored opcije iscrtavanja grafika, moguće je izračunati i iscrtati prvi, drugi, .. itd. izvod bilo koje od malopre navedenih funkcija. Zadavanje funkcije je vrlo intuitivno i gotovo identično eksplicitnom zadavanju na kakvo smo navikli u matematici. Na narednoj slici plavom bojom je prikazan grafik funkcije, a crvenom njen prvi izvod.

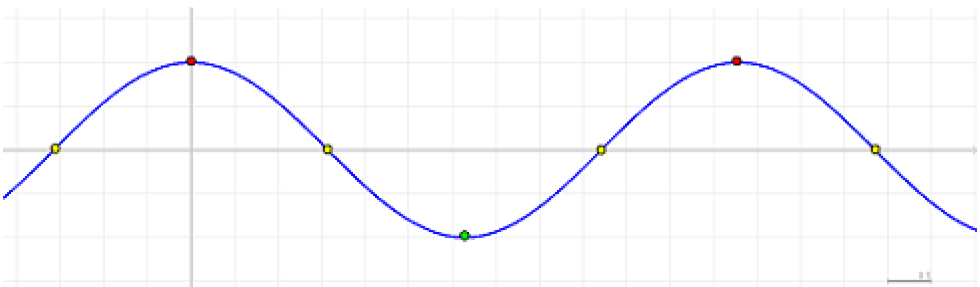


Slika 5: Funkcija i njen prvi izvod

Kod koji je iza ove slike je sledeći

```
f(x):=1/sqrt(x+3*cos(x));
g(x):=d(f(#),x);
plot(f(x),color->(0,0,1),size->3);
plot(g(x),color->(1,0,0),size->3);
```

Ukoliko nam zatrebaju nule, lokalni minimumi ili maksimumi ili tačke infleksije (promene znaka prvog izvoda) i to je vrlo lako izračunati i nacrtati u pomoć *CindyScript*-a. Na narednoj slici crvene tačke predstavljaju lokalne maksimume, zelene lokalne minimume, dok su nule i tačke infleksije predstavljene žutom bojom.

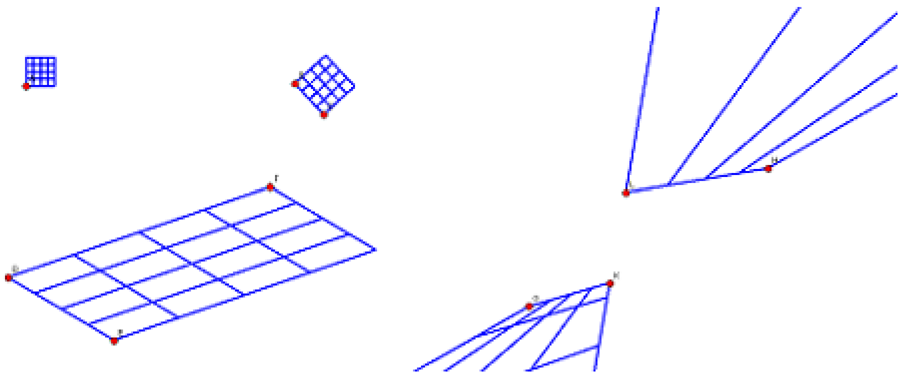


Slika 6: Značajne tačke funkcije

Kod za tu prethodnu slike je i jednostavniji nego prethodni, sve zahvaljujući jednostavnosti kojom se ovaj programski jezik ističe.

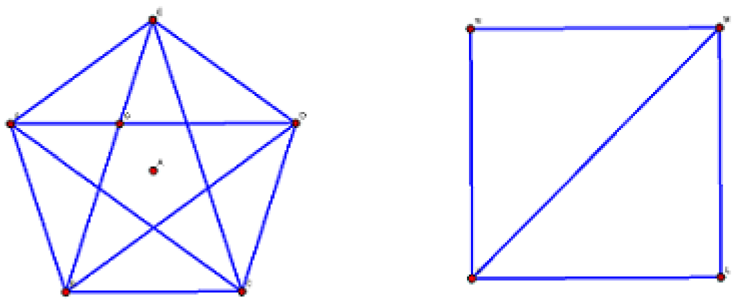
```
f(x):=cos(x);
plot(f(x),zeros->true,inflections->true,maxima->true,minima->true);
```

Izometrijske transformacije su očekivane. Pored toga, *Cinderella* vam omogućava da pored pravljenja konstrukcije promenite bazu tog prostora i samim tim primenite afino ili projektivno preslikavanje. Bazni vektori se zadaju jednostavno. Takodje, ukoliko nam je potrebna baza za preslikavanje jedne slike u drugu, možemo tražiti od *CindyScript*-a da to reši.



Slika 7: Različite baze prostora

Najnovija mogućnost *CindyScript*-a jeste mogućnost pogadjanja iracionalnog izraza od zadate realne vrednosti. Ova najsofisticiranija funkcija omogućava da nadjemo simboličko značenje vrednosti realnog broja. Funkcija *guess* traži vrednost medju formulama oblika $a + b\sqrt{c}$, gde su a, b i c racionalni brojevi. Ova funkcija ekstremno je dobra za pronalaženje skrivenih osobina geometrijskih konstrukcija. Slika koja ide u prilog tome je sledeća.



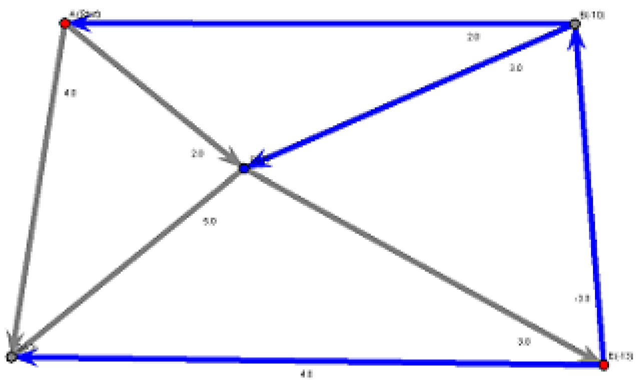
Slika 8: Pentagram i kvadrat - skrivene osobine

Sve što je potrebno uraditi da bi dobili vrednosti koje se kriju iza odnosa duži $\frac{EG}{GD}$, kao i $\frac{KM}{KL}$ jeste :

```
println(guess(|B,G|/|G,E|));
println(guess(|K,M|/|K,N|));

Rezultat : 1/2+1/2*sqrt(5)
           sqrt(2)
```

Teorija grafova je oblast koju u dosadašnjem delu rada *Cinderella* nikako nije doticala. Medjutim, alat pod imenog *Visage* služi upravo tome. Pomoću njega možemo crtati sve vrste grafova : težinske, usmerene, planarne, itd. Ali alat za crtanje grafova i nije neka velika novina. Svako ko je ikada učio grafovske algoritme i to uspešno sigurno je proveo prilično vremena crtajući slučajeve i simulirajući algoritam, korak po korak. *Cinderella* omogućava simulaciju par trenutno implementiranih algoritama na zadatom grafu. U trenutnoj verziji implementirani su **Bellman-Ford**, **Dijkstra**, **Ford-Fulkerson**, **Euler Tour**, **Bipartite Mačing**, itd. Pored svih ovih algoritama, moguća je i nadogradnja *Cinderella*-e sopstvenim algoritmima. Takodje moguća je kooperacija izmedju *CindyScript*-a i *Visage*-a. Sledeća slika prikazuje jedan korak u simulaciji Bellman-Ford algoritma.



Slika 9: **Bellman-Fordov** algoritam

I pored ovih fantastičnih funkcija i dodataka, *CindyScript* nudi mnogo više.

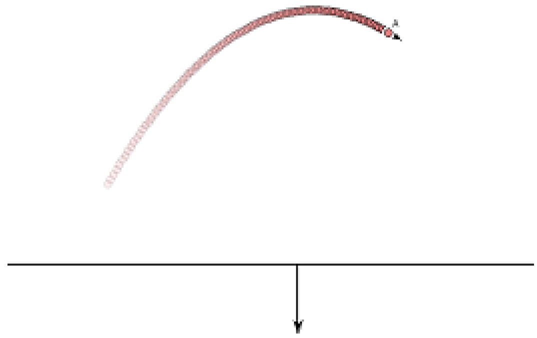
5 *CindyLab*

Niz do sada predstavljenih funkcija nije izlazio van domena matematike. *Cinderella* ima i skup funkcija koje se nazivaju *CindyLab*.

Ova mini laboratorija ima mogućnost različitih fizičkih simulacija. Koristeći se jednostavnih fizičkim predmetima moguće je stvoriti krajnje komplikovane eksperimente. U pozadini simulacija nalazi se provereni proračuni sa *RK 4* metodom.

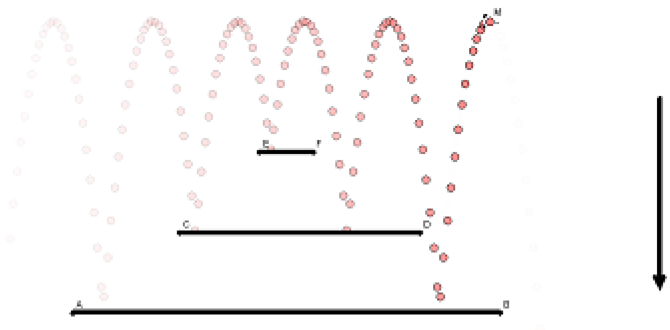
Ipak za prave simulacije se preporučuje softveri koji su specijalizovani za to, dok su one nastale u *Cinderella*-i pre svega za edukativne svrhe.

Sledeća slika sadrži slikovit prikaz parabole koju opisuje telo ispaljeno kao kosi hitac. Strelica na dole označava gravitaciju, koju je moguće zadati vektorom.



Slika 10: Kosi hitac

Pored ovoga, na sledećem jednostavnom primeru možemo švatiti pojam zakona o održanju energije.



Slika 11: Zakon o održanju energije

Literatura

[1] Cinderella - način da vidimo apstraktnu matematiku", Maturski rad, Prva kragujevačka gimnazija, 2013. V. Vranić

TREĆA MATEMATIČKA KONFERENCIJA REPUBLIKE SRPSKE
Trebinje, 7. i 8. juni 2013. godine

Uopštenje nejednakosti paralelograma

Nebojša Elez
Filozofski fakultet Pale

Marko Ćitić
Filozofski fakultet Pale
citicm@yahoo.com

Stručni rad

Apstrakt

U ovom radu je dokazano više geometrijskih nejednakosti. Pokazana je nejednakost koja uopštava jednakost trapeza. Kao njena posledica je dobijena nejednakost koja uopstava jednakost paralelograma. U kombinaciji s Ptolomejevom nejednakoscu dobijene su još dve nejednakost četvorougla od kojih se prva pretvara u jednakost u slučaju tetivnog trapeza, a druga u slucaju pravougaonika. Za kraj je pokazana nejednakost analogna prethodnim koja se odnosi na proizvoljan šestougao

1 Nejednakost paralelograma

Poznato uopštenje jednakosti paralelograma glasi: Ako su a, b, c i d stranice četverougla, a e i f dijagonale, tada je:

$$a^2 + b^2 + c^2 + d^2 \geq e^2 + f^2.$$

Jednakost važi ako i samo ako je četvorougao paralelogram.

Teorema 1.1. (Nejednakost trapeza) Neka su a, b, c i d stranice četverougla, a e i f dijagonale, gdje su a, c i b, d parovi naspramnih stranica. Tada važi nejednakost:

$$2ac + b^2 + d^2 \geq e^2 + f^2,$$

gdje važi jednakost ako i samo ako je četverougao trapez sa osnovicama a i c.

Dokaz 1.1. Obilježimo četverougao sa vrhovima A, B, C, D, gdje je $a = \overline{AB}$, $b = \overline{BC}$, $c = \overline{CD}$, $d = \overline{AD}$, $e = \overline{AC}$, $f = \overline{BD}$. Stavimo $\overrightarrow{a} = \overrightarrow{AB}$, $\overrightarrow{b} = \overrightarrow{BC}$, $\overrightarrow{c} = \overrightarrow{CD}$. Tada važi:

$$e^2 + f^2 - b^2 - d^2 = |\overrightarrow{AC}|^2 + |\overrightarrow{BD}|^2 - |\overrightarrow{BC}|^2 - |\overrightarrow{DA}|^2 =$$

$$= (\overrightarrow{a} + \overrightarrow{b}) \cdot (\overrightarrow{a} + \overrightarrow{b}) + (\overrightarrow{b} + \overrightarrow{c}) \cdot (\overrightarrow{b} + \overrightarrow{c}) - b^2 - (\overrightarrow{a} + \overrightarrow{b} + \overrightarrow{c}) \cdot (\overrightarrow{a} + \overrightarrow{b} + \overrightarrow{c}) =$$

$$= -2\overrightarrow{a} \cdot \overrightarrow{c} \leq 2ac.$$

Jednakost važi ako i samo ako su vektori $\overrightarrow{AB}$ i $\overrightarrow{CD}$ paralelni, tj. ako je četverougao trapez. SabirajuД‡i dvije nejednakosti trapeza dobijamo nejednakost paralelograma koja je jača od klasične nejednakosti
paralelograma.

Posledica 1.1.1. Ako su a, b, c i d stranice četverougla, a e i f dijagonale, tada važi nejednakost:

$$(a+c)^2 + (b+d)^2 \geq 2e^2 + 2f^2,$$

gdje jednakost važi ako i samo ako je četvereougao paralelogram.
Sabirajući nejednakost trapeza i Ptolomejevu nejednakost dobijamo tvrjenje:

Posledica 1.1.2. Za četverougao važi nejednakost

$$4ac + (b+d)^2 \geq (e+f)^2$$

gdje važi jednakost ako i samo ako je četverougao jednakokraki (tetivni) trapez sa osnovama a i c.
Iz prethodne posledice slijedi tvrjenje:

Teorema 1.2. (Nejednakost pravougaonika) Za četvorougao sa parovima naspramnih stranica a, c i b, d i sa dijagonalama e i f važi nejednakost

$$(a+c)^2 + (b+d)^2 \geq (e+f)^2$$

gdje važi jednakost ako i samo ako je četverougao pravougaonik.

Teorema 1.3. Petostruki zbir kvadrata stranica šestougla je veći ili jednak zbiru kvadrata dijagonala istog šestougla.

Dokaz 1.2. Neka su $A_i(x_i, y_i), i = 1, \ldots, 5$, vrhovi šestougla. Pokažimo da je

$$5\sum_{i=0}^{5} \overline{A_iA_{i+1}}^2 \geq \sum_{i=0}^{5} \overline{A_iA_{i+2}}^2 + \overline{A_1A_4}^2 + \overline{A_2A_5}^2 + \overline{A_3A_6}^2.$$

(sabiranje po indeksima ide po modulu 6)

Gornja nejednakost je zbir nejednakosti

$$5\sum_{i=1}^{6}(x_i - x_{i+1})^2 \geq \sum_{i=}^{6}(x_i - x_{i+2})^2 + (x_1 - x_4)^2 + (x_2 - x_5)^2 + (x_3 - x_6)^2 \quad (1)$$

i analogne nejednakosti za $y_1, \ldots, y_6$. Dokažimo nejednakost (1).

$$5\sum_{i=1}^{6}(x_i - x_{i+1})^2 - \sum_{i=1}^{6}(x_i - x_{i+2})^2 - (x_1 - x_4)^2 - (x_2 - x_5)^2 - (x_3 - x_6)^2 =$$

$$= 3(x_1 - x_2 + x_3 - x_4 + x_5 - x_6)^2 + 2(x_1 - x_2 + x_4 - x_5)^2$$
$$+2(x_2 - x_3 + x_5 - x_6)^2 + 2(x_3 - x_4 + x_6 - x_1)^2 \geq 0.$$

Literatura

[1] Radmila Bulajić Manfrino, José Antonio Gómez Ortega, Rogelio Valdez Delgado, *Geometric Inequalities - A Mathematical Olympiad Approach*, 2009. Birkhäuser Basel

[2] Oene Bottema, *Inequalities (Mathematics)*, 1969. Wolters-Noordhoff

TREĆA MATEMATIČKA KONFERENCIJA REPUBLIKE SRPSKE
Trebinje, 7. i 8. juni 2013. godine

Jedna interesantna metoda dokazivanja nejednakosti

Šefket Arslanagić
Univerzitet u Sarajevu, Prirodno-matematički fakultet
asefket@pmf.unsa.ba

Stručni rad

Apstrakt

U ovom radu je data jedna interesantna metoda za dokazivanje algebarskih nejednakosti. Ona se sastoji u sljedećem za slučaj nejednakosti od tri promjenljive $f(x,y,z) \geq 0$; $(x,y,z>0)$: Dokažemo najprije da vrijedi nejednakost

$$f(x,y,z) \geq f(t,t,z),$$

gdje je npr. $t = \frac{x+y}{2}$.
Zatim dokažemo da vrijedi nejednakost

$$f(t,t,z) \geq 0.$$

1 Uvod

Dobro nam je poznata nejednakost između aritmetičke i geometrijske sredine za tri pozitivna broja x, y, z koja glasi:

$$\frac{x+y+z}{3} \geq \sqrt[3]{xyz}, \tag{1}$$

gdje jednakost vrijedi u slučaju kada je $x=y=z$.

U [1] je dato šest raznih dokaza nejednakosti (1). Sada ćemo dati jednu interesantnu metodu pomoću koje ćemo dokazati nejednakost (1). Ova metoda se inače može uspješno koristiti za dokazivanje raznih algebarskih nejednakosti.

Dokaz 1.1. Nejednakost (1) ekvivalentna sa nejednakošću:

$$f(x,y,z) = x+y+z-3\sqrt[3]{xyz} \geq 0;\ \ (x,y,z>0)\,.(1')$$

U prvom koraku ćemo dokazati da vrijedi nejednakost:

$$f(x,y,z) \geq f(t,t,z), \tag{2}$$

gdje je $t = \frac{x+y}{2}$.

Zbog poznate nejednakosti $\frac{x+y}{2} \geq \sqrt{xy}$ slijedi da je $t^2 \geq xy$.

Imamo sada

$$f(x,y,z)-f(t,t,z)=x+y+z-3\sqrt[3]{xyz}-\left(2t+z-3\sqrt[3]{t^2z}\right)=3\left(\sqrt[3]{t^2z}-\sqrt[3]{xyz}\right)\geq 0,$$

a odavde slijedi da je nejednakost (2) tačna.

