

Potencijal tehnike pronalaženja znanja za automatsko sastavljanje testova

Miroslava Ignjatović, Dragan Bojić, Bojan Furlan, Igor Tartalja

Sadržaj – Kreiranje testova za proveru znanja u akademskim ustanovama je postupak koji se periodično ponavlja i koji zahteva od ispitivača napor i uloženo vreme. Postupak je složen i na ispitivaču je odgovornost za objektivnost i konzistentnost testova, odnosno pouzdanost testiranja. Automatizacijom pojedinih koraka se može značajno poboljšati objektivnost i olakšati postupak ispitivaču. U ovom radu će biti razmotren potencijal postupka za automatsko sastavljanje testova na osnovu banke testova tehnikom pronalaženja znanja zasnovanom na algoritmu najbližeg suseda.

Keywords — automatsko sastavljanje testova, pronalaženje znanja, test, priprema testa, test znanja, najbliži sused

I. UVOD

Provera znanja studenata je veoma važan deo procesa sticanja akademskog obrazovanja. Postupak kreiranja testa je složen postupak [1] koji se u akademskim ustanovama ponavlja nekoliko puta u toku jedne školske godine za svaki predmet. Kod ručno sastavljanog testa, ispitivač uglavnom nema precizne kriterijume po kojima sastavlja test. Najčešće korišćeni kriterijumi su vrsta i broj stavki na testu, kao i raspodela stavki po oblastima predmeta za koji se test sastavlja. Na osnovu njih, ispitivač kreira nove stavke testa i sastavlja test od novih ili već korišćenih stavki, sa ciljem da zadovolji svoje kriterijume. Procena težine novih stavki kao i težine samog testa je prepuštena subjektivnoj proceni ispitivača. Na njemu je i odgovornost da testovi pomoću kojih se periodično proverava znanje određenog predmeta budu približno iste težine. Glavni problemi koji se u postupku ručno sastavljanog testa javljaju jesu značajan mentalni napor i utrošak vremena visokoobrazovanog kadra, kao i problem konzistencije testa koji se koristi za proveru znanja. Automatizacijom nekih od koraka u ovom postupku navedeni problemi se značajno umanjuju [1].

Postojeća rešenja [2]-[10] se zasnivaju na procesu sastavljanja testa na osnovu zadate specifikacije testa. Specifikacija testa se najčešće prikazuje kao skup ograničenja atributa stavke kao i testa kao celine. Takođe se definiše ciljna funkcija koja se maksimizuje ili minimizuje [1]. Pronalaženje najbolje kombinacije stavki bi zahtevalo pretragu celog prostora stanja, što je u slučaju

realne baze stavki, zbog eksponencijalne složenosti kombinatornog algoritma, neprihvatljivo dug process. Zato se u praksi pribegava traženju suboptimalnog rešenja. Problem sastavljanja testa spada u klasu problema kombinatorne optimizacije [1]. Problemi kombinatorne optimizacije se mogu rešiti uz pomoć specifičnih softverskih alata [4]-[6], često zasnovanih na linearnom programiranju (eng. *solver*). Primeri takvih softverskih alata su IBM ILOG CPLEX [4], Lingo 14.0 [5], i biblioteka *lp_solve5.5* [6] koji mogu da se koriste i za automatsko sastavljanje testa. Problem pri korišćenju ovakvih alata je njihova složenost, što kao posledicu ima slabu primenu u akademskom testiranju. Automatsko sastavljanje testa je moguće i uz pomoć Microsoft Excela i Premium Solver Platforme kao nadogradnje [7]- [9]. Banka pitanja i specifikacija testa se smeštaju u Excel fajl, što olakšava posao specifikacije kriterijuma testa.

Osim solvera, primenjuju se i razne heuristike [10]-[13]. U radu [10] koristi se modifikovani postupak planinarenja, dok se u radovima [11]-[13] koriste varijante genetskih algoritama u postupku sastavljanja testa.

U ovom radu je prikazan rezultat razvoja programa *testMiner* za automatsko sastavljanje testova tehnikom pronalaženja znanja uz korišćenje algoritma najbližeg suseda, za pronalaženje prihvatljivog kandidata. Cilj rada je procena potencijala ove tehnike za automatsko sastavljanje testova. Za pregled stavki, oblasti, kriterijuma stavki kao i testa je korišćen program *testBase*[3]. Za razvoj programa *testMiner* je korišćen programski jezik Java. Stavke, kriterijumi i testovi su bili smešteni u MS SQL Server bazi podataka.