Dokazaćemo sada da vrijedi nejednakost:

$$f(t,t,z)=2t+z-3\sqrt[3]{t^2z}\geq 0. \tag{3}$$

Imamo

$$f(t,t,z)\geq 0$$

$$\Leftrightarrow 2t+z\geq 3\sqrt[3]{t^2z}$$

$$\Leftrightarrow (2t+z)^3\geq 27t^2z$$

$$\Leftrightarrow 8t^3+12t^2z+6tz^2+z^3-27t^2z\geq 0$$

$$\Leftrightarrow 8t^3-15t^2z+6tz^2+z^3\geq 0$$

$$\Leftrightarrow 8t^3-8t^2z-6t^2z+6tz^2-t^2z+z^3\geq 0$$

$$\Leftrightarrow 8t^2(t-z)-6tz(t-z)-z\left(t^2-z^2\right)\geq 0$$

$$\Leftrightarrow (t-z)\left[8t^2-6tz-z(t+z)\right]\geq 0$$

$$\Leftrightarrow (t-z)\left(8t^2-8tz+tz-z^2\right)\geq 0$$

$$\Leftrightarrow (t-z)\left[8t(t-z)+z(t-z)\right]\geq 0$$

$$\Leftrightarrow (t-z)^2(8t+z)\geq 0\,,$$

što znači da je nejednakost (3) tačna.
Sada iz (2) i (3) dobijamo nejednakost (1'), odnosno nejednakost (1).

Dokaz 1.2. Sada ćemo dokazati da vrijedi nejednakost:

$$f(x,y,z)\geq f(t,t,z)\,, \tag{4}$$

gdje je $t=\sqrt{xy}$.
Zbog poznate nejednakosti $\frac{x+y}{2}\geq\sqrt{xy}$ slijedi da je $2t\leq x+y$, tj. $x+y-2t\geq 0$.
Imamo sada

$$f(x,y,z)-f(t,t,z)=x+y+z-3\sqrt[3]{xyz}-(2t+z-3\sqrt[3]{xyz})=x+y-2t\geq 0,$$

a odavde slijedi da je nejednakost (4) tačna.

Dokazaćemo sada da vrijedi nejednakost:

$$f(t,t,z) = 2t + z - 3\sqrt[3]{t^2 z} \geq 0. \tag{5}$$

Imamo

$$f(t,t,z) \geq 0$$

$$\Leftrightarrow 2t + z \geq 3\sqrt[3]{t^2 z}$$

$$\Leftrightarrow (2t+z)^3 \geq 27t^2 z$$

$$\Leftrightarrow 8t^3 + 12t^2 z + 6tz^2 + z^3 - 27t^2 z \geq 0$$

$$\Leftrightarrow 8t^3 - 15t^2 z + 6tz^2 + z^3 \geq 0$$

(pokazano u Dokazu 1.) $\Leftrightarrow (t-z)^2 (8t+z) \geq 0$,
što znači da je nejednakost (5) tačna.

Iz nejednakosti (4) i (5) slijedi nejednakost (1'), odnosno nejednakost (1).

Sada ćemo koristeći ovu metodu dokazati poznatu **Nesbittovu nejednakost** čijih se deset dokaza nalazi u [2], još jedanaest u [3] i jedan u [4]. Riječ je o sljedećoj nejednakosti:

$$\frac{a}{b+c} + \frac{b}{c+a} + \frac{c}{a+b} \geq \frac{3}{2}; \ \ (a,b,c>0). \tag{6}$$

Dokaz 1.3. U prvom koraku ćemo dokazati nejednakost:

$$f(a,b,c) \geq f(t,t,c), \tag{7}$$

gdje je $f(a,b,c) = \frac{a}{b+c} + \frac{b}{c+a} + \frac{c}{a+b}$ i $t = \frac{a+b}{2}$.

Imamo

$$f(a,b,c) - f\left(\frac{a+b}{2}, \frac{a+b}{2}, c\right) = \frac{a}{b+c} + \frac{b}{c+a} + \frac{c}{a+b} - \left(\frac{a+b}{a+b+2c} + \frac{a+b}{a+b+2c} + \frac{c}{a+b}\right)$$

$$= \frac{a}{b+c} + \frac{b}{c+a} - \frac{2(a+b)}{a+b+2c} = \frac{a^3 + ca^2 + cb^2 + b^3 - 2abc - ab^2 - a^2 b}{(b+c)(c+a)(a+b+2c)}. \tag{8}$$

Na osnovu $AM - GM$ nejednakosti za $n=3$ i $n=2$ imamo:

$$a^3 + ca^2 + cb^2 + b^3 = \frac{a^3 + a^3 + b^3}{3} + \frac{a^3 + b^3 + b^3}{3} + ca^2 + cb^2 \geq a^2 b + ab^2 + 2abc. \tag{9}$$

Sada iz (8) i (9) slijedi:

$f\left(a,b,c\right)-f\left(\frac{a+b}{2},\frac{a+b}{2},c\right)\geq 0$, tj.

$$f\left(a,b,c\right)\geq f\left(\frac{a+b}{2},\frac{a+b}{2},c\right),$$

a ovo je nejednakost (7).

Preostaje nam još da dokažemo nejednakost:

$$f\left(\frac{a+b}{2},\frac{a+b}{2},c\right)=f\left(t,t,c\right)\geq\frac{3}{2}. \tag{10}$$

Imamo

$$f\left(t,t,c\right)\geq\tfrac{3}{2}$$

$$\Leftrightarrow \tfrac{t}{t+c}+\tfrac{t}{t+c}+\tfrac{c}{2t}\geq\tfrac{3}{2}$$

$$\Leftrightarrow \tfrac{2t}{t+c}+\tfrac{c}{2t}-\tfrac{3}{2}\geq 0$$

$$\Leftrightarrow 4t^2+c\left(t+c\right)-3t\left(t+c\right)\geq 0$$

$$\Leftrightarrow t^2-2tc+c^2\geq 0$$

$$\Leftrightarrow \left(t-c\right)^2\geq 0\ ,$$

što je svakako tačno, pa je i nejednakost (10) tačna.

Sada iz nejednakosti (7) i (10) slijedi nejednakost $f\left(a,b,c\right)\geq\frac{3}{2}$, tj. nejednakost (6), q.e.d.

Vrijedi jednakost u (6) ako i samo ako je $a=b=c$.

Sada ćemo koristeći ovu metodu dokazati jednu algebarsku nejednakost sa četiri promjenljive koja glasi:

$$a^4+b^4+c^4+d^4+2abcd\geq a^2b^2+b^2c^2+c^2d^2+d^2a^2+a^2c^2+b^2d^2, \tag{11}$$

gdje su $a,b,c,d\geq 0$.

Dokaz 1.4. Data nejednakost je simetrična pa ne umanjujući općost možemo pretpostaviti da je $a\geq b\geq c\geq d$.

Neka je

$$f\left(a,b,c,d\right)=a^4+b^4+c^4+d^4+2abcd-a^2b^2-b^2c^2-c^2d^2-d^2a^2-a^2c^2-b^2d^2,$$

odnosno

$$f\left(a,b,c,d\right)=a^4+b^4+c^4+d^4+2abcd-a^2c^2-b^2d^2-\left(a^2+c^2\right)\left(b^2+d^2\right).$$

Imamo sada $f\left(a,b,c,d\right)-f\left(\sqrt{ac},b,\sqrt{ac},d\right)=$

$$= a^4 + b^4 + c^4 + d^4 + 2abcd - a^2c^2 - b^2d^2 - \left(a^2 + c^2\right)\left(b^2 + d^2\right) -$$

$$\left[a^2c^2 + b^4 + a^2c^2 + d^4 + 2abcd - a^2c^2 - b^2d^2 - 2ac\left(b^2 + d^2\right)\right] =$$

$$= a^4 + c^4 - 2a^2c^2 - \left(b^2 + d^2\right)\left(a^2 + c^2 - 2ac\right) =$$

$$= \left(a^2 - c^2\right) - \left(b^2 + d^2\right)(a - c)^2 = (a - c)^2\left[(a + c)^2 - \left(b^2 + d^2\right)\right] =$$

$$= (a - c)^2\left[\left(a^2 - b^2\right) + \left(c^2 - d^2\right) + 2ac\right] \geq 0\,,$$

jer je $a \geq b \geq c \geq d$.

Dakle, imamo:

$$f(a, b, c, d) \geq f\left(\sqrt{ac}, b, \sqrt{ac}, d\right). \tag{12}$$

Dokazaćemo sada da je

$$f\left(\sqrt{ac}, b, \sqrt{ac}, d\right) \geq 0, \tag{13}$$

odnosno stavljajući da je $a = b = c = t \geq d$:

$$f(t, t, t, d) \geq 0$$

$$\Leftrightarrow 3t^4 + d^4 + 2t^3d \geq 3t^4 + 3t^2d^2$$

$$\Leftrightarrow d^4 + 2t^3d \geq 3t^2d^2$$

$$\Leftrightarrow \tfrac{d^4 + t^3d + t^3d}{3} \geq \sqrt[3]{d^4 \cdot t^3d \cdot t^3d}\,,$$

a ovo je $AM - GM$ nejednakost za $n = 3$ koja je tačna pa je i nejednakost (13) tačna. Sada iz (12) i (13) slijedi data nejednakost (11).

Vrijedi jednakost u (11) ako i samo ako je $a = b = c = d$ ili $a = b = c,\ d = 0$, te $a = b = d,\ c = 0$, odnosno $a = c = d,\ b = 0$ i $b = c = d,\ a = 0$.

Na kraju ćemo dati dva dokaza jedne nejednakosti i to koristeći prezentiranu metodu a zatim dati dokaz na drugi način ne koristeći ovu metodu.

Riječ je o sljedećoj nejednakosti:

$$a^4 + b^4 + c^4 - 3abc \geq \frac{3}{8}, \tag{14}$$

gdje su a, b, c realni brojevi takvi da je $a + b + c = 0$ i $c \geq 1$.

Dokaz 1.5. Iz datog uslova $a + b + c = 0$ slijedi:

$$a^4 + b^4 + c^4 = 2\left(a^2b^2 + b^2c^2 + a^2c^2\right),$$

pa je sada dovoljno dokazati da vrijedi nejednakost:

$$f\left(a,b,c\right) \geq \frac{3}{8},$$

gdje je $f\left(a,b,c\right)=2\left(a^2b^2+b^2c^2+a^2c^2\right)-3abc$.

Imamo sada

$$f\left(\frac{a+b}{2},\frac{a+b}{2},c\right)=f\left(-\frac{c}{2},-\frac{c}{2},c\right)=2\left(\frac{c^4}{16}+\frac{c^4}{4}+\frac{c^4}{4}\right)-\frac{3c^3}{4}=$$

$$=\frac{c^3}{4}\left(\frac{9c}{2}-3\right)\geq\frac{1}{4}\left(\frac{9}{2}-3\right)=\frac{3}{8} \qquad (15)$$

(zbog $c\geq 1$).

Dalje slijedi

$$f\left(a,b,c\right)-f\left(\tfrac{a+b}{2},\tfrac{a+b}{2},c\right)=$$

$$=2\left[a^2b^2-\left(\tfrac{a+b}{2}\right)^4\right]+2c^2\left[a^2+b^2-2\left(\tfrac{a+b}{2}\right)^2\right]-3c\left[ab-\left(\tfrac{a+b}{2}\right)^2\right]=$$

$$=2\left[ab-\left(\tfrac{a+b}{2}\right)^2\right]\left[ab+\left(\tfrac{a+b}{2}\right)^2\right]+2c^2\cdot\tfrac{2a^2+2b^2-a^2-b^2-2ab}{2}-3c\cdot\tfrac{4ab-a^2-b^2-2ab}{4}=$$

$$=\tfrac{(a-b)^2}{4}\left[-2\left(ab+\left(\tfrac{a+b}{2}\right)^2\right)+4c^2+3c\right]=$$

$$=\tfrac{(a-b)^2}{4}\left(4c^2+3c-2ab-\tfrac{c^2}{2}\right)=\tfrac{(a-b)^2}{4}\left(\tfrac{5}{2}c^2+3c+a^2+b^2\right)\geq 0\ ,$$

odnosno

$$f\left(a,b,c\right)\geq f\left(\frac{a+b}{2},\frac{a+b}{2},c\right). \qquad (16)$$

Dobijamo sada iz (15) i (16):

$f\left(a,b,c\right)\geq f\left(\frac{a+b}{2},\frac{a+b}{2},c\right)\geq\frac{3}{8}$, q.e.d.

Vrijedi jednakost u (14) tj. $f\left(a,b,c\right)=\frac{3}{8}$ ako i samo ako je $a=b=-\frac{1}{2}$ i $c=1$.

Dokaz 1.6. Iz uslova $a+b+c=0$ slijedi $a+b=-c\leq -1$ (zbog uslova $c\geq 1$).

Razlikovaćemo dva slučaja:

1^0 Jedan od brojeva a ili b je nenegativan. Ne umanjujući opštost pretpostavićemo da je $a\geq 0$. Tada je $b\leq 0$ i imamo sada da je $-abc\geq 0$. Slijedi sada

$$a^4+b^4+c^4-3abc\geq c^4\geq 1>\frac{3}{8},$$

što znači da je data nejednakost (14) tačna.

2^0 Oba broja a i b su nepozitivna, tj. $a, b \leq 0$. Tada je $-a = m > 0$ i $-b = n > 0$, tj. $m + n = c \geq 1$ te je sada potrebno dokazati da vrijedi nejednakost:

$$m^4 + n^4 + c^4 - 3mnc \geq \frac{3}{8}. \tag{17}$$

Imamo sada

$$m^4 + n^4 + c^4 - 3mnc = m^4 + n^4 + (m+n)^4 - 3mn\,(m+n) =$$

$$= m^4 + n^4 + m^4 + 4m^3n + 6m^2n^2 + 4mn^3 + n^4 - 3m^2n - 3mn^2 =$$

$$= 2m^4 + 2n^4 + m^3n + mn^3 + \left(3m^3n + 3m^2n^2 + 3m^2n^2 + 3mn^3\right) - \left(3m^2n + 3mn^2\right) =$$

$$= 2m^4 + 2n^4 + m^3n + mn^3 + (m+n)\left(3m^2n + 3mn^2\right) - \left(3m^2n + 3mn^2\right) =$$

$$= 2m^4 + 2n^4 + m^3n + mn^3 + (m+n-1)\left(3m^2n + 3mn^2\right) \geq$$

$$\geq 2m^4 + 2n^4 + m^3n + mn^3 = m^4 + n^4 + m^3\,(m+n) + n^3\,(m+n) =$$

$$= \left(m^4 + n^4\right) + (m+n)\left(m^3 + n^3\right) \geq \tfrac{1}{8} + 1 \cdot \tfrac{2}{8} = \tfrac{3}{8}\ ,$$

jer je $m + n = c \geq 1$ i također:
$\sqrt[4]{\frac{m^4+n^4}{2}} \geq \frac{m+n}{2} \geq \frac{1}{2}$ i $\sqrt[3]{\frac{m^3+n^3}{2}} \geq \frac{m+n}{2} \geq \frac{1}{2}$,
te odavde
$m^4 + n^4 \geq \frac{1}{8}$ i $m^3 + n^3 \geq \frac{2}{8}$.
Ovim je dokazana nejednakost (17), odnosno data nejednakost (14).

Vrijedi jednakost u (17) i (14) ako i samo ako je $m + n - 1 = 0$ i $m = n$, tj. $m = n = \frac{1}{2}$ te $a = b = -\frac{1}{2}$ i $c = 1$.

Svakako je Dokaz 1. kraći i elegantniji od Dokaza 2., što ide u prilog tvrdnji da je izložena metoda dokazivanja nejednakosti veoma efikasna i značajna.

Dobro bi bilo da budući čitaoci ovog članka pokušaju primjeniti ovu metodu kod dokazivanja nekih algebarskih nejednakosti sa tri, četiri i više promjenljivih.

Literatura

[1] Š. Arslanagić, Metodička zbirka zadataka sa osnovama teorije iz elementarne matematike, Grafičar promet d.o.o., Sarajevo, 2006.

[2] Š. Arslanagić, Matematička čitanka 1, Grafičar promet, d.o.o., Sarajevo, 2009.

[3] Š. Arslanagić, Matematička čitanka 2, Grafičar promet, d.o.o., Sarajevo, 2010.

[4] Š. Arslanagić, Matematička čitanka 5, Grafičar promet, d.o.o., Sarajevo, 2013.

[5] Z. Cvetkovski, Inequalities – Theorems, Techniques and Selected Problems, Springer-Verlag Berlin/Heidelberg, 2012.

[6] Mitrinović, D. S., Pečarić, J. E., Fink, A. M., Classical and New Inequalities in Analysis, Kluwer Academic Publishers, Dordrecht/Boston/London, 1993.

TREĆA MATEMATIČKA KONFERENCIJA REPUBLIKE SRPSKE
Trebinje, 7. i 8. juni 2013. godine

Logička formulacija elementarne teorije brojeva (FETB)

Radoslav Milošević
Filozofski fakultet, Pale
radoslav_milosevic@yahoo.com

Stručni rad

Apstrakt

Ovdje je riječ o logičkoj formalizaciji elementarne teorije brojeva u "stilu" izgradnje logike iskaza i logike predikata, kao primjer formalizacije neke konkretne matematičke teorije. Navešćemo osnovne definicije, terme (izraze), formule, aksiome i teoreme tako formalizovane elementarne teorije brojeva. Između ostalog, na kraju ćemo istaći konzistentnost, potpunost, odlučivost, te aksiomu izbora i hipotezu kontinuuma logički formalizovane elementarne teorije brojeva.

1 Uvod

U današnje vrijeme u Teoriji brojeva koriste se elementarne i analitičke metode za rješavanje problema prostih brojeva u raznim numeričkim nizovima. Klasično rasuđujući, (intuitivna, tj. neformalizovana) elementarna teorija brojeva jeste kategorička teorija. I većina "modernih" matematičara intimno je uvjerena da je ona i konzistentna. (Veliki broj matematičara vjeruje takođe da je svaki individualni smisleni problem što se unutar elementarne teorije brojeva može formulisati – npr. Fermaova (Fermatova) teorema – u načelu rješiv, bez obzira na to je li rješiv ili će biti rješiv ili će eventualno ostati nerješen). Prema intuicionistima za objašnjenje nekih pojmova teorije skupova koji su nepotpuno formirani u matematici dovode do glomaznih i zamašnih konstrukcija. Intuicionisti negiraju, posebno, zakon isključenja trećeg za beskonačne skupove. Intuicionizam dovodi do poricanja mnogih dostignuća matematičke analize. Intuicionisti su postigli neke rezultate, naročito u konstruktivnoj logici.

Za ilustraciju kako se "u stilu" izgradnje logike iskaza i logike predikata formalizuje neka konkretna matematička teorija skiciraćemo početke izgradnje *formalizovane elementarne teorije brojeva* (ubuduće kraće FETB). Pri tome ćemo i nulu "0" smatrati prirodnim brojem, jer je ovdje takva konvencija korisna.

2 Jedna izagrdnja formalne elementarne teorije brojeva

Jedna od niza mogućnosti takve izgradnje može se započeti ovako:

Definicija 2.1. *Slova FETB* jesu:

a) *Konstant*e (nul-mjesnih) predikata FETB: T, $\perp$

b) (Individualne) *promjenljive* ili *varijable* FETB: *a,b,c,...* (pretpostavimo da ih ima potencijalno prebrojivo beskonačno mnogo).

v) *Operatori* FETB: , $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$, $\forall$, $\exists$.

g) *Dvomjesni predikat jednakosti* FETB: =.

d) Simboli *funkcija* FETB:

- + (intuitivno mu odgovara zbir)
- Δ (intuitivno mu odgovara proizvod)
- ′ (intuitivno mu odgovara sljedbenik, tj. za 1 veći broj, npr. ′0=1, 0′′=2 itd),

e) *Individualni* simbol FETB: 0 (intuitivno mu odgovara broj nula).

i) *Zagrade* FETB: (,).

Konvencijama za izostavljanje zagrada dodajemo da "=" veže jače od operatora, a jednakost vezanja simbola funkcija onda pripada nizu ′,B·,+.

Definicija 2.2. *Riječi* FETB su konačni (neprazni) nizovi slova FETB.

Za razliku od ranije skiciranih teorija logike iskaza i logike predikata ovdje nam pored pojma *formule* treba još i pojam *terma*. Intuitivno termu – grubo rečeno-uopšte odgovara neki brojevni izraz; formuli uopšte odgovara neka relacija među takvim izrazima.

Terme (izrazi) FETB definišu se rekurzivno. To su opet određene "istaknute" riječi FETB.

Terme FETB definišemo ovako:

a) 0 je terma FETB.

b) Varijabla FETB je terma FETB.

v) Ako je t terma FETB, (t)′ je terma FETB.

g), d) Ako su u, v terme FETB, onda su i $(u)+(v)$, (u)B·(v) terme FETB.

e) Termi FETB su samo one riječi FETB koje se mogu dobiti (eventualno višestruko ponovnjenom) konstrukcijama a) do e).

Formule FETB se definišu rekurzivno. I to su određene "istaknute" riječi FETB.

Formule FETB definišemo ovako:

a) Konstante $\perp$ su formule FETB.

b) Ako su u, v terme FETB, $(u) = (v)$ je formula FETB.

v) Ako je A formula FETB, (A) je formula FETB.

g) do g) Ako su A , B formule FETB, onda su i (A) $\wedge$ (B), (A) $\vee$ (B), (A) $\Rightarrow$ (B) , (A)$\Leftrightarrow$ (B) formule FETB.

d), e)) Ako je u varijabla FETB i A formula FETB, onda su ($\forall$ u) (A), ($\exists$) (A) formule FETB.

i) Formule FETB su samo one riječi FETB koje se mogu dobiti (eventualno višestruko ponovljenim) konstrukcijama a) do i).

Opet se može dokazati da je za bilo koju riječ FETB u konačno mnogo koraka moguća odluka o tome je li to terma ili formula FETB ili nije; tekođe, ako jeste, u konačno mnogo koraka moguća je (jednoznačna) rekonstrukcija izgradnje te terme ili formule prema III a) do e) odnosno IV a) do i), opet npr. u obliku "stabla".

Slobodne i vezane varijable formula FETB definišu se analogno kao u logici predikata. Neka varijabla na nekom mjestu u formuli A FETB slobodna je onda i samo onda ako tamo nije u dosegu nekog kvantora $\forall$ ili $\exists$ koji veže istu varijablu; inače je vezana.

Formula koja ne sadrži nijedne slobodne varijable zove se *slobodna.*

Pojam "slobodan za ... u formuli" modifikuje se ovako:

Definicija 2.3. Terma je u FETB *slobodan* za varijablu v u formuli A FETB ako v u A ni u kojem mjestu ne dolazi u dosegu nekog kvantora $\forall$ ili $\exists$ koji veže jednu od varijabli koje dolaze u u.

Oznaka A $(u|v)$ sada znači formulu FETB koja nastaje iz formule A FETB kada se u njoj varijabla v na svim mjestima gdje dolazi slobodna zamijeni teoremom u. A (x) znači formulu koja na nekim mjestima (možda) sadrži slobodnu varijablu x a A (t) znači formulu koju dobijamo ako x na svim mjestima zamijenimo termom t.

Aksiomi FETB definišu se ovako: Neka su A, B, C bilo koje formule FETB. U g) i h) neka je u terma slobodan za varijablu v u formuli A. Tada su ove formule aksiomi FETB:

a) do j) kao u logici predikata

k) $a{=}b{\Rightarrow}((a{=}c){\Rightarrow}(b{=}c))$;

l) $a{=}b{\Rightarrow}\ a\prime\ {=}b\prime$;

m) $\neg a'\ {=}0$;

n) $a\prime\ {=}b\prime\ \Rightarrow\ a{=}\ b$;

o) $a{+}0{=}a$;

p) $a\ +\ b'\ =\ (a\ +\ b)'$

r) $aB{\cdot}0{=}0$;

s) $aB{\cdot}b'{=}aB{\cdot}\ b{+}\ a$;

t) $\mathrm{A}\ (0)\ \wedge(\forall x)\ (\mathrm{A}(x)\ \Rightarrow\ \mathrm{A}\ (x')\ \Rightarrow\ \mathrm{A}(x)$.

Teoreme FETB definišu se kao teoreme logike predikata, s tim što svugdje "logika predikata" treba zamijeniti sa "FETB".

Ako je A teorema FETB, pišemo opet $-$ A .

Kako je direktno izvođenje teorema FETB prema VIII redovno dosta tegobno ili čak efektivno jedva provedeno (npr. već dokaz $-\ a{=}a$ traži bar 17 ne uvijek posve jednostavnih koraka), i tu se za razvij teorije služimo metapravilima dedukcije koja, *mutatis mutandis*, glase kao i u logici predikata.

Tako postavljena FETB razrađuje se dalje putevima više ili manje sličnim, paralelni ili srodnim onima pri izgradnji odgovarajuće neformalizovane teorije. Pri tom su, potrebne brojne dodatne definicije kojima se razni pojmovi elementarne teorije brojeva svode na one koji su implicitno sadržani u aksiomama FETB; pa to, nije uvijek posve jednostavno, već često zahtijeva dosta vještine i, dosljedno i potpuno sprovođeno, strpljenja i upornosti: Najpre se – relativno lako – dokazuju opšta osobine zbira i proizvoda kao što su npr. komutativnost, asocijativnost i distrubutivnost te postojanje jedinstvenosti neutralnog elementa za te operacije. Zatim se uvodi relacija uređenja, tako da $a\leq b$ po definiciji znači formulu $(\exists c)$ $(a{+}c{=}b)$ i dokazuje da su prirodni brojevi tom relacijom *potpuno* uređeni, tj. da tako definisana relacija ima osobine refleksivnosti, antisimetije i tranzitivnosti i da za svaki par prirodnih brajeva a,b vrijedi bar jedna od relacija $a\leq b,\ b\leq a,$ te da je skup prirodnih brojeva relacijom $\leq$ i dobro uređen (tj. svaki njegov neprazni podskup ima najmanji element). Teorema o postojanju beskonačno mnogo primbrojeva dokazuje se u (ekvivalentnom) obliku da za svaki prirodni broj postoji od njega veći koji je predbroj; itd. itd.U detalje svega toga ovdje ne ulazimo, već ćemo navesti neke od važnih matematičkih rezultata ispitivanja takve (i srodnih) FETB.

3 Konzistentnost, potpunost i odlučivost fetb

Što se tiče pitanja u vezi sa adekvatnošću, konzistencijom, potpunošću i kategoričnošću FETB, može se dokazati da nastupaju okolnosti bitno drugačije od onih kod logike iskaza i predikata.

No FETB ne "pokriva" elementarnu teoriju brojeva onako kao što logika iskaza odnosno predikata "pokriva" algebru iskaza odnosno predikata. Ili, ako se s time "ne želimo pomiriti", moramo radikalno revidirati klasična shvaćanja o tome *šta je uopšte* (intuitivna) elementarna teorija brojeva.

Uopšte, ovdje ćemo (u poglavlju o FETB) pod *konsistentnošću* (neprotivrječnošću) *formalnog* sistema smatrati konsistentnost u Hilbertovu smislu, tj. FETB ili neki srodni sistem je *konsistentan* ako ni za jednu njegovu formulu A nije jednako – A i – ¬A . Pored pojma konsistentnosti uvodi se u FETB i pojam ω-konsistentnosti: FETB ili neki srodni sistem je ω- *konsistentan* ako ni za jednu varijablu x i formulu A *(x)* nije jednako :

– A (0), – A (1), – A (2), ... , – A (n), ... ; –¬($\forall x$) A (x),

gdje je 1 oznaka za (0)' , 2 za ((0)')' itd.).U protivnom slučaju, ako FETB nije ω- konsistentna, ona je ω- *nekonsistentna* ili ω-*protivrječna.* Lako se uviđa da ω- konsistentnost FETB (ili srodnog sistema) povlači njenu konsistentnost, tj. ako je FETB ω- konsistentna, ona je *a fortiori* konsistentna.

Slično ćemo u FETB pod *potpunošću* smatrati potpunost u Hilbertovu smislu, tj. FETB ili neki srodni sistem je *potpun* ako je za *svaku zatvorenu* formulu A vrijedi bilo − A , bilo −¬A, (ako je FETB i konsistentna, sigurno neće vrijediti oboje). U protivnom slučaju, ako postoji neka zatvorena formula A za koju niti je – A , niti je –¬A , FETB je *nepotpuna.*Uvodi se pojam ω- nepotpunost: FETB ili neki srodni sistem je ω-*nepotpun* ako postoji neka varijabla x i neka formula A (x), tako da je

– A (0), – A (1), – A (2), ... , – A (n), ... ;

ali da *nije* – ($\forall x$) A (x).

Činjenica da sistem možda može biti ω- nepotpun pokazuje da on možda može biti konsistentan, a da nije i ω- *konsistentan.* (Uostalom, postoje još oštrije definicije konsistentnosti i potpunosti). Takav konsistentin, ali ne i ω-konsistentan sistem zove se *izvana* nekonsistentan, jer nije kompatibilan sa svojom intendiranom interpretacijom; naime čak i neki dokaz njegove konsistentnosti ne bi garantovao da u njemu ne može biti dokazano nešto što je intuitivno neispravno.

Fundamentalni rezultati K. Godela koji se odnose na pitanje konsistentnosti i potpunosti FETB bili su možda i ne toliko definitivan udarac Hilbertovu programu "refundiranja" matematike njenom formalizacijom uz finitnu matematiku (iako mogu sugerisati reviziju toga programa) koliko su za svoje vrijeme (prije nešto više od četiri decenije) bili neočekivani, može se reći senzacionalni. U svakom slučaju može se smatrati da su oni otvorili (ili bar bitno produbili) novu "krizu" matematike i njenih osnova, ali da su ih jednako i revolucionirali isto tako duboko kao što je ranije Kopernik revolucionirao astronomiju, Darvin biologiju i Einštejn fiziku.

Za FETB (ne samo kako je ovdje skicirana već i za sve srodne formalizacije "sličnog tipa" - detaljnije ga ovdje ne možemo opisivati) može se među ostalim pokazati (dokazi su, dakako, vrlo složeni):

Svaka FETB (takva određenog tipa) nužno je *bilo nekonsistentnost, bilo nepotpuna*, tj. *ili* je u njoj uopšte svaka formula teoreme (i time čitava teorija bezvrijedna), *ili* postoje (i mogu se efikasno navesti) zatvorene formule (koje intuitivno odgovaraju određenim slovima) A , takve da se može dokazati da ne može biti niti – A , niti – A ; drugim riječima, ako je FETB konsistentna, u njoj nužno postoje tzv. *neodlučive* tvrdnje. Nadalje, "unutar" same FETB, ako ona jeste konsistentna, *ne može* se dokazati da je konsistentna. Takođe, ako je FETB konsistentna, može se efikasno konstruisati takva zatvorena formula FETB A da :

1. A bude *neodlučivo* u FETB,
2. A bude "*neistinito* u intendiranoj interpretaciji" FETB
3. FETB proširena sa A kao dodatnim aksiomom bude (*ostane*) konsistentna, ali da sigurno *nije* ω*-konsistentna.*

G. Gentzen je pokazao da se uz odstupanje od Hilbertova zahtjeva striktne finitnosti matematike može "dokazati" konsistentnost (određenih) FETB; međutim, ona se time dokazuje samo relativno. Naime, svodi se na (striktno finitno "neosiguranu") konsistentnost tzv. transfinitne indukcije (do određenog rednog broja) u metateoriji. U kojoj je mjeri opravdano i u kom se smislu iz svega toga može zaključiti na *načelu* nemogućnosti "adekvatne" formalizacije "intuitivne" elementarne teorije brojeva *uopšte* uz "striktno finitnu" matematiku ili nekim drugim "isto tako sigurno legitimnim" načinom, stvar je shvaćanja i diskusije; mišljenja se o tome razilaze, a konačne odluke za sad još nipošto nema. (Uprkos pojedinim kategoričkim i emfatičkim suprotnim izjavama nekih matematičara i filozofa). Kao svugdje, i oddje treba biti krajnje oprezan i suzdržljiv u iznošenju konačnih iskaza i prognoza – ako već ni zbog čega drugog, a ono na osnovi iskustva što su se već mnoge takve ocjene najvećih matematičara i filozofa pokazale neispravnim.

4 O aksiomi izbora i hipotezi kontinuuma FETB

U *formalnoj teoriji skupova* okolnosti su, od formulacije do rezultata, znatno komplikovanije. Jedan od najvažnijih rezultata je da su u nekim takvim formalizacijama određenog tipa (što opet, na žalost, ovdje ne možemo tačno precizirati), ako one uopšte jesu konzistentne, *aksiom izbora i hipoteza kontinuuma* neodlučive tvrdnje, tj. ne mogu se u njima ni dokazati ni pobiti (tj. dokazati njihova negacija). (Intuitivno, aksiom izbora tvrdi da za svaku familiju disjuktnih nepraznih skupova postoji skup njihovih reprezentanata koji iz svakog člana familije sadrži tačno po jedan element. Hipoteza kontinuuma može se intuitivno formulisati tako da za svaki beskonačni podskup skupa svih realnih brojeva postoji *bilo* uzajamno jednoznačno preslikavanje tog podskupa na skup svih racionalnih brojeva, *bilo* uzajamno jednoznačno preslikavanje tog podskupa na skup svih realnih brojeva). Neočekivan je bio i raniji rezultat T. Skolema da formalizovana teorija skupova (određenog tipa), ako uopšte ima model, nužno ima i samo prebrojivo beskonačan model – pa čak i da je njegov "jednostavniji" rezultat o tzv. nestandardnim modelima Peanove aksiomatike za prirodne brojeve. Tačnije značenje toga, i uopšte makar i grubu skicu tzv. *teorije modela*, koja je tekođe od izuzetne važnosti za *neka shvaćanja* matematičke logike, ovdje ne možemo dati jer bi predstavljalo relativno opsežnije znanje iz niza grana "klasične" (tj. neformalizovane) moderne matematike (npr. teorije skupova, apstraktne algebre, topologije, teorije kategorija itd.).

Godelovi, Kohenovi i Skolemovi rezultati (i niz daljih rezultata velikih matematičara - logičara) bacili su novo svjetlo i na *neformalizovanu*, "klasičnu" ili "naivnu" teoriju skupova.O tome što oni za nju znače mišljenja i opet nisu jedinstvena, čak uvijek ni među matematičarima-logičarima "istog smjera". U vrlo duboka i teška pitanja, s tim u vezi, za koja bi se moglo reći da spadaju u "filozofiju" matematike – razumije se da ovdje ne možemo ulaziti.

Literatura

[1] A. Marcja, C. Toffalori, A guide to classical and modern model theory, Kluwer Academic Publishers, 2003.

[2] A. S. Troelstra, H. Schwichtenberg, Basic proof theory, Cambridge University Press, 2000.

[3] B. Poizat, A course in model theory, Springer-Verlag, 2000.

[4] Vladimir Devide, Matematička logika, Matematički institut, Beograd, 1964

[5] L. A. Kalužnin, Što je matematička logika, Školska knjiga, Zagreb. 1975

[6] P. Papić, Uvod u teoriju skupova, HMD, Zagreb, 2000.

[7] P. Blackburn, M. de Rijke, Y. Venema, Modal logic, Springer-Verlag, 2001.

[8] R. Cori, D. Lascar, Mathematical Logic I, II, Oxford University Press, 2000.

[9] S. G. Simpson, Mathematical logic, skripta, The Pennsylvania State University, 2001. htt://www.math.psu.edu/simpson/

[10] Slaviša B.Prešić, Elementi matematičke logike, 3avod za izdavanje udžbenika SR Srbije, Beograd, 1968

TREĆA MATEMATIČKA KONFERENCIJA REPUBLIKE SRPSKE
Trebinje, 7. i 8. juni 2013. godine

Relacije i matematičke operacije na jednoj klasi Pitagorinih trouglova

Milan Živanović
Visoka škola strukovnih studija za obrazovanje vaspitača, Kruševac, Srbija,
mzivanovic@vaspks.edu.rs

Stručni rad

Apstrakt

U radu će biti predstavljena nova (prirodna) parametrizacija problema Pitagorinih trojki. Na osnovu te parametrizacije biće definisane relacije poretka i ekvivalencije na celom skupu Pitagorinih trojki. Takođe će biti predstavljena bijekcija izmedju skupa $\mathbb{N}_0$ i jedne klase Pitagorinih trojki, što će omogućiti uvođenje aritmetičkih operacija na toj klasi Pitagorinih trojki.

Ključne reči: Pitagorine trojke, parametrizacija Pitagorinih trojki, relacija poretka, relacija ekvivalencije, aritmetičke operacije

1 Pitagorine trojke

Pod Pitagorinom trojkom podrazumevamo trojku (a, b, c) prirodnih brojeva koja zadovoljava Pitagorinu jednačinu

$$x^2 + y^2 = z^2. \tag{1}$$

Ako su brojevi a, b, c uzajamno prosti kazhemo da je trojka *osnovna* (*primitivna*), a u suprotnom je *izvedena*. Takođe, pravougle trouglove sa cjelobrojnim stranicama nazivamo *Pitagorinim trouglovima*. Problem nalazhenja svih takvih trojki (trouglova) bio je privlačan matematičarima kroz razne epohe, tako da postoji više njegovih rešenja, tzv. parametrizacija. Najstarije i ujedno najpoznatije je rešenje koje potiče još od Euklida. Navodimo bez dokaza savremenu formulaciju ovog tvrđenja.