U radu je najpre opisan problem i predlog rešenja novog alata, a zatim opisani rezultati primene tog alata.

II. OPIS PROBLEMA

Globalni ciljevi rada su da se proces sastavljanja testa učini efikasnijim i postigne konzistentnost između novo sastavljenih testova i prethodno držanih testova. Automatizam se odnosi na postupak odabira stavki testa na osnovu definisanih kriterijuma, banke stavki i prethodno održanih testova. Postupak kojim se ostvaruje ovaj automatizam se odnosi na otkrivanje znanja o sastavljanju testova na osnovu ranije držanih testova, koji postoje u banci testova. Konzistentnost se postiže uz pomoć metrike za određivanje rastojanja novog testa od prethodno održanih testova sa ciljem da novo sastavljeni test bude dovoljno „blizu“ testovima iz referentnog skupa, koji čine svi testovi odgovarajuće vrste iz banke testova, bez obzira na način kako su ti testovi sastavljeni – ručno ili automatski (nekom drugom metodom). Pretpostavka je da su testovi u banci valjani i pouzdani. Neposredni cilj rada

Miroslava Ignjatović je sa Visoke škole strukovnih studija za informacione i komunikacione tehnologije, Zdravka Čelara 16, 11000 Beograd, Srbija; E-mail: miroslava.ignjatovic@ict.edu.rs a Dragan Bojić, Bojan Furlan i Igor Tartalja, su sa Elektrotehničkog fakulteta Univerziteta u Beogradu; E-mail: bojic@etf.rs, bojan.furlan@etf.rs, tartalja@rcub.bg.ac.rs.

je da se, bez posebne optimizacije postupka pretrage prostora stanja pri kombinovanju stavki iz postojeće banke stavki, samo proceni potencijal tehnike pronalaženja znanja i metrike. Postupak je zasnovan na rastojanju novo sastavljenog testa od ranije držanih testova.

III. PREDLOG REŠENJA

Program testMiner je namenjen sastavljanju novog testa znanja tehnikom pronalaženja znanja, na osnovu postojećih stavki, ranije držanih testova i kriterijuma koji ograničavaju izbor stavki za test. Osnovna ideja rešenja je da se pronađe skup stavki koji ne narušava ograničenja testa, a čije je rastojanje od ranije držanih testova u opsegu postojećih uzajamnih rastojanja.

A. Atributi stavke i testa

Svaki test se sastoji od stavki. Stavke su opisane sledećim atributima:

- Vrsta stavke: korišćena baza stavki je sadržala dve vrste stavki – pitanja sa ponuđenim odgovorima (bira se jedan od ponuđenih odgovora) i zadaci (stavka čije se rešenje kreira).

- Oblast pripadnosti stavke: svaka stavka može pripadati eksplicitno ili implicitno jednoj ili više oblasti iz stabla oblasti. Za svaki predmet je definisan skup oblasti koje predmet pokriva.

- Težina stavke: za stavke koje su se nalazile na već održanim testovima, težina (0-100%) se određuje na osnovu rezultata testiranja ispitanika, a za nove stavke se procenjuje na osnovu ekspertskog iskustva sastavljača testa. Na pitanjima se težina određuje kao 100-(procenat tačnih odgovora), a na zadacima kao 100-(srednja procentualna ispunjenost ocenjivanih koncepata).

Svaki test ili pod-test (skup stavki iste vrste, koje pripadaju istoj oblasti) je takođe opisan posebnim skupom atributa. Svi skupovi atributa, za ceo test ili pod-test, uključuju iste attribute. Sumarno, test je opisan jednim skupom vrednosti atributa za ceo test i jednom matricom (broj vrsta stavki x broj oblasti predmeta) čiji svaki element sadrži skup vrednosti istih atributa kojima se opisuje i ceo test.