Teorema 1.1. Neka su m i n uzajamno prosti prirodni brojevi različite parnosti i $m > n$. Tada su formulom

$$a = m^2 - n^2, \quad b = 2mn, \quad c = m^2 + n^2$$

predstavljena sva osnovna rešenja jednačine (1), odnosno, svi osnovni Pitagorini trouglovi kojima je prva kateta neparna, a druga parna*.

*Ubuduće ćemo u tekstu pod osnovnim Pitagorinim trouglovima podrazumevati takve trouglove, bez naglašavanja da je prva kateta neparna, a druga parna.

Sam Pitagora je pre ovog rešenja koristio jednoparametarsku parametrizaciju za klasu pravouglih trouglova kod kojih je hipotenuza za 1 veća od veće (parne) katete. Ako je n neparna kateta, onda je parna kateta bila $\frac{n^2-1}{2}$ a hipotenuza $\frac{n^2+1}{2}$.

Iz elementarne geometrije nam je poznato da se pravougli trougao projekcijama upisane kruzhnice na njegove stranice mozhe podeliti na duži

$$a = 2r + n, \quad b = 2r + p, \quad c = 2r + n + p. \tag{2}$$

Zamenom transformacije (2) u jednačinu (1) dobija se Diofantova jednačina

$$2r^2 = np.$$

Ako ovde umesto prirodnog broja r ubacimo njegovu kanonsku faktorizaciju $r = 2^\alpha \prod_{i=1}^m p_i^{\alpha_i}$, gde je $\alpha \in \mathbb{N} \cup \{0\}$, $\alpha_i, m \in \mathbb{N}$, i p_i su različiti neparni prosti brojevi, dobijamo da je

$$np = 2^{2\alpha+1} \prod_{i=1}^m p_i^{2\alpha_i}.$$

Da bi odgovarajuća trojka brojeva (a, b, c) bila osnovna moraju n i p biti uzajamno prosti. Ako je pri tome p parno, tada skup svih rešenja za n možemo predstaviti formulama

$$n = \prod_{j\in J} p_j^{2\alpha_j},$$

gde je J proizvoljan podskup skupa svih indeksa $I = \{1, 2, \dots, m\}$ ili $n = 1$ ako je $J = \emptyset$. Tada je

$$p = 2^{2\alpha+1} \prod_{k\in I\setminus J} p_k^{2\alpha_k}.$$

Na taj način dobijamo da se stranice Pitagorinog trougla opisanog oko kruga poluprečnika r mogu dobiti pomoću sledećih formula

$$a = 2^{\alpha+1} \prod_{i\in I} p_i^{\alpha_i} + \prod_{j\in J} p_j^{2\alpha_j}, \tag{3}$$

$$b = 2^{\alpha+1} \prod_{i\in I} p_i^{\alpha_i} + 2^{2\alpha+1} \prod_{k\in I\setminus J} p_k^{2\alpha_k}, \tag{4}$$

$$c = 2^{\alpha+1} \prod_{i\in I} p_i^{\alpha_i} + \prod_{j\in J} p_j^{2\alpha_j} + 2^{2\alpha+1} \prod_{k\in I\setminus J} p_k^{2\alpha_k}. \tag{5}$$

Drugim rečima,

Teorema 1.2. Oko kruga poluprečnika r, $r \in \mathbb{N}$, koji ima tačno m prostih delilaca različitih od 2, može se opisati tačno 2^m osnovnih nepodudarnih trouglova, čije su stranice zadate formulama $(3) - (5)$.

Dogovorimo se da Pitagorinu trojku (a, b, c) možemo pisati i kao trojku $\langle r, n, p\rangle$, tj. $(a, b, c) = \langle r, n, p\rangle$.

Primer 1.1. Odrediti sve osnovne Pitagorine trouglove kojima je poluprečnik upisane kružnice $r = 2310$.

Rešenje. Najpre ćemo napisati kanonsku faktorizaciju broja $r = 2310 = 2\cdot 3\cdot 5\cdot 7\cdot 11$. Prema teoremi 1.1 postoji tačno $2^4 = 16$ osnovnih Pitagorinih trouglova opisanih oko kruga datog poluprečnika. U sledećoj tabeli su odgovarajuće trojke uredjene po rastućem parametru n.

r	n	p	$\langle r,n,p\rangle$	(a,b,c)
2310	1	10672200	$\langle 2310, 1, 10672200\rangle$	$(4621, 10676820, 10676821)$
2310	9	1185800	$\langle 2310, 9, 1185800\rangle$	$(4629, 1190420, 1190429)$
2310	25	426888	$\langle 2310, 25, 426888\rangle$	$(4645, 431508, 431533)$
2310	49	217800	$\langle 2310, 49, 217800\rangle$	$(4669, 222420, 222469)$
2310	121	88200	$\langle 2310, 121, 88200\rangle$	$(4741, 92820, 92941)$
2310	225	47432	$\langle 2310, 225, 47432\rangle$	$(4845, 52052, 52277)$
2310	441	24200	$\langle 2310, 441, 24200\rangle$	$(5061, 28820, 29261)$
2310	1089	9800	$\langle 2310, 1089, 9800\rangle$	$(5709, 14420, 15509)$
2310	1225	8712	$\langle 2310, 1225, 8712\rangle$	$(5845, 13332, 14557)$
2310	3025	3528	$\langle 2310, 3025, 3528\rangle$	$(7645, 8148, 11173)$
2310	5929	1800	$\langle 2310, 5929, 1800\rangle$	$(10549, 6420, 12349)$
2310	11025	968	$\langle 2310, 11025, 968\rangle$	$(15645, 5588, 16613)$
2310	27225	392	$\langle 2310, 27225, 392\rangle$	$(31845, 5012, 32237)$
2310	53361	200	$\langle 2310, 53361, 200\rangle$	$(57981, 4820, 58181)$
2310	148225	72	$\langle 2310, 148225, 72\rangle$	$(152845, 4692, 152917)$
2310	1334025	8	$\langle 2310, 1334025, 8\rangle$	$(1338645, 4628, 1338653)$

Tabela 1:

2 Relacije i matematičke operacije na jednoj klasi Pitagorinih trouglova

Rešenje jednačine (1) opisano u teoremi 2.1 omogućava i uvodjenje relacija poretka i relacija ekvivalencije na skupu svih osnovnih Pitagorinih trojki kod kojih je drugi član paran. Taj skup, proširen članom $(1, 0, 1)$, obeležimo sa $\overline{\prod}$.

Definicija 2.1. Neka su date dve Pitagorine trojke $\pi_1 = \langle r_1, n_1, p_1\rangle$ i $\pi_2 = \langle r_2, n_2, p_2\rangle$ kod kojih su n_1 i n_2 parni brojevi. Kažemo da je $\pi_1 \geq \pi_2$ ako i samo ako je $r_1 > r_2$ ili $r_1 = r_2$ i $n_1 \geq n_2$.

Teorema 2.1. Relacija $\geq$ na skupu $\overline{\prod}$ je relacija linearnog poretka.
Dokaz. Neka je $\pi = \langle r, n, p\rangle$ proizvoljan element skupa $\overline{\prod}$. Kako je $r \geq r$ i $n \geq n$, sledi da je $\pi \geq \pi$, pa je relacija refleksivna.

Za proizvoljne dve trojke π_1 i π_2, iz $r_1 \geq r_2$, $n_1 \geq n_2$ i $r_2 \geq r_1$, $n_2 \geq n_1$ sledi da je $r_1 = r_2$ i $n_1 = n_2$. Dakle, relacija je antisimetrična.

Analogno se tranzitivnost relacije $\geq$ za Pitagorine trojke izvodi iz tranzitivnosti relacije $\geq$ za prirodne brojeve. Relacijom $\geq$ su uporedive svake dve osnovne Pitagorine trojke, pa je ova relacija na skupu $\overline{\prod}$ relacija linearnog poretka.

Definicija 2.2. Neka su date dve Pitagorine trojke π_1 i π_2 na skupu $\overline{\prod}$. Kažemo da je $\pi_1 \rho_i \pi_2$, $i \in \{r, n, p\}$ ako i samo ako je $i_1 = i_2$ $(r_1 = r_2, n_1 = n_2, p_1 = p_2)$.

Može se dokazati da je svaka od relacija ρ_i, $i \in \{r, n, p\}$, relacija ekvivalencije. U geometrijskoj interpretaciji klase ekvivalencije u odnosu na relaciju ρ_n su klase Pitagorinih trouglova sa zadatom razlikom izmedju hipotenuze i parne katete, klase relacije ρ_p su klase Pitagorinih trouglova zadate razlike izmedju hipotenuze i neparne katete, a klase relacije ρ_r su klase Pitagorinih trouglova sa zadatim prečnikom upisane kružnice.

Primer 2.1. Količnički skup skupa $\overline{\prod}$ u odnosu na relaciju ρ_p je skup $\overline{\prod}/\rho_p = \{\overline{\prod}_{p=2n^2} \wedge n \in \mathbb{N}_0\}$.

Tako je $\overline{\prod}_{p=2} = \{(4k^2 - 1, 4k, 4k^2 + 1) \wedge k \in \mathbb{N}\}$ prebrojiva klasa osnovnih Pitagorinih trouglova sa razlikom izmedju hipotenuze i jedne katete jednakom 2.

Klasa $\overline{\prod}_{r=15} = \{(31, 480, 481), (39, 80, 89), (55, 48, 73), (255, 32, 257)\}$ je konačna i predstavlja sve osnovne nepodudarne Pitagorine trouglove opisane oko kružnice polupečnika 15.

U nekim ovako definisanim klasama moguće je uvesti osnovne aritmetičke operacije. Tako su u radu [3] definisane operacije sabiranja i oduzimanja Pitagorinih trojki iz klase $\overline{\prod}_{n=1}$, a u radu [4] su uvedene operacije NZS i NZD Pitagorinih trojki iste klase.

Podsetimo se operacija sabiranja i množenja u datoj klasi.

Definicija 2.3. Neka je $\overline{\prod}_{n=1} = \{\langle r, 1, 2r^2\rangle \wedge r \in \mathbb{N}_0\}$ klasa osnovnih Pitagorinih trouglova kod kojih je hipotenuza za 1 veća od katete, proširena trojkom $(1, 0, 1)$,i π_1, $\pi_2 \in \overline{\prod}_{n=1}$. Tada je

$$\pi_1 + \pi_2 = \langle r_1 + r_2, 1, 2(r_1 + r_2)^2\rangle =$$

$$= (2r_1 + 2r_2 + 1, 2(r_1 + r_2)(1 + r_1 + r_2), 2(r_1 + r_2)(1 + r_1 + r_2) + 1)$$

i

$$\pi_1 \cdot \pi_2 = \langle 2r_1r_2, 1, 2r_1^2r_2^2\rangle = (1 + 2r_1r_2, 2r_1r_2(1 + r_1r_2), 2r_1r_2(1 + r_1r_2) + 1).$$

Primer 2.2. $(5, 12, 13)+(9, 40, 41) = \langle 2, 1, 8\rangle+\langle 4, 1, 32\rangle = \langle 6, 1, 72\rangle = (13, 84, 85)$,

$$(5, 12, 13) \cdot (9, 40, 41) = \langle 2, 1, 8\rangle \cdot \langle 4, 1, 32\rangle = \langle 8, 1, 128\rangle = (17, 144, 145).$$

Po analogiji sa stepenovanjem u skupu prirodnih brojeva definiše se pojam stepena na skupu $\overline{\prod}_{n=1}$.

Primer 2.3. $(5, 12, 13)^3 = (17, 140, 141)$, $(3, 4, 5)^n = (3, 4, 5)$. Uopšte za $\pi = \langle r, 1, 2r^2 \rangle \in \overline{\prod}_{n=1}$ je $\pi^n = \langle r, 1, 2r^2 \rangle^n = \langle r^n, 1, 2r^{2n} \rangle$.

Po analogiji sa operacijom deljenja u skupu prirodnih brojeva možemo tu operaciju uvesti i na klasu $\overline{\prod}_{n=1}$. Dakle, ako je $\pi_1 \cdot \pi_2 = \pi_3$ u $\overline{\prod}_{n=1}$, kažemo da je $\pi_3 : \pi_1 = \pi_2$, $\pi_3 : \pi_2 = \pi_1$. čak šta više, ako za poluprečnike upisanih krugova u Pitagorine trouglove važi $r_3 = r_1 r_2 + o$, $0 \leq o < r_2$, tada je i

$$\langle r_3, 1, 2r_3^2 \rangle = \langle r_1, 1, 2r_1^2 \rangle \cdot \langle r_2, 1, 2r_2^2 \rangle + \langle o, 1, 2o^2 \rangle,$$

odnosno, u posmatranoj klasi postoji deljenje sa ostatkom.

Primer 2.4. Lako izračunavamo da je $\langle 9, 1, 162 \rangle = \langle 4, 1, 32 \rangle \cdot \langle 2, 1, 8 \rangle + \langle 1, 1, 2 \rangle$, odnosno $(19, 180, 181) = (9, 40, 41) \cdot (5, 12, 13) + (3, 4, 5)$.

Teorema 2.2. Preslikavanje $F : \mathbb{N}_0 \longrightarrow \overline{\prod}_{n=1}$ definisano sa

$$F(r) = (2r + 1, 2r^2 + 2r, 2r^2 + 2r + 1), \quad r \in \mathbb{N}_0,$$

je bijekcija i saglasno je sa operacijama sabiranja i množenja u ova dva skupa.

Dokaz. Ako su r_1, r_2 različiti prirodni brojevi, tada su i trojke brojeva $(2r_1 + 1, 2r_1^2 + 2r_1, 2r_1^2 + 2r_1 + 1)$ i $(2r_2 + 1, 2r_2^2 + 2r_2, 2r_2^2 + 2r_2 + 1)$ takodje različite, pa je preslikavanje „*1-1*". Takodje je poluprečnik svakog Pitagorinog trougla prirodan broj pa je preslikavanje "*na*". Iz definicije sabiranja i množenja trojki u klasi $\overline{\prod}_{n=1}$ neposredno sledi saglasnost sa odgovarajućim operacijama u skupu $\mathbb{N}_0$.

Posledica 2.2.1. Sve osobine skupa prirodnih brojeva, bijekcijom uvedenom u teoremi 2.2, mogu se preneti na skup osnovnih Pitagorinih trouglova kojima je hipotenuza za 1 veća od parne katete.

Literatura

[1] Vidan Govedarica, *Gornja granica broja predstavljanja prirodnog broja u obliku zbira dva kvadrata*, Radovi Filozofskog fakulteta, Broj 9, Knjiga 2, 127–132, Pale, 2007.

[2] Milan Živanović, *Algebra Pitagorinih trojki*, Spomenica akademika Veselina Perića, ANURS, Banja Luka, (2011) 525–535.

[3] Milan Živanović, *Jedna klasa Pitagorinih trojki*, Zbornik radova sa druge matematičke konferencije Republike Srpske, Trebinje, 2012, 145–148.

[4] Milan Živanović, Milenko Pikula, *Least common multiple and greatest common divisor of Pythagora's triplets*, Science And Higher Education In Function Of Sustainable Development - SED 2012, Užice, 2012.

[5] Neville Robbins, *On the number of primitive Pythagorean triangles with a given inradius*, Fibonacci quarterly, 44/4 (2006), 368-369.

[6] Wacław Sierpinski, *Pythagorean triangles*, Dover Publications, 2003.

TREĆA MATEMATIČKA KONFERENCIJA REPUBLIKE SRPSKE
Trebinje, 7. i 8. juni 2013. godine

Vjerojatnost kockareve propasti u igri ruleta

Ivan Budimir
Sveučilište u Zagrebu, Grafički fakultet
ibudimir@grf.hr

Stručni rad

Apstrakt

Slučajnost je obilježje velike većine pojava u svijetu u kojem živimo. Matematička disciplina koja se bavi slučajnim pojavama je teorija vjerojatnosti. Stoga se primjene teorije vjerojatnosti mogu pronaći u gotovo svim aspektima "realnog života". Jedan od ciljeva nastave vjerojatnosti treba biti osposobljavanje polaznika za prepoznavanje realnih problema, te njihovo vjerojatnosno modeliranje. Nastavno gradivo iz vjerojatnosti treba biti obogaćeno što većim brojem primjera koji su razumljivi svim polaznicima. Danas su veoma aktualne igre na sreću poput ruleta. Veliki broj vjerojatnosnih principa može se demonstrirati na primjerima ove svima poznate igre na sreću. Osim toga, vjerojatnost je kao znanstvena disciplina nastala iz proučavanja kockarskih igara. U radu je naveden primjer kockareve propasti kroz koji je moguće objasniti formulu potpune vjerojatnosti.

1 Kockarev kraj

U radu je opisan jedan problem povezan s igrom ruleta. Izračunata je vjerojatnost kockarevog gubitka u jednoj seriji igara ruleta, koju je kockar odlučio igrati u skladu s u nastavku opisanom strategijom. Rulet ima 36 brojeva i nulu. Kockar je odlučio igrati samo na serije crnog. Vjerojatnost dobitka jednog žetona u jednoj igri je $\frac{18}{37}$, a gubitka $1-\frac{18}{37}=\frac{19}{37}$. Kockar raspolaže s 50 žetona, a kocka dok ne dođe do 100 žetona ili dok ne izgubi sve. U nastavku je određena vjerojatnost da kockar izgubi sve.
Neka kockar u nekom trenutku raspolaže sa k žetona. Označimo događaj

$$A=\{kockar\ koji\ raspolaže\ s\ k\ žetona\ je\ na\ kraju\ pogubio\ sve\ novce\}.$$

Vjerojatnost da kockar koji je došao do razine od k žetona izgubi igru označimo s:

$$P(A)=p(k)=p_k$$

Početni uvjeti su:

$$p_0=1 \text{ i } p_{100}=0$$

U slijedećem koraku moguća su ostvarenja samo dvije mogućnosti (hipoteze):

H_1={*kockar je izgubio jedan žeton*} ili H_2={*kockar je dobio jedan žeton*}.

Prema formuli koja vrijedi za potpun sistem događaja možemo pisati:

$$P(A) = P(H_1) \cdot P(A|H_1) + P(H_2) \cdot P(A|H_2)$$

Primijetimo da je u skladu s našim oznakama

$$p_{k-1} = P(A|H_1)$$

i

$$p_{k+1} = P(A|H_2)$$

Iz prethodne formule dobije se rekurzija oblika

$$p_k = \frac{19}{37} \cdot p_{k-1} + \frac{18}{37} \cdot p_{k+1}$$

Preoblikujemo li prethodnu rekurziju dobijemo

$$18 \cdot p_{k+1} - 37 \cdot p_k + 19 \cdot p_{k-1} = 0$$

Karakteristična jednadžba glasi

$$18 \cdot \lambda^2 - 37 \cdot \lambda + 19 = 0$$

Rješenja su

$$\lambda_1 = 1, \lambda_2 = \frac{19}{18}$$

Tražena vjerojatnost u općem slučaju ima oblik

$$p_k = a \cdot \lambda_1^k + b \cdot \lambda_2^k$$

tj. u našem slučaju

$$p_k = a \cdot 1^k + b \cdot \left(\frac{19}{18}\right)^k$$

odnosno, ljepše zapisano

$$p_k = a + b \cdot \left(\frac{19}{18}\right)^k$$

Uvažimo li početne uvjete $p_0 = 1$ i $p_{100} = 0$ dobijemo parametre a i b

$$a = \frac{\left(\frac{19}{18}\right)^{100}}{\left(\frac{19}{18}\right)^{100} - 1}$$

$$b = -\frac{1}{\left(\frac{19}{18}\right)^{100} - 1}$$

Dakle

$$p_k = \frac{\left(\frac{19}{18}\right)^{100}}{\left(\frac{19}{18}\right)^{100} - 1} - \frac{1}{\left(\frac{19}{18}\right)^{100} - 1} \cdot \left(\frac{19}{18}\right)^k$$

ili

$$p_k = \frac{\left(\frac{19}{18}\right)^{100} - \left(\frac{19}{18}\right)^k}{\left(\frac{19}{18}\right)^{100} - 1}$$

Sada uvrstimo $k = 50$ pa dobijemo

$$p_{50} = 0.937 = 93.7\%$$

Ovaj problem predstavlja idealizaciju strategije igrača koja se javlja u "realnom svijetu". Vidimo da je vjerojatnost kockareve propasti veoma velika, odnosno vjerojatnost da će kockar izaći kao pobjednik je veoma mala. Rezultat je takav upravo zbog postojanja zelenog polja ili nule. Upravo zbog postojanja nule u igri ruleta, u svakoj pojedinoj igri ruleta, uz bilo kakvu strategiju igrača, uvijek je šansa da igrač dobije manja od šanse da izgubi. Isti zaključak vrijedi i za serije igara.

2 Zaključak

U nastavu vjerojatnosti i statistike potrebno je implementirati što veći broj primjera iz konteksta realnog života. Zadaci koji se koriste u nastavi trebaju biti povezani s polaznikovim iskustvom. Stoga je prikladno gradivo vjerojatnosti i statistike povezivati sa svim aspektima gospodarskih, socijalnih i sportskih zbivanja. Nastavni proces koji bi uključivao primjere slične opisanom u ovom radu bio bi kvalitetniji i pristupačniji polaznicima.

Literatura

[1] Nikola Sarapa, Teorija vjerojatnosti, 3. prerađeno izdanje, Školska knjiga, Zagreb, 2002.

TREĆA MATEMATIČKA KONFERENCIJA REPUBLIKE SRPSKE
Trebinje, 7. i 8. juni 2013. godine

Servisno orijentisana arhitektura informacionog sistema integrisanog univerziteta

Vladan Mastilović
Ministarstvo Odbrane, Sarajevo, Bosna i Hercegovina
vladan.mastilovic@gmail.com
Zorana Banjanin
FFTS Beograd
zoranabanjanin@gmail.com

Stručni rad

Apstrakt

U naslovu ovog rada figurišu dva osnovna koncepta koji definišu problemsko polje i predmet istraživanja. To su servisno orijentisana arhitektura (SOA), i informacioni sistem za integrisanje funkcija univerziteta. Globalna informaciono-komunikaciona infrastruktura je osnova za trenutnu integraciju ekonomija, kultura i društava širom svijeta. Ogromne količine pisanih dokumenata ali i multimedijalnih aplikacija, cirkulišu između fakulteta, isnstituta, laboratorija, katedri, studentskih tijela i drugih organizacionih entiteta univerziteta. Po paradigmi toka i paradigmi poruke u računarskoj komunikaciji digitalni dokumenti generisani u jednoj interaguju sa entitetima u drugoj oreganizacionoj jedinici pa je kreiranje dokumenata u papirnoj formi, zatim slanje poštom ili faksom i na kraju ponovan unos u računar očigledno stvaranje nepotrebnih utrošaka vremena, energije, novca, mentalnih i fizičkih napora.

Ovo istraživanje je ciljno orijentisano na dizajniranje i razvoj informacionog sistema sa web servisima za unificiran pristup bazama podataka, razmjenu i dijeljenje podataka u komunikaciji između organizacionih jedinica integrisanog univerziteta. Razvijeno rešenje treba da omogući brže, konzistentnije, pouzdanije i kvalitetnije pružanje usluga zaposlenima, studentima i ostalim stejkholderima univerziteta.

1 Uvod

U naslovu ovog rada figurišu dva osnovna koncepta koji definišu problemsko polje i predmet istraživanja. To su servisno orijentisana arhitektura (SOA), i informacioni sistem za integrisanje funkcija univerziteta. Globalna informaciono-komunikaciona infrastruktura je osnova za trenutnu integraciju ekonomija, kultura i društava širom svijeta. Ogromne količine pisanih dokumenata ali i multimedijalnih aplikacija, cirkulišu između fakulteta, isnstituta, laboratorija, katedri, studentskih tijela i drugih organizacionih entiteta univerziteta. Po paradigmi toka i paradigmi poruke u računarskoj komunikaciji digitalni dokumenti generisani u jednoj interaguju sa entitetima u drugoj oreganizacionoj jedinici pa je kreiranje dokumenata u papirnoj formi, zatim slanje poštom ili faksom i na kraju ponovan unos u računar očigledno stvaranje nepotrebnih utrošaka vremena, energije, novca, mentalnih i fizičkih napora.

Prema [1], SOA je stil softverske arhitekture za dizajniranje i razvoj softvera u formi više interoprabilnih servisa. Ti servisi imaju dobro definisane poslovne funkcionalnosti koje su napravljene kao softverske komponente koje kasnije mogu biti višestruko korištene za različite svrhe. Ova je nova arhitektura za razvoj slabo povezanih distribuiranih komponenti.

Prema [2] *WWW, World Wide Web, W3,* ili jednostavno *Web* je sistem međusobno povezanih, hipertekstovanih dokumenata, koji se nalaze na Internetu. Pojava *web-a,* kao jednog od servisa Interneta, promenila je način na koji pristupamo informacijama. One su nam sada dostupne 24 sata dnevno, bez obzira na lokaciju na kojoj se fizički nalazimo.

U poslednje vreme, veoma značajnu ulogu u razmeni podataka i informacija putem *web-a* zauzimaju *web* servisi. *Web* servisi predstavljaju izvor informacija na *web*-u, koji omogućuje korisnicima da pristupe različitim resursima, na način na koji to njima najviše odgovara. Postoji veliki broj definicija *web* servisa, ali sve one određuju *web* servise [3] kao deo poslovne logike koji se nalazi negde na *web-u* i kome korisnici mogu da pristupaju putem standardnih Internet protokola kao što su HTTP ili SMTP. Prema [4] web servisi:

1. imaju **osnovni cilj,** da omoguće povezivanje poslovanja odnosno, omogućiti programsko povezivanje distribuiranih softverskih komponenti bez obzira na kojoj su platformi realizovani, koji je programski jezik je tom prilikom korišten, kao i platforma na kojoj se izvršavaju,

2. treba da ispune **dva osnovna zahteva**, da imaju mogućnost ponovne upotrebe (*reusability*) i da omogućavaju interoperabilnost (*interoperability*) i

3. imaju **viziju,** postojanje miliona nezavinih komponenti dostupnih preko Interneta koje su upotrebljive na bilo kojoj platformi i svim razvojnim jezicima.

Ovo istraživanje je ciljno orijentisano na dizajniranje i razvoj informacionog sistema sa web servisima za unificiran pristup bazama podataka, razmjenu i dijeljenje podataka u komunikaciji između organizacionih jedinica integrisanog univerziteta. Razvijeno rešenje treba da omogući brže, konzistentnije, pouzdanije i kvalitetnije pružanje usluga zaposlenima, studentima i ostalim stejkholderima univerziteta. Sve ustanove mogu da pristupe podacima o studentima koji su u sastavu bilo koje ustanove univerziteta koja je obuhvaćena predloženim informacionim sistemom.

Rad je koncipiran tako da je u drugom odeljku opisan problem koji treba rešiti, postojeća rešenja i navedene su oblasti u kojima je moguće primeniti *Web* servise. Treći odeljak opisuje predloženo rešenje. U četvrtom su opisane primenjene tehnologije. Logički model i realizacija predloženog rešenja prikazana je u petom odeljku. U šestom odeljku date su mogućnosti daljeg usavršavanja, a sedmi odeljak predstavlja zaključak.

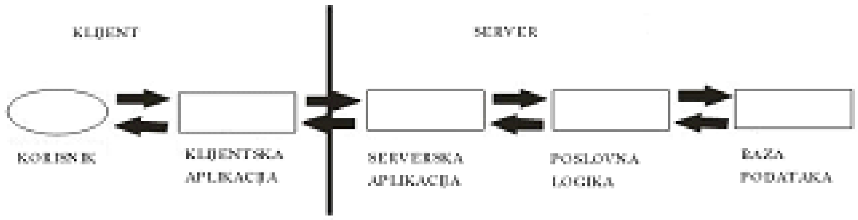
2 Opis problema i postojeća rešenja

Filozofski fakultet, Pale kao visokoškolska ustanova egzistira u sastavu Univerziteta u Istočnom Sarajevu. Fokus je stavljen na sagledavanje načina i vrsta komunikacija i razmene podataka i informacija između različitih celina Univerziteta u Istočnom Sarajevu. Uočen je niz problema koji su u eri rapidnog napretka informatike i telekomunikacija prosto nezamislivi. Upravo te probleme treba prevazići prelaskom na savremeni informacioni sistem za integrisanje funkcija univerziteta, koji će biti razvijan zasnovano na servisno orjentisanoj arhitekturi, a uz primenu savremenih informacionih tehnologija.

U okviru Univerziteta u Istočnom Sarajevu egzistira više nezavisnih celina, fakulteta i ustanova. Svaka od tih celina vodi elektronsku evidenciju o svojim studentima, koja ni na koji način nije povezana sa evidencijom koju vodi neka druga celina o svojim studentima. Svaki student poseduje identifikacionu karticu, koju mu izdaje nadležna ustanova ili fakultet, što znači da su podaci koji se vode o studentu dostupni samo jednoj celini univerziteta. Da bi te kartice mogle da se koriste u sistemu čitavog univerziteta, neophodno je da postoji mehanizam koji će svim zasebnim celinama obezbeđivati podatke o studentima i karticama koje su izdate u drugim celinama. Taj informacioni sistem je razvijen u okviru ovog rada i nazvan je Centralni registar studenata (*u daljem tekstu CRS*).

Fakulteti i druge ustanove univerziteta vode podatke o studentima u svojim relacionim bazama podataka. Relacione baze podataka omogućuju osoblju jedne ustanove znatno efikasniji uvid u sve potrebne informacije o studentima, lakšu i bržu pretragu po raznim parametrima. Prednosti su brojne, a sve vode pouzdanijem i efikasnijem pružanju usluga. Međutim, nekada ovi sistemi ne mogu da ispune sve zahteve ili se ranije definisani zahtevi proširuju. Možda se zahteva brža i pouzdanija razmena i dijeljenje podataka između različitih ustanova univerziteta. Formati baza podataka, odnosno sami modeli relacionih baza podataka se razlikuju od ustanove do ustanove, što znatno otežava komunikaciju između dve ili više ustanova. Sve to otežava razmenu i dijeljenje podataka između različitih ustanova.

Razmatranjem navedenih nedostataka uočeni su određeni problemi koje treba da prevaziđe informacioni sistem (*CRS*) razvijen u okviru ovog rada. Pri tom se pošlo od pretpostavke da u ciljnim zasebnim celinama Univerziteta u Istočnom Sarajevu postoje relacione baze podataka u kojima se vode podaci o studentima. Odgovarajući podaci o svim studentima univerziteta će biti objedinjeni u bazi



Slika 1: Opšta struktura aplikacije

CRS-a.

Svaka ustanova, kao što je navedeno, vodi podatke o svojim studentima. To znači da bilo ko od ovlašćenog osoblja te zasebne celine ima detaljan uvid u podatke o studentima te ustanove. Problem nastaje kada se student koji je pripadnik neke ustanove obrati za uslugu nekoj drugoj ustanovi univerziteta. U matičnoj ustanovi postoje svi podaci o njemu, ali tim podacima ne može da pristupi osoblje neke druge celine univerziteta, kojoj se student obratio. Drugi problem koji se javlja, jeste problem ažurnosti informacija. Usled čestih unosa, izmena i brisanja podataka u svakoj bazi zasebnih celina univerziteta dolazi do neažurnosti informacija. Kao posledica ovoga javlja se pojava da ostale celine univerziteta imaju podatke o studentu koji nisu validni.

2.1 Predloženo rešenje

Rešenje razvijeno u ovom radu omogucuje razmenu i dijeljenje podataka o svim studentima univerziteta, između zasebnih celina univerziteta i Centralnog registra studenata. Ovaj informacioni sistem treba da omogući i obrnutu komunikaciju, razmenu i dijeljenje podataka o studentima između CRS-a i svih ustanova univerziteta.

Na osnovu razmatranja postavljenih zahteva i distribuirane prirode problema, zakljuceno je da je potrebno da se aplikacija sastoji iz nekoliko slojeva. Medutim, sve komponente predloženog rešenja mogu se podeliti u dve grupe, odnosno, na komponente koje cine klijentsku stranu i komponente koje cine serversku stranu, kao što je prikazano na slici 1.

Klijent/server arhitektura može imati nekoliko oblika, ali najcešce se u implementacijama srecu dve vrste. To su, aplikacije sa вЂћtankim klijentom i debelim serverom" i aplikacije sa "debelim klijentom i tankim serverom" [3].

Prvi tip klijent/server arhitekture podrazumeva da se sva poslovna (*procesna*) logika nalazi na serverskoj strani aplikacije. Prilikom komunikacije izmedu klijenta i servera, klijent prosleduje serveru zahteve u odredenom formatu. Serverska strana prihvata zahteve i vrši njihovu obradu. Nakon završene obrade, server prosleduje klijentu samo rezultate obrade. Klijent sada ne mora vršiti nikakvu dodatnu obradu dobijenih podataka, vec ih samo prikazuje korisniku.

Drugi tip klijent/server arhitekture ima nešto drugaciju logiku procesiranja zahteva. Naime, kod ovog tipa klijent prosleduje zahtev serveru, ali server ne vrši skoro nikakvu obradu, vec sve podatke vraca klijentu koji vrši dalje procesirane dobijenih podataka. A kako su klijent i server najcešce dva razlicita racunara povezana mrežom, to znaci da se zahteva znatno veci protok podataka kroz mrežu.

U problemu koji je predstavljen u dosadašnjem tekstu, pomenuto je da se obrada vrši nad podacima koji se nalaze u razlicitim bazama podataka, kao i da se te baze podataka nalaze u razlicitim ustanovama. Cest je slucaj da izmedu tih ustanova ne postoji racunarska mreža, pa je stoga odluceno da rešenje treba zasnovati na *Web* tehnologijama. Naravno, sama *web* aplikacija u suštini i jeste klijent/server softverski sistem. Prema [5] postoje razlicite arhitekture *web* aplikacija:

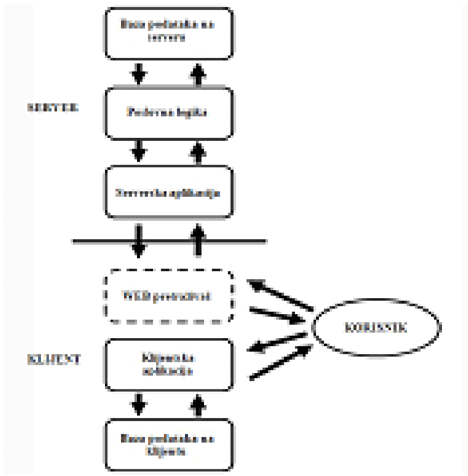
1. *Tanak web klijent* (engl. *Thin web client*) – koja se najcešce koristi za aplikacije zasnovane na Internetu. U njima postoji slaba kontrola nad konfiguracijom klijenta, a od njega se zahteva da poseduje samo *web* citac. Sva poslovna logika se odvija na serveru. Primeticuje se da je ovaj tip arhitekture *Web* aplikacija isti kao arhitektura "tankog klijenta i debelog servera".

2. ***Debeo web klijent*** (engl. *Thick web client*) – veci deo logike sistema se izvršava na klijentskoj strani. Klijent koristi dinamicki *HTML*, *Java* aplete ili *ActiveX* kontrole da bi izvršio logiku sistema. Komunikacija sa serverom se i ovde obavlja korišcenjem *HTTP* protokola.

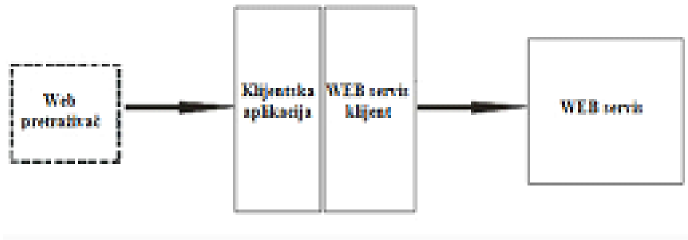
3. ***Web dostava*** (engl. *web delivery*) – pored *HTTP* protokola, za komunikaciju sa serverom koriste se i drugi protokoli, kao što su *IIOP* (engl. *Internet InterOrb Protocol*) i *DCOM* (engl. *Distributed Component Object Model*). *Web* citac se, pre svega, ponaša kao uredaj koji služi za dostavljanje i cuvanje distribuiranog objektnog modela.

U predloženom rešenju zahteva se prenos informacija kroz mrežu, od klijenta do servera i obratno. Takođe je potrebna izvesna obrada podataka na klijentu pre slanja ka Centralnom registru studenata. Iz prethodnog opisa se zakljucuje da je arhitektura koja najviše odgovara ovom slucaju, *web* dostava. Medutim, predloženo rešenje je ipak u većoj meri specificno, u smislu da predstavlja višeslojnu arhitekturu. Detaljna arhitektura predloženog rešenja je predstavljena na slici 2.