Sagledani su sledeći atributi testa, odnosno delova testa:

- BrojStavki: broj stavki na testu, odnosno pod-testu.
- Vrsta testa: identifikator koji ukazuje na vrstu testa (npr. kolokvijum 1, kolokvijum 2, završni ispit).
- MinTezina: najmanja težina za pojedinačnu stavku obuhvaćenu testom, odnosno pod-testom.
- MaxTezina: najveća težina za pojedinačnu stavku obuhvaćenu testom, odnosno pod-testom.
- SrednjaTezina: srednja težina stavki obuhvaćenih testom, odnosno pod-testom.

B. Filtriranje stavke i kandidat-testova

Kriterijumi testa koji se sastavlja, odnosno dozvoljene granične vrednosti atributa stavke i testa, se određuju na osnovu referentnog skupa. Referentni skup čine svi ranije držani testovi zadatog predmeta i zadate vrste testa. Izračunate granične vrednosti atributa se koriste kao kriterijumi za filtriranje stavke i testova da bi se smanjio prostor pretrage. Postoje dve vrste kriterijuma za

filtriranje:

1. Pojedinačni eliminatorni kriterijumi (PE). Kriterijum PE služi za filtriranje stavke na osnovu pripadnosti oblasti: oblast stavke mora da pripada predmetu, odnosno oblastima koje pripadaju tom predmetu. Filtriranje se obavlja na početku, pre generisanja prvog kandidat-testa.

2. Grupni eliminatorni kriterijumi (GE). Kriterijumi GE se odnose na test u celini. Oni sprečavaju pojavu kombinacija stavki koja ih ne ispunjava. Vrednosti sledećih atributa (pod)testa treba da budu u opsegu minimalne i maksimalne dozvoljene vrednosti utvrđene na prethodno opisani način:

- a. težina testa;
- b. broj stavki po oblastima;
- c. težina svakog pod-testa.

Koristi se i GE kriterijum koji se odnosi na ponavljanje stavki sa ranijih testova u novom testu: broj stavki iste vrste koje su se zajedno pojavile na nekom testu iz referentnog skupa ne sme da pređe polovinu ukupnog broja stavki te vrste testa na novom testu.

Ukoliko kandidat-test ne zadovoljava GE, on se eliminiše iz daljeg razmatranja.

C. Postupak sastavljanja testa

Test se sastavlja na sledeći način:

1. Na početku se definiše iz kog predmeta i koja vrsta testa se sastavlja, kao i broj stavki po vrstama stavki (na primer, ispit iz predmeta Programiranje 1 sa 5 pitanja i 2 zadatka). Definiše se i ciljni broj zadovoljavajućih kandidat-testova kao kriterijum završetka postupka. Zadovoljavajući kandidat-test je kandidat koji zadovoljava eliminatorne kriterijume i čije je rastojanje od referentnog skupa u opsegu srednjeg rastojanja testova unutar referentnog skupa plus/minus standardna devijacija.

2. Na osnovu zadatog predmeta i vrste testa se obrazuje referentni skup testova. Izračunavaju se dozvoljene granične vrednosti atributa testa (koje se koriste kao parametri GE kriterijuma), srednje rastojanje između testova referentnog skupa, kao i standardna devijacija tog rastojanja.

3. Filtriraju se stavke na osnovu PE kriterijuma. Konkretno, proverava se samo pripadnost oblastima predmeta za koji se sastavlja test. Ne proverava se težina stavke u odnosu na min i max težinu stavke iz referentnog skupa, kako bi se dozvolilo da se pojavi i teža stavka od najteže, odnosno lakša od najlakše korišćene na testu u referentnom skupu.

4. Stavke koje prolaze test PE kriterijuma se ne rangiraju, već se pseudo-slučajno biraju i redom smeštaju u matricu stavki u kojoj svaki red sadrži stavke koji pripadaju jednoj vrsti stavke. Na ovaj način se obezbeđuje različitost u sastavljenim predlozima testa za istu bazu stavki, isti referentni skup testova i iste kriterijume sastavljanja testa.