Klijentu je potrebna *desktop* aplikacija koja će izvršiti potrebnu obradu podataka, odnosno eksportovanje podataka u određeni format koji je podesan za slanje serveru. Klijentska aplikacija na zahtev korisnika pristupa lokalnoj bazi podataka i eksportuje sve promene (*nove zapise i stare koji su promenjeni*) nakon poslednjeg ažuriranja u *xml* fajl. Ovo se obezbeđuje nekom vrstom *snapshot*-a, odnosno korišćenjem posebnog *xml* fajla u kome se čuva stanje čitave baze podataka nakon poslednjeg eksportovanja. Kada se naredni put javi zahtev za ažuriranjem onda se u tom trenutku pravi slika baze i takođe čuva u *xml* fajlu. Sada se taj fajl poredi sa predhodnim stanjem baze, čime dobijamo sve zapise



Slika 2: Detaljan prikaz arhitekture predloženog rešenja



Slika 3: Komunikacija izmedu klijenta i *web* servisa

koji su izmenjeni, dodati ili obrisani, i od njih formiramo novi *xml* fajl koji ćemo slati serveru. Kao što se vidi na slici 2 na serveru postoji serverska aplikacija koja ovaj fajl prihvata i na osnovu njega ažurira podatke u centralnoj bazi podataka. Takođe, klijent mora imati *Web* pretraživač kojim pristupa serverskoj, odnosno *Web* aplikaciji. Komunikacija od servera ka klijentu se vrši na identičan način sa zamenjenim ulogama. Naime, na zahtev klijenta, server šalje promene od poslednjeg ažuriranja klijentu, koji ih prihvata i ažurira svoje podatke. Što znaci da se ne vrši sva obrada na serveru, već i na klijentu. Zbog toga je ranije navedeno da, generalno, aplikacija pripada arhitekturi *web* dostave. Razlika je u tome što klijent komunicira sa serverom posredstvom *web* servisa, kao što je prikazano na slici 3.

Web servis predstavlja još jedan sloj poslovne logike koji prihvata klijentske zahteve i prosleduje ih dalje do servera na kome se vrši obrada podataka. Pritom treba primetiti, da se sve komponente predloženog rešenja mogu nalaziti na razlicitim lokacijama. Ta cinjenica je dosta važna jer omogucuje visoku modularnost i pouzdanost sistema. Medutim, sa druge strane, komunikacija izmedu pojedinih komponenti se obavlja putem racunarske mreže. U jednom ovakvom sistemu brzina odziva mora da bude što veca. To znaci da kroz mrežu treba slati što je moguce manje podataka.

U skladu sa navedenim bilo je potrebno odabrati i tehnologije koje to omogucuju. Naravno, postoji više tehnologija koje omogucuju rad sa bazama podataka i svaka od njih ima svoje prednosti i nedostatke. Medutim, ideja je bila da se razvije distribuirana poslovna aplikacija koja se sastoji iz više komponenti, odnosno da ostvari odredenu skalabilnost i da omogucuje istovremeni pristup više korisnika. Ove zahteve u potpunosti ispunjava *Enterprise Java Beans* (*EJB*) arhitektura [6, 7]. Pored toga, ove aplikacije pišu se jednom, a zatim se mogu postaviti na bilo koju serversku platformu koja podržava *Enterprise Java Beans* specifikaciju.

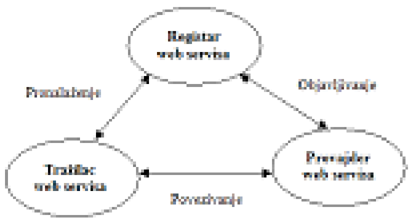
Ovde treba napomenuti da postoje još dve srodne arhitekture, *Hibernate* [8, 9] i *Spring* [10], koje poseduju skoro sve osobine *EJB* arhitekture. Ipak, one su izvedene iz *EJB* arhitekture, kako bi olakšale odredene segmente izrade aplikacija, medutim, *EJB* arhitektura se pokazala kao znatno pouzdanija [6, 7]. Zbog lakšeg razvoja aplikacije za pristup lokalnim bazama podataka odlučio sam se za *Hibernate,* a zbog pouzdanosti za pristup i rad sa centralnom bazom odlučio sam se za *EJB.*

Na kraju je trebalo izabrati i tehnologiju kojom ce biti razvijana klijentska aplikacija. Potreba je bila tehnologija koja razdvaja procesiranje od prezentacije. Kao logican izbor nametnula se tehnologija *Java Server Pages* (*JSP*) [11]. Prema [11, 12], *JSP* predstavlja kljucni element *Java 2 Enterprise Edition* (*J2EE*) platforme i može na najbolji nacin da iskoristi prednosti mnogih *Java Enterprise* biblioteka, kao što su *JDBC*, *JNDI* i *EJB*. A to je upravo ono što je bilo potrebno u izradi aplikacije koju prati ovaj seminarski rad.

3 Opis primenjenih tehnologija

3.1 Web servisi

Web servisi se prema [3], definišu kao deo poslovne logike kaja se nalazi negde na *web*-u i kome korisnici mogu da pristupe u svakom trenutku, putem standardnih Internet protokola kao što su HTTP ili SMTP. Oni predstavljaju novu eru u razvoju distribuiranih informacionih sistema, kao i dobar mehanizam za izgradnju sistema koji se baziraju na servisno orijentisanoj arhitekturi. *Web* servisi nude rešenje za dva velika problema koja su postojala pre njihove pojave. Prvi problem je da se funkcionalnosti jedne aplikacije učine dostupnim spoljnom svetu. Spoljni svet u ovom slučaju čine veliki broj korisničkih računara koji rade pod različitim platformama. Ovo omogućava korisnicima da za izgradnju svojih aplikacija koriste gotove komponente koje su im dostupne preko Interneta. Na taj način se omogućava izgradnja velikih distribuiranih sistema. Drugi problem je komunikacija i razmena podataka između različitih aplikacija kao i između različitih platformi.



Slika 4: Prikaz interakcije osnovnih komponenata *web* servisa [4]

Da bi se ovo omogućilo bilo je potrebno da se servisi koji se žele učiniti dostupni putem Interneta na neki način objave i označe, kako bi ih udaljeni korisnici mogli pronaći i identifikovati. Da bi se moglo opisati korišćenje servisa od strane udaljenih korisnika moraju se prvo definisati osnovne komponenete *web* servisa kao što su provajder *web* servisa, klijent *web* servisa i registar *web* servisa.

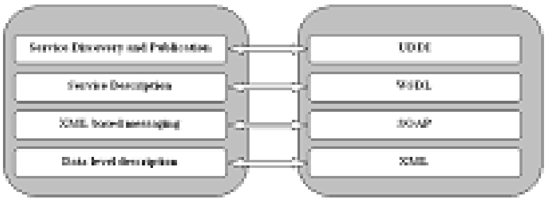
Provajder, pružalac *web* servisa obezbeđuje servise sa određenom funkcionalnošću. Provajder mora da poseduje na sebi aplikacioni server koji podržava tehnologije *web* servisa. Klijent, tražilac *web* servisa koristi udaljene servise koje nude provajderi. Registar *web* servisa je mesto gde se nalazi registar svih servisa koje nude svi provajderi. Na slici 4 prikazan je način na koji ove tri komponente međusobno interaguju.

Ceo proces funkcionisanja *web* servisa počinje tako što provajder servisa objavi servis koji želi učiniti dostupan putem *web*-a. Opis servisa, kao i način na koji se ovi servisi identifikuju tada se upisuju u registar *web* servisa. Kada klijent-tražilac želi da koristi određeni servis on traži servis na osnovu njegovog opisa u registru. Kada ga pronađe onda iz registra dobija podatak kako da dati servis identifikuje na *web*-u. Na osnovu tog podatka vrši se povezivanje između korisnika i provajdera i omogućuje se korisniku da koristi servise koje je zahtevao. Komuniciranje između klijenta i provajdera može se odvijati sinhrono i asinhrono. Sinhrona komunikacija podrazumeva da je korisnik blokiran sve dok se ne završi izvršavanje servisa. Asinhrona komunikacija omogućava klijentu da pozove servis i nastavi sa izvršavanjem drugih operacija dok čeka rezultat.

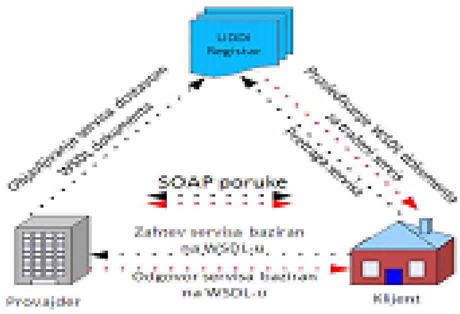
Da bi se ovo postiglo koriste se sljedeći standardi i protokoli: *WSDL (Web Service Description Language), UDDI (Universal Description, Discovery, and Integration)*, *SOAP (Simple Object Access Protocol)* i *XML (eXtensible Markup Language).* Ove tehnologije i njihova namena prikazani su na slici 5.

3.1.1 WSDL (Web Service Description Language)

WSDL (Web Service Description Language) je *XML* tehnologija koja opisuje interfejs *web* servisa na standardizovan način. *WSDL* standardizuje način na koji *web* servisi predstavljaju ulazne i izlazne parametre, strukturu funkcija, način pozivanja funkcija i mapiranje protokola servisa. To omogućava različitim klijentima da automatski razumeju kako treba da komuniciraju sa *web* servisom.



Slika 5: Prikaz standarda *web* servisa [4]



Slika 6: Jednostavna interakcija sa *web* servisom

3.1.2 UDDI (Universal Description, Discovery, and Integration)

UDDI (Universal Description, Discovery, and Integration) obezbeđuje registar za *web* servise, svetskih razmera, zbog objavljivanja, otkrivanja i integracije. Poslovni analitičari koriste *UDDI* kako bi pronašli raspoložive *web* servise traženjem po imenima, identifikatorima, kategorijama specifikacija koje implementira *web* servis. *UDDI* predstavlja strukturu za prikazivanje *web* servisa kao i pristupnih tačaka za njih.

3.1.3 SOAP (Simple Object Access Protocol)

SOAP (Simple Object Access Protocol) predstavlja standard za razmenu struktuiranih informacija u *web* okruženju. Kao takav, predstavlja glavni način komunikacije između glavnih učesnika *SOA*: provajder servisa, registar servisa i korisnik servisa. Glavni ciljevi njegovog dizajna su jednostavnost i proširivost. Na slici 6 prikazana je jednostavna interakcija sa *web* servisom korišćenjem opisanih protokola i standarda.

Provajder objavljuje svoj servis slanjem *WSDL* dokumenta *UDDI* registru. U *UDDI* registru se smešta dati *WSDL* dokument i dodeljuje mu se jedinstvena identifikacija po kojoj se može pronaći u registru. Kada klijent traži neki servis on mora proslediti parametre za identifikaciju opisa servisa *UDDI* registru. Ukoliko pronađe opis servisa registar šalje klijentu opis datog servisa u formi *WSDL* dokumenta. Na osnovu njega klijent sada zna kako da uspostavi vezu sa provajderom servisa i šalje mu zahtev za korišćenjem servisa i čeka odgovor. Ukoliko mu provajder odgovori potrvrdno započinje komunikacija između njih razmenjivanjem *SOAP* poruka.

3.2 XML (eXtensible Markup Language)

XML (eXtensible Markup Language) [13] je markerski jezik koji se koristi za opisivanje strukture podataka. Njegove mogućnosti mogu da dođu do izražaja svuda gde se obavljaju operacije ulaza i izlaza, memorisanja ili prenosa podataka sa jednog mesta na drugo.

Za razliku od *HTML*-a *XML* nema strogo definisane tagove, već nazive tagova definiše sam korisnik. Zato se i zove univerzalni proširivi jezik za označavanje.

Nastao je iz potrebe razmene podataka između aplikacija koje koriste različitu strukturu podataka ili rade na različitim platformama. Zbog toga je nezamenjiv kao sredstvo za opisivanje struktura podataka koje se razmenjuju na *web*-u.

XML opisuje klasu objekata podataka pod imenom *XML* dokumenti i delimično opisuje ponašanje računarskih programa koji ih obrađuju. *XML* dokumenti se sastoje od skladišnih jedinica koje se nazivaju entiteti koji sadrže parsirane ili neparsirane podatke. Parsirani podaci se sastoje od znakova od kojih su neki znakovni podaci, a neki oznake. Oznake opisuju formu smeštanja podataka u *XML* dokument i logičku strukturu dokumenta. Pored *XML*-a postoje druge tehnologije koje proširuju mogućnosti obrade i manipulacije *XML* dokumentima. To su *XSLT* (*Extensible Style sheet Language for Transformation*), *XPath*, *DTD* (*Document Type Definition*), *XML Shema*, *XML* parseri.

3.2.1 XML parseri

XML parseri [13] su alati koji služe za dobijanje informacija sadržanih u nekom *xml* dokumentu. Oni imaju mogućnost da prepoznaju tagove u okviru dokumenta i da se kreću kroz dokument i pozicioniraju na određeno mesto u dokumentu. Postoje dva osnovna tipa parsera koji se često koriste. To su *JAXP (JAVA API for XML)* i *SAX* (*Simple API for XML*) i *DOM* (*Document Object Model*) [14] parser. Ovi parseri implementirani su na gotovo svim programskim jezicima koji se danas koriste, pa u svakom od njih postoji API (Aplication Programming Interface) koji omogućava korišćenje funkcionalnosti datih parsera.

3.2.2 XML Shema

XML šema je prema [13] opšte definisana kao organizacija ili struktura baze podataka obično izvedena iz modeliranja podataka. Ova struktura je obično opisana korišćenjem neke vrste kontrolisanog rečnika koji imenuje stavke podataka i sadrži sva ograničenja koja se mogu primeniti (*vrsta podataka, legalne i ilegalne vrednosti, posebnao formatiranje itd.*). Odnosi među stavkama podataka su takođe važan deo svake šeme.

XML Shema predstavlja *xml* dokument u kome se nalazi opis pravila koja treba da ispuni jedan *xml* dokument da bi se mogao smatrati validnim dokumentom. Validacija *xml* dokumenta je postupak u kome se utvrđuje da li određeni dokument ispunjava pravila navedena u *xml* šemi. Validacija je obavezan postupak u razmeni podataka preko *xml* dokumenata. Prijemna strana mora da proveri da li pristigli dokument zadovoljava određeni format kako bi bila sigurna da se određeni podaci nalaze tačno tamo gde se i traže. *XML* parseri podržavaju validaciju *xml* dokumenata.

3.3 DOM (Document Object Model)

DOM se prema [15] definiše kao skup interfejsa za izgradnju objektnog modela u obliku stabla od parsiranog *XML* dokumenta. Ovim modelom možemo manipulisati korišćenjem različitih metoda koje omogućavaju dodavanje, uklanjanje i izmenu čvorova stabla. *DOM* omogućava direktan pristup određenim podacima u *XML* dokumentu.

DOM predstavlja skup čvorova, ili delova informacija organizovanih u hijerarhiju. Ova hijerarhija omogućava pretraživanje stabla i pronalaženje odgovarajućih informacija. *DOM* podrazumeva parsiranje čitavog dokumenta u objektni model. Treba imati u vidu da kod velikih dokumenata, ovo parsiranje može biti dosta sporo i oduzimati mnogo resursa, i u takvim slučajevima, naročito ako nam nije potrebna izmena bilo kog dela dokumenta, treba razmisliti o korišćenju nekog drugog načina parsiranja (recimo *Simple API for XML* (*SAX*)).

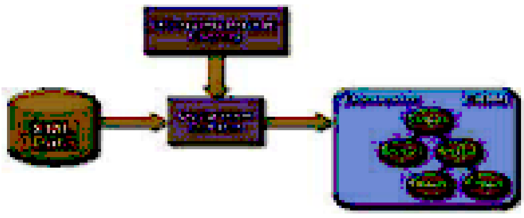
U *DOM*-u, rad sa *XML* dokumentima podrazumeva da se one najpre podele u čvorove. Svaki čvor pripada nekom tipu, a najčešći tipovi su: čvorovi elemenata *(element node)*, tekstualni čvorovi *(text node)*, čvorovi atributa *(attribute node)* i dokument čvor *(attribute node)* - roditelj svih drugih čvorova u dokumentu. Pored nabrojanih postoji još vrsta čvorova koji se ređe koriste. Neki od njih su: *CDATA* – sadrži informacije koje parser ne treba da analizira, već samo da prosledi kao običan tekst, *Comment* – komentari, *Processing Instructions* – informacije koje se prosleđuju aplikaciji.

Da bi obrađvali informacije iz *XML* dokumenta, dokument mora biti parsiran da kreira *Document* objekat. Ovaj objekat je interfejs, što znači da ne može biti instanciran direktno.

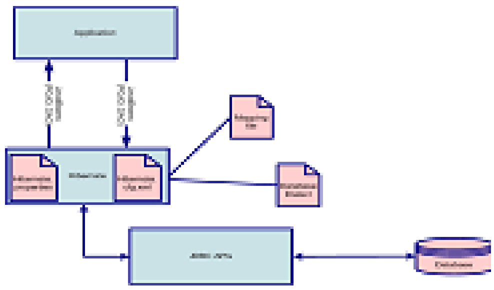
Parsiranje dokumenta je proces koji se sastoji iz 3 koraka. Prvi korak je kreiranje *DocumentBuilderFactory*-ja. Zatim se kreira *DocumentBuilder*. I poslednji korak je parsiranje fajla da bi *Document* objekat bio napravljen. Na slici 7 je predstavljen proces parsiranja dokumenta.

3.4 Hibernate

Hibernate je prema [8] *open source* objektno/relacioni alat za mapiranje za *JAVA*-u koji omogućava razvoj perzistentnih klasa i perzistentne logike nezavisno od načina čuvanja podataka. *Hibernate* ne brine samo o mapiranju *JAVA* klasa



Slika 7: Proces parsiranja dokumenta [19]



Slika 8: *Hibernate* arhitektura [20]

u tabele u bazi podataka (i mapiranju *JAVA*-inih tipova podataka u *Structure Query Language (SQL)* tipove), već i podržava upite nad podacima i pronalaženje objekata i time znatno smanjuje vreme potrebno za razvoj aplikacije koje bi inače potrošili za ručno upravljanje podacima u *SQL*-u ili *JAVA Database Connectivity*-ju *(JDBC)*. Prema [8], cilj *Hibernate*-a je oslobađanje programera od 95% uobičajenih perzistentnih programskih zadataka.

Hibernate čini lakšim razvoj robusnih aplikacija koje koriste baze podataka u *JAVA*-i. Potrebno je samo napisati jednostavan *Plain Old Java Object (POJO)*, napraviti *XML* fajl za mapiranje koji opisuje relacije između baze podataka i atributa klase, i poziva *Hibernate API* za učitavanje/smeštanje perzistentnih objekata. Na slici 8 je predstavljena arhitektura aplikacije koja koristi *Hibernate*.