5. Sastavlja se kandidat-test. Iz matrice filtriranih stavki se biraju stavke i formira se test postupkom koji generiše sledeću kombinaciju koristeći princip da se u kombinaciju što kasnije uključe stavke koje imaju veći broj indeksa. Na primer, neka u matrici postoje dve vrste, prva sa pitanjima

$V1=\{P0,P1,P2,P3,P4\}$ i druga sa zadacima $V2=\{Z0,Z1\}$. Ukoliko bi novi test trebalo da sadrži dva pitanja i jedan zadatak postupak bi izgenerisao sledeće kandidate, navedenim redom: P0P1Z0, P0P1Z1, P0P2Z0, P0P2Z1, P1P2Z0, P1P2Z1, P0P3Z0, P0P3Z1, P1P3Z0, P1P3Z1, P2P3Z0, P2P3Z1, P0P4Z0, P0P4Z1, P1P4Z0, P1P4Z1, P2P4Z0, P2P4Z1, P3P4Z0, P3P4Z1.

6. Za svaki kandidat-test se ispituje da li zadovoljava GE kriterijume. Ukoliko kandidat-test ne zadovoljava GE kriterijume, ponavlja se korak 5 i generiše se sledeći kandidat-test.

7. Ukoliko kandidat-test zadovoljava GE kriterijume, računa se srednje rastojanje između datog kandidat-testa i testova iz referentnog skupa. Ukoliko je to rastojanje u definisanom opsegu (parametri pronađeni u koraku 2), kandidat-test postaje zadovoljavajući test. On se pamti kao izabrani kandidat ukoliko ima manje rastojanje od prethodno izabranog kandidata. Ukoliko kandidat-test ne zadovolji GE kriterijume, ponavlja se korak 5.

8. Ukoliko se ne pronađe zadovoljavajući test u zadatom vremenskom roku (podrazumevano 5s), postupak se ponavlja od koraka 5 (*random-restart*), čime se ponovo pseudo-slučajno smeštaju filtrirane stavke u niz stavki i kombinacije ponovo počinju da se generišu od prve. One su različite od kombinacija iz prethodnog pokušaja, zbog drugačijeg uređenja stavki.

9. Postupak se završava ukoliko je analiziran zadat broj zadovoljavajućih testova. Rezultat postupka je poslednji zapamćeni izabrani kandidat-test.

D. Merenje rastojanja između dva testa

Rastojanje između dva testa se meri na osnovu vrednosti atributa testova čije se rastojanje traži. Svaki test je opisan skupom atributa testa i matricom skupova atributa podtestova. Rastojanje (d) između dva testa se izračunava prema formuli 1:

$$d^{Total}(x^*, y^*) = \frac{1}{2}d^{test} + \frac{1}{2 * k * m} \sum_{j=1}^{k * m} d_j^{podtest} \quad (1)$$

gde je k – broj vrsta stavki, m – ukupan broj oblasti predmeta. Vrednosti d^{test} i $d^{podtest}$ se računaju prema formuli 2 za euklidsko rastojanje:

$$d(x^*, y^*) = \sqrt{\sum_{i=1}^n w_i (x_i^* - y_i^*)^2} \quad (2)$$

gde su x^* i y^* vektori normalizovanih vrednosti atributa dva testa čije se rastojanje traži, a w_i je težinski faktor i -tog atributa, pri čemu je: $\sum_{i=1}^n w_i = 1$. Podrazumevane vrednosti težinskih faktora atributa: minimalna težina, maksimalna težina, srednja težina i broj stavki su 0.2, 0.2, 0.2 i 0.4 respektivno.

Kako bi svi atributi testa imali podjednak uticaj (ne računajući težinske faktore koji se eksplicitno dodeljuju), pri određivanju rastojanja kandidata od referentnog skupa, potrebno je izvršiti normalizaciju atributa, tj. njihovo preslikavanje u opseg [0,1]. Korišćena je MIN-MAX normalizacija, te se normalizovana vrednost atributa dobija prema formuli 3:

$$x^* = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (3)$$

Za min i max vrednosti atributa se uzimaju granice mogućih vrednosti atributa, nezavisne od vrednosti atributa testova referentnog skupa.

IV. REZULTATI

U korišćenoj bazi podataka su se nalazili stavke i testovi (ispiti i kolokvijumi) sa predmeta Programski jezici 1 i Programski jezici 2, na smeru Računarska tehnika i informatika, koji su održani u periodu od februara 2004. do oktobra 2012. na Elektrotehničkom Fakultetu u Beogradu. Baza je sadržala ukupno 150 testova i 1075 stavki. Testove je rešavalo oko 7500 studenata. Broj oblasti koje su se pojavljivale na testovima jeste 34.

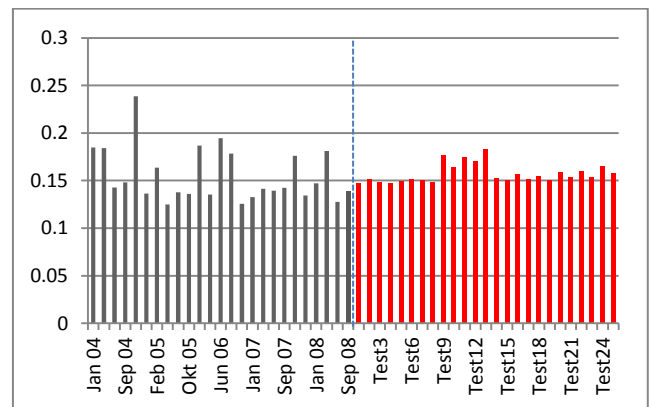
Kao primer rada programa su prikazani rezultati sastavljanje testa iz predmeta Programiranje1 sa 6 pitanja i 2 zadatka. Traženo je 100 zadovoljavajućih kandidat-testova.

Posle filtriranja stavki, na osnovu oblasti predmeta, za pitanja je bilo na raspolaganju 437 stavki, dok je za zadatke bilo na raspolaganju 118 stavki. Da se pretraživao ceo prostor stanja, sa ciljem pronajdenja kombinacija stavki sa najmanjim rastojanjem, bilo bi potrebno pronaći rastojanja za sledeći broj kombinacija pitanja i zadataka:

$$\binom{437}{6} * \binom{118}{2} = 9,34 * 10^{12} * 6903 = 6.45 * 10^{16}$$

Pošto je zbog eksponencijalne složenosti problema bilo nemoguće pretražiti ceo prostor stanja u realnom vremenu, kombinacije su se generisale samo dok nije pronađen zadat broj zadovoljavajućih kandidat-testova, uz primenu tehnike *random-restart*.

Na Sl. 1 su prikazana srednja rastojanja po 25 testova referentnog skupa i testova automatski sastavljenih prethodno opisanom postupkom. Srednje rastojanje testova referentnog skupa je 0.155 sa standardnom devijacijom 0.028. Srednje rastojanje skupa sastavljenih testova je 0.157 sa standardnom devijacijom 0.0098.



Sl. 1. Srednja rastojanja testova referentnog skupa (levo) i automatski sastavljenih testova (desno)

U Tabeli 1 su prikazani rezultati F-testa kojim se utvrđivao statistički značaj razlike standardne devijacije rastojanja za skup kandidat-testova koji su zadovoljili samo GE kriterijume, kao i skupa zadovoljavajućih testova, koji pored zadovoljenih GE kriterijuma imaju prihvatljivo rastojanje od testova u referentnom skupu, u

odnosu na standardnu devijaciju rastojanja testova referentnog skupa.

TABLE 1: REZULTATI JEDNOSMERNOG F-TESTA

	<i>Testovi referentnog skupa</i>	<i>Kandidat testovi koji zadovoljavaju GE</i>	<i>Zadovoljavajuć i testovi</i>
Srednja vrednost	0.155	0.174	0.157
Varijansa	7.9E-04	6E-04	9.6E-05
Broj testova	25	40	25
Statistika testa F		1.312	8.22
F_{krit}		1.8	1.98

Rezultat F-testa pokazuje da razlika u standardnoj devijaciji rastojanja testova referentnog skupa i standardnoj devijaciji rastojanja kandidat-testova koji zadovoljavaju samo GE kriterijume nije statistički značajna, zato što je $F < F_{krit}$. Međutim, razlika u standardnoj devijaciji rastojanja testova referentnog skupa i standardnoj devijaciji rastojanja zadovoljavajućih testova statistički jeste značajna, zato što je $F > F_{krit}$. Taj zaključak opravdava uvođenje dodatnog kriterijuma kojim se ograničava rastojanje kandidat-testa, odnosno primenu tehnike automatskog sastavljanja testova koja se predlaže u ovom radu.