Kao što je prikazano na slici *Hibernate* koristi *JDBC API* za komunikaciju sa bazom podataka. *Hibernate* konfiguracioni fajl *Hibernate.cfg.xml* sadrži podatke koje *Hibernate* koristi da bi uspostavio konekciju sa bazom podataka. Tu su smešteni podaci o SUBP-u, samoj bazi koju koristi aplikacija i drajveru koji se koristi za konekciju na potrebnu bazu podataka. Pisanje ovog fajla se može izbeći, ali je u tom slučaju potreban fajl *Hibernate.properties* koji sadrži sve ove podatke. Takođe, u njemu se nalaze i imena fajlova koji se koriste za mapiranje (fajlovi sa ekstenzijom *.hbm.xml)*.

Obično se u aplikaciji koja koristi *Hibernate* za svaku tabelu iz baze pravi klasa sa atributima koji odgovaraju kolonama u tabeli. Za svaku klasu je potrebno napisati i fajl sa imenom *ime_ tabele.hbm.xml,* u kome su smešteni detalji o načinu mapiranja kolona iz tabele i atributa te klase. Takođe, u ovom fajlu možemo regulisati kardinalnosti veza.

3.5 EJB (Enterprise Java Beans)

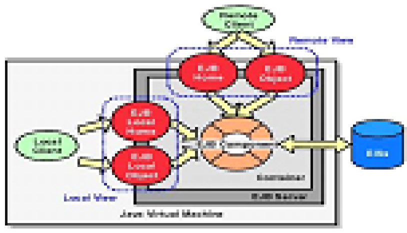
Prema [10, 11] *EJB* predstavlja arhitekturu za serverske, modularne distribuirane aplikacije velikih razmera pisane u *Java* programskom jeziku. Od njenog uvođenja do danas, tehnologija je ostvarila snažan impuls kod proizvođača platformi, kao i kod timova za razvoj aplikacija velikih razmera. Glavni razlog za to je što *EJB* model pojednostavljuje razvoj poslovnih komponenti sistema koje su: sigurne, distribuirane, perzistentne, transakcionalne, skalabilne i prenosive.

Sama arhitektura smanjuje kompleksnost razvoja poslovnih komponenti tako što obezbeđuje automatsku podršku za servise sistemskog nivoa, omogućujući na taj način programeru da se u potpunosti posveti ravijanju poslovne logike. Na slici 9 je prikazana *EJB* arhitektura. Međutim, veoma je važno razjasniti pojmove *EJB* servera i *EJB* kontejnera. Prema [7] *EJB* server predstavlja deo aplikacije na kome se nalaze *EJB* kontejneri. Nekada se naziva i *Enterprise Java Server* (*EJS*). *EJB* server obezbeđuje implementaciju servisa koji su zajednički za sve *EJB* aplikacije. Uloga *EJB* servera je da sakrije kompleksnost ovih servisa od komponenti koje ih koriste.

EJB specifikacija navodi sedam servisa koje *EJB* server mora da obezbedi:

1. Imenovanje (engl. *Naming*),
2. Transakcije (engl. *Transaction*),
3. Bezbednost (engl. *Security*),
4. Stalnost (engl. *Persistence*),
5. Konkurentnost(engl. *Concurency*),
6. Životni ciklus (engl. *Life cycle*),
7. Razmenu poruka (engl. *Messaging*).

Sa druge strane, *EJB* kontejner funkcioniše kao *runtime* okruženje za komponente *EJB* aplikacije, na taj način što održava i prilagođava primarne servise koji su neophodni za upravljanje svakom komponentom u vreme njenog izvršavanja. Pored toga što predstavlja posrednika između komponenti aplikacije i servisa koje pruža *EJB* server, *EJB* kontejner obezbeđuje i upravljanje životnim ciklusom instance *EJB* komponente, kao i njenu identifikaciju. *EJB* kontejneri kreiraju instance *beanova*, održavaju skupove (engl. *pool*) instanci i uništavaju ih. Kontejneri su transparentni za klijenta, to znači da ne postoji klijentski *API* koji omogućuje manipulisanje sa kontejnerom i ne postoji način da klijent odredi u kom će se kontejneru razvijati koja komponenta sistema. Jedna od osnovnih



Slika 9: Pogled na *EJB* arhitekturu [7]

uloga kontejnera jeste da udaljenim klijentima obezbedi pristup do komponenti koje su u njemu razvijene.

EJB komponente se nalaze unutar *EJB* kontejnera, koji za njih predstavlja izvršno okruženje. Kako se navodi u [6, 7] kontejner obezbeđuje servise za upravljanje životnim ciklusom komponenti, kao i interfejse prema *EJB* serveru.

3.5.1 Interfejsi

Kako bi objekti klijenti mogli da šalju poruke *EJB* komponentama, komponente moraju da obezbede određene poglede. Pogled predstavlja interfejs klijenta prema *bean*-u i može biti lokalan ili udaljeni. Lokalni interfejs mogu da koriste samo lokalni klijenti kako bi pristupili *EJB* komponentama. Udaljeni interfejs omogućuje svim klijentima da pristupe *EJB* komponentama. Koji će se interfejs koristiti zavisi od toga da li se klijentski kod i kod metode *Bsession bean*-a izvršavaju na istoj ili na različitoj *JAVA Virtual Machine*-I (*JVM*). Ukoliko se izvršavaju na istoj JVM treba koristiti loaklni interfejs. U suprotnom treba koristiti udaljeni (*remote*) interfejs.

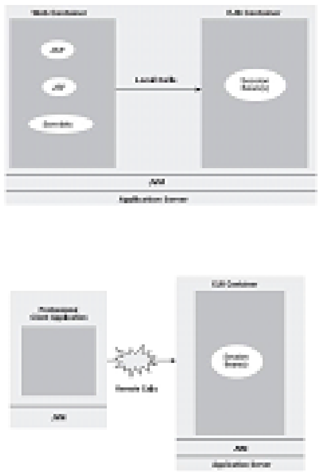
Koncept lokalnog interfejsa uveden je u *EJB 2.0* specifikaciji, a inspirisan je činjenicom da udaljeni pozivi zahtevaju znatno više resursa od lokalnih. Koji od ova dva interfejsa će se koristiti zavisi od toga kako klijent vidi komponente *EJB* aplikacije. *EJB* klijent može biti eksterni (*udaljeni*) klijent, kao što je servlet koji se pokreće na nekom drugom procesu, ili interni (*lokalni*) klijent, kao što je drugi *EJB*.

Koji interfejs treba koristiti, koje klase nasleđivati i ostala pravila koja treba slediti pri razvoju *bean*-ova, zavisi od tipa *enterprise bean-a* za koji se odlučimo u našoj aplikaciji. U nastavku je dat kratak uvod u tipove *enterprise bean*-ova.

3.5.2 *EJB* tipovi

Prema [6, 7] postoje tri glavna tipa *EJB-a*: *entity beans*, *session beans* i *message driven beans*.

Entity beans su modelovani za predstavljanje poslovnih ili domenskih kocepata i obično predstavljaju imenice koje se pojavljuju u sistemu. Odnosno predstavl-



Slika 10: Korišćenje lokalnog i udaljenog interfejsa *Session bean*-a od strane *web* klijenta [7]

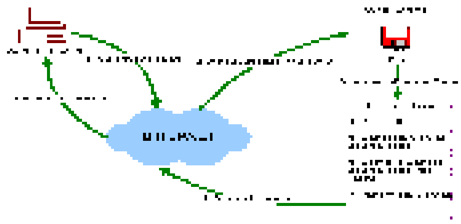
jaju podatke (*entitete*) koji se čuvaju u bazama podataka.Na osnovu toga kako se održava njihova perzistentnost, *entity bean*-ovi se dele na dva tipa. Prvi tip značava slučaj kada sam kontejner brine o perzistentnosti i to je *container managed persistence* tip (*CMP*), a drugi tip je kada sam *bean* definiše načine za održavanje perzistentnosti, odnosno *bean managed persistence* tip (*BMP*).

Session bean-ovi su modelovani za predstavljanje zadataka ili tokova podataka koji se odvijaju u sistemu, kao i da obezbede koordinaciju između ovih aktivnosti. Obično se koriste za implementaciju "fasade" u odnosu na ostale *EJB* module.

Session bean je *enterprise bean* koji direktno komunicira sa korisnikom i sadrži poslovnu logiku aplikacije. *Session bean* predstavlja pristupanje aplikaciji od strane klijenta, postavljenoj na serveru, pozivanjem neke njegove metode. Aplikacija može sadržati više sesija, u zavisnosti od broja korisnika koji joj pristupaju. *Session bean* pravi interaktivnu sesiju samo za jednog korisnika i štiti ga od složenosti izvršavanjem poslovnog zadatka na strani servera. *Session bean* koristi *entity bean*-ove za pristup ili modifikaciju podataka. U njemu je enkapsulirana logika koja odlučuje koji će podaci biti modifikovani. *Session bean* je relativno kratkog životnog veka. *EJB* kontejner može uništiti *Session bean* ukoliko nije korišćen izvesno vreme.

Na slici 10 A, je prikazana upotreba *Session bean*-a preko lokalnog interfejsa, a na slici 10 B je predstavljeno korišćenje *Session bean*-a preko udaljenog interfejsa.

Postoje dve vrste *session bean*-ova. *Stateless session bean*-ovi ne održavaju nikakva konverzaciona stanja i kontejner odlučuje kada će upotrebiti određenu instancu. *Statefull session bean*-ovi na neki način čuvaju podatke o prethodnim sesijama sa određenim klijentima. Što znači da će određeni klijent uvek koristiti istu instancu *bean*-a. *Message driven bean*-ovi su *enterprise bean*-ovi koji



Slika 11: Generisanje dinamičkog sadržaja sa *JSP* elementima [12]

omogućavaju *JAVA EE* aplikacijama da asinhrono procesiraju poruke. Poruka može biti poslata od strane bilo koje *JAVA EE* komponente (*klijentske aplikacije, drugog enterprise bean-a, ili web komponente*), ili od strane *JMBS* aplikacije ili sistema koji ne koristi *JAVA EE* tehnologiju. *Message driven bean*-ovi mogu procesirati *JMBS* poruke ili druge tipove poruka.

3.6 JSP (Java Server Pages)

Prema definiciji koja se nalazi u [11] *JSP* predstavlja tehnologiju za razvoj *web* stranica koje imaju dinamički sadržaj. Za razliku od običnih *HTML* stranica, koje imaju statički sadržaj koji se nikada ne menja, *JSP* stranica može da menja svoj sadržaj u zavisnosti od velikog broja različitih parametara, uključujući identitet korisnika, tip pretraživača koji korisnik upotrebljava, informacija koje korisnik unosi, izbora koji korisnik napravi, itd. Ove karakteristike predstavljaju osnovu mnogih poslovnih aplikacija.

JSP stranica sadrži standardne markerske elemente, kao što su *HTML* tagovi, kao i svaka druga obična *web* stranica. Međutim, pored toga, *JSP* stranica sadrži i specijalne *JSP* elemente koji omogućuju serveru da postavi dinamički sadržaj na stranicu. *JSP* elementi se mogu koristiti u različite namene, kao što su uzimanje informacija iz baze podataka, ili proveravanje korisničkog unosa.

Kada korisnik zatraži neku *JSP* stranicu, *web browser* šalje zahtev *web* serveru preko Interneta. *Web* server prepoznaje da korisnik zahteva *JSP* stranicu po ekstenziji (.jsp). Tada *web* server prosleđuje *JBSP* fajl ka *JSP Servlet Engine*-u. Ukoliko se *JSP* fajl poziva prvi put, tada se on prvo parsira. Sledeći korak je generisanje posebnog *Servlet*-a na osnovu *JSP* fajla. *Servlet source code* se kompajlira u klasu. Zatim se *Servlet* instancira, izvršava svoje metode i na osnovu toga generiše *HTML* fajl koji se šalje korisniku. Ukoliko već postoji klasa *Servlet*-a onda se odmah instancira *Servlet* i kao u predhodnom slučaju, formira se *HTML* fajl koji se šalje korisniku. Kompletan scenario je prikazan na slici 11.

JSP definiše veliki broj standardnih elemenata koji su korisni za veliki broj *web* aplikacija, kao što su elementi kojima se pristupa *JavaBeans* [10] komponentama, prosleđivanje kontrole između stranica i deljenje informacija između zahteva, stranica i korisnika. Programeri mogu da prošire *JSP* sintaksu implementiranjem elemenata koji su specifični za datu aplikaciju, a koji obavljaju zadatke kao što su pristup *EJB* komponentama, bazama podataka, slanje elektronske pošte i generisanje *HTML* koda za predstavljanje podataka specifičnih za datu aplikaciju. Kombinovanjem standardnih i prilagođenih elemenata mogu se kreirati veoma složene web aplikacije.

U [6] je dato i objašnjenje zašto treba koristiti *JSP*. Naime, u prvim danima *web* a, *Common Gateway Interface* (*CGI*) bio je jedini alat za razvoj dinamičkog *web* sadržaja. Međutim, *CGI* ne predstavlja dovoljno efikasno rešenje. Za svaki zahtev koji se prosledi, *web* server mora da kreira novi proces operativnog sistema, učita interpreter i skript, izvši skript, a zatim sve to ponovo nastavi.Ovo je veoma zahtevno za server i dovodi do velikih problema kada se poveća saobraćaj.

Veliki broj alternativa za *CGI* kreiran je tokom godina, neki od njih su *FastCGI*, *mod_perl* za *Apache*, *NSAPI* od *Netscape*a, *ISAPI* od *Microsoft*a i *Java* servleti od *Sun Microsystems*. Dok ove tehnologije pružaju bolje performanse i veću skalabilnost, sve one i dalje imaju isti problem: generišu *web* stranice tako što ugrađuju *HTML* direktno u programski kod. Ovo iziskuje kreiranje *web* stranica na način na koji to vide sami programeri. *JavaServer Pages*, međutim, sve to menja.

Ovo poglavlje predstavlja kratak uvod u glavne tehnologije korišćene u izradi seminarskog rada. U nastavku će biti dat prikaz rešenja problema postavljenog u ovom seminarskom radu.

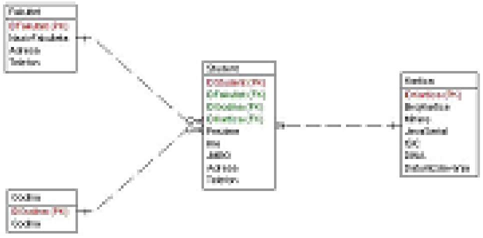
4 Realizacija predloženog rešenja

U ovom poglavlju biće opisan logički model klijentske i serverske baze podataka, kao i logički model i implementacija klijentske i serverske aplikacije. Za izradu modela relacionih baza podataka korišćen je alat *CASE studio 2*, dok su logički modeli aplikacija izrađeni objektno-orijentisanim jezikom *UML* [16], uz pomoć alata *Rational Rose 2003*.

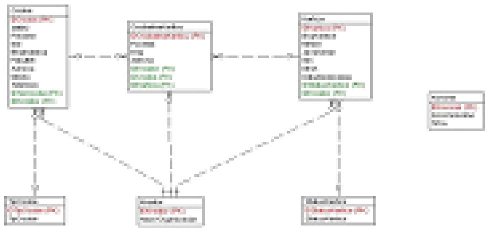
4.1 Logički modeli baza podataka

4.1.1 Logički model klijentske baze podataka

Kao što je navedeno u Poglavlju 2 ovog rada, svaka od ustanova Univerziteta u Istočnom Sarajevu vodi podatke o svojim studentima u relacionim bazama podataka. Strukture, kao i sistemi za upravljanje bazama podataka (SUBP) se razlikuju od ustanove do ustanove, zbog čega je potreban mehanizam unificiranog pristupa bazama podataka. Na slici 12 je prikazan primer logičkog modela baze podataka koja je korišćena u procesu izrade i testiranja informacionog sistema razvijenog u ovom radu i u kojoj se nalaze svi relevantni podaci o studentu.



Slika 12: Primer logičkog modela klijentske baze podataka



Slika 13: Logički model serverske baze podataka

4.1.2 Logički model serverske baze podataka

Centralna baza podataka, razvijena za potrebe informacionog sistema se nalazi u CRS-u, i u njoj se vode sledeći podaci:

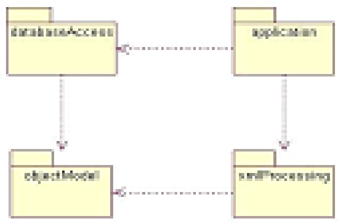
1. podaci o korisnicima (*studentima*) – osnovni lični podaci,
2. podaci o karticama – broj kartice, broj kontaktnog i MIFARE čipa, status kartice, apleti koji se na kartici nalaze i slično,
3. pripadnost kartice korisniku,
4. izdavač kartice, odnosno podaci o ustanovi koja je izdala karticu,
5. svi ostali potrebni podaci (*izvršeni lekarski pregledi i slično*).

Logički model serverke baze podataka je prikazan na slici 13.

4.2 Logički model i implementacija klijentske aplikacije

Osnovni zadatak klijentske aplikacije je eksportovanje u *xml* fajl onih podataka koji su na bilo koji način izmenjeni (*dodani, obrisani, promenjeni zapisi*) nakon poslednjeg ažuriranja podataka u centralnoj bazi podataka. Pored ovog, klijentska aplikacija obezbeđuje mogućnost ažuriranja podataka u lokalnoj bazi podataka

Slika 14: Dijagram slučajeva upotrebe klijentske aplikacije



Slika 15: Paketi klijentske aplikacije

na osnovu *xml* fajla sa promenama, preuzetog sa servera, kao i pregled izveštaja. Dijagram slučajeva upotrebe (*engl. Use Case Diagram*) klijentske aplikacije je prikazan na slici 14.

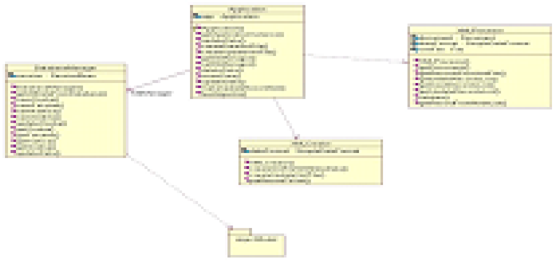
Klijentska aplikacija implementirana je u programskom jeziku *Java*. Koncipirana je tako da ima četiri paketa, tačnije pakete *application*, *objectModel, databaseAccess* i *xmlProcesing*, kao što je prikazano na slici 15.

U paketu *application* nalaze se klase koje obezbeđuju funkcionalnost aplikacije. U paketu *objectModel* nalaze se klase koje mapiraju tabele klijentske baze. To znači da svaka klasa ovog paketa predstavlja određenu tabelu iz relacione baze podataka iz čega se zaključuje da se paket *objectModel* bitno razlikuje od ustanove do ustanove baš kao i baze podataka različitih ustanova. Na slici 16 prikazan je dijagram klasa paketa *objectModel* za navedeni logički model klijentske baze podataka.