CPU vreme potrebno da se reši ovaj model na PC-u sa I5 CPU 1.7GHz i 6MB RAM-a je bio između 1s i 20s. Performanse zavise od početnog nasumičnog smeštanja stavki u niz stavki, ali navedene vrednosti ukazuju na prihvatljivo vreme za automatsko sastavljanje testa, čak i u najgorem slučaju.

V. ZAKLJUČAK

Opisan postupak automatskog sastavljanja testa tehnikom pronalazanja znanja daje zadovoljavajuće rezultate i značajno olakšava postupak sastavljanja testa od postojećih stavki, oslobodivši ispitivača subjektivnog uticaja na izbor stavki. Pokazano je da tehnika ima potencijal, čak i uz primenu jednostavnog mehanizma za izbegavanje upadanja pretrage u deo prostora bez brzog rešenja, kao što je *random-restart*. Ukazani potencijal tehnike otvara novi pravac za istraživanje primene boljih heuristika pretrage.

Pošto je u postojećoj banci testova ponavljanje stavki na testovima bilo zanemarljivo, nisu se mogli koristiti atributi stavke koji se odnose na ponavljanje stavki, kao što su broj ponavljanja i vreme od poslednjeg pojavljivanja stavke na Međutim, predloženi princip sastavljanja testa se na taj način ne bi promenio.

LITERATURA

- [1] W. Linden "Linear Models for Optimal Test Design". 1st ed. New York, NY: Springer Science+Business Media, Inc., 2005
- [2] J. Protić, D. Bojić, I. Tartalja „test: Tools for evaluation of students' tests - a development experience“, *Proc. 31st ASEE/IEEE*

Frontiers in Education Conference, Reno, NV, USA, str. F3A-6-F3A-12., 2001

- [3] M. Krunić, J. Protić, I. Tartalja, "Struktura programa testBASE za održavanje baze ispitnih problema, baze kriterijuma izbora i interaktivnu pripremu ispita", *Zbornik radova XL konferencije ETRAN-a, Vol. 3*, Budva, str.259-262, 1996
- [4] IBM ILOG CPLEX Optimization Studio, <http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud>, last access 08.06.2014.
- [5] Lingo software, <http://www.lindo.com/> last access 09.07.2014.
- [6] Ip_solve 5.5, <http://lpsolve.sourceforge.net/>, last access 08.06.2014.
- [7] K. Cor, C. Alves, M.J. Gierl, "Computer Software Review: Conducting Automated Test Assembly Using the Premium Solver Platform Version 7.0 With Microsoft Excel and the Large-Scale LP/QP Solver Engine Add-In" *Applied Psychological Measurement*, 32(8), pp.652-663, 2008
- [8] K. Cor, C. Alves, M.J. Gierl "Three applications of automated test assembly within a user-friendly modeling environment". *Practical Assessment, Research & Evaluation*, 14(14), <http://pareonline.net/pdf/v14n14.pdf>, last access 09.07.2014.
- [9] Premium Solver Platform, <http://www.solver.com/premium-solver-platform>, last access 08.06.2014.
- [10] D. Bojić, P. Cerović, J. Protić, I. Tartalja, "testGEN: Programski alat za poluautomatsko generisanje testa znanja", *Zbornik trećeg simpozijuma YUINFO*, Brezovica, pp. 687-691, 1997
- [11] A. Verschoor, "Genetic Algorithm for Automated Test Assembly". Ph.D. dissertation Universiteit Twente., 2007
- [12] K. Sun, Y. Chen, S. Tsai and C. Cheng "Creating IRT-based parallel test forms using the genetic algorithm method" *Applied measurement in education*, 21(2), pp.141-161, 2008
- [13] M. Yildirim "A genetic algorithm for generating test from a question bank" *Computer Applications in Engineering Education*, 18(2), pp.298-305, 2010

ABSTRACT

Creating knowledge tests in academic institutions is a procedure that is frequently repeated, and which requires examiner's effort and significant time invested. The procedure is complex and demands objectivity of the examiner, test consistency and reliability. Automating the procedure can improve objectivity and facilitate the process to the examiner. This paper analyses potential of the proposed method for automated test assembly based on existing test bank, using method knowledge discovery and the nearest neighbor algorithm.

KNOWLEDGE DISCOVERY POTENTIAL IN AUTOMATED TEST ASSEMBLY

Miroslava Ignjatović, Dragan Bojić, Bojan Furlan, Igor Tartalja