Paket *databaseAccess* je odgovoran za pristupanje i rad sa bazom podataka, dok su klase paketa *xmlProcesing* zadužene za kreiranje i obradu *XML* fajlova. Dijagram klasa klijentske aplikacije prikazan je na slici 17.

Klasa *XMLCreator* odgovorna je za kreiranje *xml* fajlova. Ona poseduje funkcije koje kreirju *xml* fajl sa svim podacima iz baze podataka (funkcija *createXmlFileWithNewData())* i *xml* fajl sa izmenjenim podacima (funkcija *createUpdateXMLFile())*. Klasa *XMLProcessor* odgovorna je za procesiranje *xml* fajlova, odnosno izvlačenje podataka iz *xml* fajlova i kreiranje lista sa svim podacima iz fajla. U klasi *Application* nalaze se osnovne funkcionalnosti sistema. Klasa *Applica-*

Slika 16: Dijagram klasa paketa *objectModel*

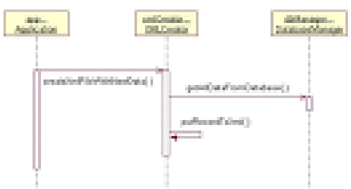


Slika 17: Dijagram klasa klijentske aplikacije

tion koristi funkcionalnost klase *XMLCreator* za kreiranje *xml* fajla sa izmenjenim podacima, dok za kreiranje *xml* fajla sa svim podacima koristi funkcionalnosti klase *DatabaseManager*. Klasa *DatabaseManager* obezbeđuje prustup i rad sa zapisima u bazi podataka. Klasa *Application* koristi funkcionalnosti klase *XML-Processor*.

Na slici 18 prikazan je dijagram sekvenci koji opisuje kako se međusobnom interakcijom instanci odgovarajućih klasa obezbeđuje kreiranje *xml* fajla sa kompletnom slikom podataka iz klijentske baze.

Na slici 19 prikazan je dijagram sekvenci za kreiranje *xml* fajla koji sadrži samo izmenjene podatke.



Slika 18: Dijagram sekvenci za *Application.createXmlFileWithNewData()*

Slika 19: Dijagram sekvenci za *Application.createUpdateXmlFile()*

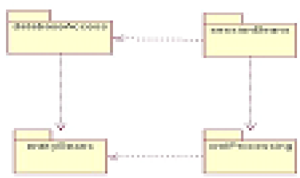
Slika 20: Dijagram slučajeva upotrebe serverske aplikacije

4.3 Logički model i implementacija serverske aplikacije

Serverska aplikacija obezbeđuje slanje *xml-update* fajla serveru od strane klijenata kao i ažuriranje podataka u centralnoj bazi CRS-a na osnovu sadržaja pristiglih *xml* fajlova. Pored navedenog, serverska aplikacija omogućuje klijentima, odnosno ustanovama Univerziteta u Istočnom Sarajevu preuzimanje *xml* fajlova sa servera radi ažuriranja klijentskih baza podataka, kao i pregled izmenjenih zapisa u vidu izveštaja. Dijagram slučajeva upotrebe serverske aplikacije prikazan je na slici 20.

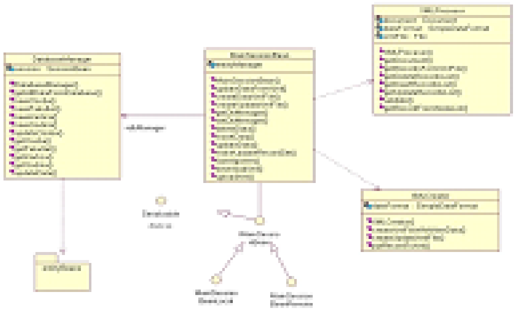
Administrator sistema je, kao što se vidi na slici, zadužen za administriranje naloga (*otvaranje, brisanje, ograničavanje naloga i slično*), i za ažuriranje podataka u centralnoj bazi CRS-a.

Serverska aplikacija implementirana je u programskom jeziku *Java*. Koncipirana je tako da ima četiri paketa, tačnije pakete *sessionBeans*, entityBeans, *databaseAccess* i *xmlProcesing*, kao što je prikazano na slici 21.



Slika 21: Paketi serverske aplikacije

Slika 22: Dijagram klasa paketa *entityBeans*



Slika 23: Dijagram klasa serverske aplikacije

Prilikom implementacije serverske aplikacije korišćena je tehnologija *Enterprise Java Beans 3.0*. Upotrebljene su dve vrste *bean*-ove: *session bean*, koji se nalaze u paketu *sessionBeans* i implementiraju poslovnu logiku sistema, i *entity bean*, koji se nalaze u paketu *entityBeans* i predstavljaju objektni model serverske aplikacije.

U paketu *sessionBeans* nalaze se klase koje obezbeđuju funkcionalnost aplikacije. U paketu *entityBeans* nalaze se klase koje mapiraju tabele serverske baze. To znači da svaka klasa ovog paketa predstavlja određenu tabelu iz relacione baze podataka centralne baze CRS-a. Na slici 22 prikazan je dijagram klasa paketa *entityBeans* za navedeni logički model klijentske baze podataka.

Dijagram klasa klijentske aplikacije prikazan je na slici 23.

Klasa *XMLCreator* odgovorna je za kreiranje *xml* fajlova. Ona poseduje funkcije koje kreirju *xml* fajl sa svim podacima iz centralne baze podataka (funkcija *createXmlFileWithNewData()*) i *xml* fajl sa izmenjenim podacima (funkcija *createUpdateXMLFile()*). Klasa *XMLProcessor* odgovorna je za procesiranje *xml* fajlova, odnosno čitanje podataka iz *xml* fajla i kreiranje lista sa svim podacima iz fajla. U klasi *MainSessionBean* nalaze se osnovne funkcionalnosti sistema. Klasa *MainSessionBean* predstavlja *session bean*. Ona realizuje interfejs *IMainSessionBean* koga nasleđuju interfejsi *MainSessionBeanLocal*, koji služi za lokalne pozive

Slika 24: Dijagram sekvenci za *MainSessionBean.updateData()*

bean-a, i *MainSessionBeanRemote*, za udaljene pozive *bean*-a. Koristi funkcionalnost klase *XMLCreator* za kreiranje *xml* fajla sa izmenjenim podacima, dok za kreiranje *xml* fajla sa svim podacima koristi i funkcionalnosti klase *DatabaseManager* (pre svega funkciju *getAllData()*). Klasa *DatabaseManager* obezbeđuje serverskoj aplikaciji prustup i rad sa zapisima u bazi podataka. Klasa *MainSessionBean* koristi i funkcionalnosti klase *XMLProcessor* koje su joj potrebne za procesiranje *xml* fajlova.

Na slici 24 prikazan je dijagram sekvenci koji opisuje kako se međusobnom interakcijom instanci odgovarajućih klasa obezbeđuje kreiranje *xml* fajla sa kompletnom slikom podataka iz klijentske baze.

4.4 Izgled korisničkog interfejsa aplikacija

U ovom delu će biti prikazano kako izgledaju korisnički interfejsi klijentske i serverske aplikacije.

4.4.1 Izgled korisničkog interfejsa klijentske aplikacije

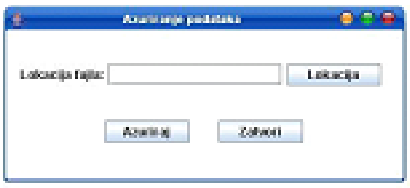
Celokupan korisnički interfejs klijentske aplikacije je izrađen korišćenjem biblioteke *Swing*. Prema [17] *Swing* je biblioteka grafičkih komponenti razvijena u okviru *JAVA2*, i predstavlja zamenu za stariju *AWT* [18] biblioteku.

Klijentska aplikacija se sastoji od glavnog prozora i nekoliko podprozora. Prvo što korisnik vidi nakon pokretanja aplikacije je, svakako, glavni prozor koji je predstavljen na slici 25.

Kao što je prikazano na slici, na glavnom prozoru klijentske aplikacije korisnik može da se odluči da izvrši neku od sledećih operacija: Ažuriranje podataka, eksportovanje podaka, pregled izveštaja i, naravno može da zatvori aplikaciju.

Ukoliko se klijent odluči na izvršavanje operacije ažuriranja podataka, aplikacija će od njega zahtevati da unese lokaciju *xml* fajla preuzetog sa servera posredstvom serverske aplikacije. Sistem će proveriti validnost *update* fajla i ukoliko je sve u redu ažuriraće klijentsku bazu podataka. Izgled prozora Ažuriranje podataka je prikazan na slici 26.

Slika 25: Izgled glavnog prozora klijentske aplikacije



Slika 26: Izgled prozora Ažuriranje podataka

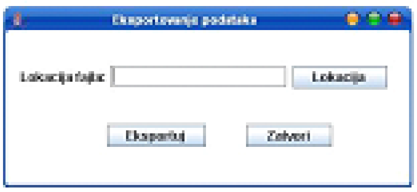
Prozor Eksportovanje promena je po samom izgledu veoma sličan prozoru Ažuriranje podataka. Bitna razlika je što se u ovom slučaju od korisnika sistema zahteva unošenje imena i lokacije fajla u kome će biti sačuvani svi zapisi koji su na bilo koji način izmenjeni u periodu od poslednjeg eksportovanja promena. Izgled prozora Eksportovanje promena je predstavljen na slici 27.

Pre nego što se odluči za izvlačenje promena u novi *xml* fajl, korisnik sistema može najpre steći uvid u broj, kao i u same zapise koji su izmenjeni od predhodnog eksportovanja promena. Ovo može uraditi izborom akcije Izveštaj. Izgled prozora Izveštaj je prikazan na slici 28.

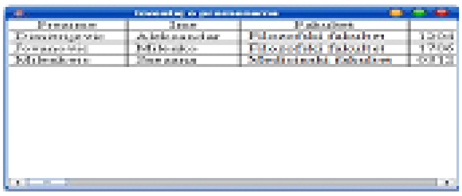
4.4.2 Izgled korisničkog interfejsa serverske aplikacije

Sam interfejs korisničke aplikacije sastoji se iz *HTML* i *JSP* stranica. Prva stranica u aplikaciji jeste *login.jsp*, čiji izgled je prikazan na slici 29.

Da bi mogli da koriste usluge inforamcionog sistema, članovi osoblja uni-



Slika 27: Izgled prozora Eksportovanje promena



Slika 28: Izgled prozora Izveštaj

Slika 29: Prikaz stranice *login.jsp*

verzitetskih ustanova unose korisničko ime i lozinku. Nakon toga vrši se provera unetih parametara i u zavisnosti od toga da li su uneti podaci ispravni ili ne, korisniku se otvara stranica *welcomeScreen.jsp* odnosno *error.jsp*. Na stranici *error.jsp* se samo ispisuje poruka o grešci i omogućuje korisniku da se ponovo vrati na stranicu za unos korisničkog imena i lozinke, dok na stranici *welcomeScreen*.jsp, čiji je izgled predstavljen na slici 30, korisnik može da bira, da li želi da preuzme fajl sa poslednjim promenama u centralnoj bazi podataka, ili želi da pošalje serveru fajl sa svojim izmenjenim podacima.

U zavisnosti od izbora, korisniku se otvara dijalog u kome se od korisnika zahteva da odabere lokaciju na svom računaru gde će fajl biti snimljen nakon

Slika 30: Prikaz stranice *welcomeScreen.jsp*

Slika 31: Prikaz stranice *upload.jsp*

čega počinje preuzimanje fajla i njegovo smeštanje na željenu lokaciju ukoliko je odabrao opciju Preuzimanje fajlova sa poslednjim promenama, odnosno otvara se stranica *upload.jsp* ukoliko je dabrao opciju Slanje fajla sa poslednjim promenama. Izgled stranice *upload.jsp* predstavljen je na slici 31.

Na ovoj stranici korisnik bira lokaciju fajla na kome su sačuvane poslednje promene iz klijentske baze podataka, i isti šalje serveru kako bi se ažurirali i podaci u serverskoj bazi, a samim tim i kako bi ostale ustanove Univerziteta u Istočnom Sarajevu dobile informaciju o izmeni određenih podataka i ažurirali svoje lokalne baze podataka.

5 Mogućnosti daljeg usavršavanja razvijenog rešenja

Osnovni zahtevi postavljeni u ovom radu bili su razvoj informacionog sistema za integrisanje funkcija univerziteta zasnovanog na SOA, a koji će pokazati primenu *Web* servisa i obezbediti unificiran pristup bazama podataka. Razvijena aplikacija zadovoljava zahteve postavljene u radu i sa tog aspekta je potpuno funkcionalna.

Aplikaciju je do sada koristio samo autor, što znači da još uvek nije testirana u realnim uslovima. Ipak, za sada se aplikacija pokazala kao dovoljno pouzdana, sa dovoljno brzim transportom podataka kroz mrežu.

S obzirom na modularnost rešenja, veoma je lako dodati nova proširenja. Za sada rešenje radi sa dve baze podataka, odnosno sadrži dva modula poslovne logike. Za svaku narednu bazu, odnosno svaku novu ustanovu univerziteta koja bi se priključila na ovaj informacioni sistem treba dodati još jedan modul, što zahteva male izmene u poslovnoj logici.

Klijentska aplikacija je razvijena samo za potrebe testiranja sistema, konačna aplikacija treba da izgleda nešto drugačije. Glavna zamisao je da se podaci o studentima vode na *Smart* karticama, što bi omogućilo automatsko identifikovanje prilikom njihovog obraćanja bilo kojoj zasebnoj jedinici univerziteta. U svim ustanovama univerziteta bi u tom slučaju morao da postoji čitač kartica, gde bi se prilikom posete, ubacila kartica studenta i zaposleni bi trenutno dobio sve bitne podatke o studentu.

Mora se voditi računa i prilikom identifikacije osoblja. Pre svega, potrebno je poboljšati sigurnost prilikom identifikacije. Pored toga, potrebno je zaštiti i sam *Web* servis. Značajnu ulogu pri tom ima jedna od novih oblasti zaštite računarskih sistema, zaštita *Web* servisa. Tehnologija je još uvek nova i nedovoljno prihvaćena, a samim tim predstavlja i dodatni izazov, kao i neophodan korak u daljem razvoju rešenja.

6 Zaključak

U radu je prikazan razvoj informacionog sistema za integrisanje funkcija univerziteta zasnovanog na SOA primenom tehnologije *web* servisa za ostvarenje unifikacije pristupa bazama podataka, na primeru razmene i dijeljenja podataka između visokoškolskih ustanova Univerziteta u Istočnom Sarajevu. Razvijena aplikacija omogućuje stejkholderima, ustanova univerziteta uvid u sve bitne podatke o studentima koji se vode u različitim ustanovama, čime bi trebalo da se poboljša kvalitet pruženih usluga.

U rešenju su primenjene relativno nove tehnologije koje su danas sve zastupljenije u svetu i kod nas. Zasniva se na primeni *web* servisa baziranih na *Enterprise Java Beans* tehnologiji, dok je klijentska aplikacija razvijena u *Java Server Pages* tehnologiji.

Razvijeno rešenje može da se primeni na nivou jednog univerzitetskog centra koji čini više ustanova. Sistem se može, uz odgovarajuću nadogradnju, primeniti i na nivou širem od jednog univerziteta, pa i na nivou čitave države. Primenom sličnih principa može se razviti informacioni sistem koji bi se koristio u drugim oblastima.

U daljem radu se očekuje testiranje aplikacije u realnim uslovima. Pored toga planira se proširenje predloženog rešenja kako bi se zadovoljili svi korisnički zahtevi. Važan zadatak je i poboljšanje karakteristika sistema sa bezbednosnog aspekta.

Razvijeno rešenje bi trebalo da predstavlja početak implementacije jednog složenog informacionog sistema, koji će doprineti znatno boljoj komunikaciji između visokoškolskih ustanova. Zbog kompleksnosti problema, za dalji rad će biti potreban tim programera koji bi rešenje realizovali u potpunosti.

Literatura

[1] Thomas Erl, Service-Oriented Architecture, CISCO, August 2009.

[2] George Chang Marcus J. Healey James A. M. McHugh Jason T. L. Wang, Mining the World Wide Web, Kluwer Academic Publishers, 2001.

[3] David Chappell, Tyler Jewell, Java Web Services, O'Reilly, March 2002.

[4] Dr Milica Vučković, Marko Petrović, Razvoj web servisa korišćenjem .NET platforme, Fakultet Organizacionih Nauka –Beograd.

[5] J. Connalen, Building Web Applications with UML, Addison–Wesley, 2000.

[6] Sun Microsystems, Enterprise Java Beans Specification Version 3.0, Sun Microsystems, November 12, 2005.

[7] R. P. Sriganesh, Mastering Enterprise Java Beans 3.0, Wiley, April 2006.

[8] C. Bauer, G. King, Hibernate in Action, Manning Publication, 2005.

[9] Will Iverson, Hibernate - a J2EE Developers Guide, Permalink, April 2005.

[10] C. Walls, R. Breidenbach, Spring in Action, Manning Publication, 2005.

[11] Hans Bergsten, Java Server Pages, 3rd edition, O'Reilly, December 2003.

[12] Marty Hall, Core Servlets And Java Server Pages, Sun Press, 2004.

[13] Chuck White, Liam Quinn, Linda Burman: Mastering XML

[14] Eric Newcomer: Understanding web services XML, SOAP, UDDI and WSDL, May 2002

[15] Jeremy Keith: DOM Scripting, September 2005.

[16] G. Booch, J. Rumbaugh, I. Jacobson, The Unified Modeling Language User Guide, AddisonWesley, 1999.

[17] Marc Loy, Robert Eckstein, Dave Wood, James Elliott, Brian Cole, Java Swing Second Edition, O'Reilly, November 2002.

[18] John Zukowski, Java Awt Reference, O'Reilly, January 1998.

[19] Tutorial: The Document Object Model APIs, Figure 4-2 DOM APIs

[20] Tutorial: Hibernate Tutorial, Intruduction to Hibernate

CIP - Каталогизација у публикацији
Народна и универзитетска библиотека
Републике Српске, Бања Лука

51(082)

MATHEMATICAL Conference of the Republic of Srpska (3 ; 2013 ; Trebinje)

Proceedings. Volume II / Third Mathematical Conference of the Republic of Srpska, Trebinje, 07-08 June 2013 = Zbornik radova. tom II / Treća matematička konferencija Republike Srpske ; [glavni urednik Milenko Pikula]. - 1. izd. - Trebinje : Fakultet za proizvodnju i menadžment, 2014 (Trebinje : Grafokomerc). - 128 str. : ilustr. ; 25 cm

Tiraž 300. - Bibliografija uz sve radove. - Abstracts.

ISBN 978-99976-600-1-5

COBISS.RS-ID 4289304