

РАЗРАБОТВАНЕ НА АЛГОРИТЪМ ЗА ГЕНЕРИРАНЕ  
НА КОМПЮТЪРНА ТЕСТОВА КОМБИНАЦИЯИван КУЮМДЖИЕВ<sup>1</sup>JEL I210, C830,  
C880

Резюме

## Ключови думи:

оценяване във висшето образование, компютърно генериране на тестова комбинация, адаптивни тестове

В статията са разгледани компютърно генерираните тестове като един от най-ефективните методи за оценяване във висшето образование. Изследването разкрива проблеми свързани с автоматизираното генериране на тестова комбинация с произволни въпроси. Предложен е алгоритъм, който да се справи с посочените проблеми като използва предварително дефинирана сложност на всеки от въпросите в теста. Проведени са многобройни тестове на алгоритъма, за да бъде оценено неговото качество при различни изисквания зададени от потребителите. Направените изследвания и получените резултати демонстрират ефективен метод за компютърно генериране на форма на изпитване, която може да бъде описана като предхождаща адаптивните тестове.

В условията на криза повече от всякога е необходимо повишаване на качеството на висшето образование, както и увеличаване ефективността на работа на университетите. Важна част от процеса на обучение е оценяването на знанията на студентите. Процесът на оценяване включва няколко заинтересовани страни. На първо място, това са студенти и преподаватели. Студентите получават представа дали знанията и уменията, които са получили са ясно изразени и измерими към даден момент от време. За преподавателя то става условие и механизъм в регулацията на собствената му работа по отношение на учебното съдържание – разбрани и усвоени ли са отделните теми, кои от тях затрудняват студентите повече от нормалното и т.н. (Десев и др, 1977).

Друг участник в процеса на оценяване са висшите учебни заведения. Проверката на нивото на знанията на студентите (в т.ч. кандидат-студентски изпити, текущ контрол, семестриални оценки, държавни изпити) заема голямо количество ресурси – заетост на учебни зали и преподаватели, печатни материали и т.н., което значи, че повишаване на ефективността дори в малка степен би довело до значителни по размер икономии в глобален мащаб.

<sup>1</sup> Катедра Информатика, Икономически университет – Варна, България. e-mail: ivan\_ognyanov@ue-varna.bg

Заинтересуваните страни включват и законодатели, работодатели и родители (Patrick и др., 2008). Държавата субсидира частично висшите училища и налага различни критерии за акредитация, в които са включени и изисквания за ефективност на оценяването и използване на модерни методи за тази цел (НАОА, 2011). Работодателите, от друга страна, имат нужда от увереност, че наемат подготвени специалисти, а родителите често плащат за образованието на децата си и приемат оценяването като измерител на прогреса им.

От посоченото следва, че повишаването на ефективността на изпитването ще доведе до положителен отзвук, както от страна на учащи, работодатели и родители, така и от преподаватели, университетско ръководство и държавни институции. Вземайки предвид този факт, целта на статията е разработване и тестване на ефективността на модул от тестова система, представляващ алгоритъм за генериране на тестови комбинации.

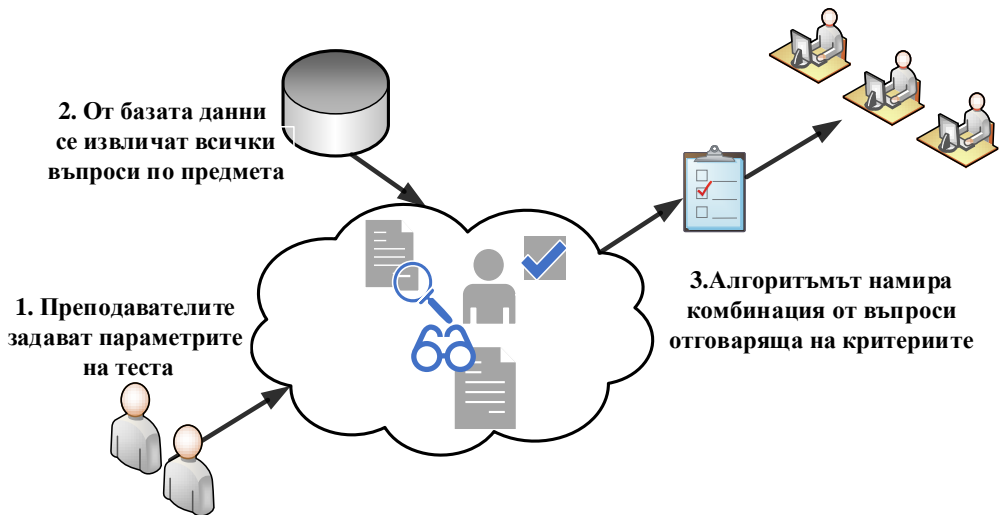
### **1. Тестова форма на изпитване и проблеми при компютърно генериране на тестова комбинация при определени критерии**

Най-разпространената форма на изпитване е използването на тестове. Разновидностите могат да бъдат определени на база формата на въпросите (истина/неистина, кратък отговор, въпроси за допълване, комбиниране на отговорите, въпроси с множествен избор (Гюрова, В. и др., 2006)) или методът на оценяване (оценка само на верни отговори, отнемане на точки за грешни отговори, награждаване за пропуснати отговори, оценка на частични знания, въпросите с повече от 1 верен отговор, отнемане на точки за пропуснати отговори (Lesage и др, 2013)). Извършени са множество изследвания в търсене на отговора на въпроса „кой е най-точния метод за оценяване“ и „какъв е най-добрият формат на теста“, които стигат до разнопосочни резултати (Kastner и Stangla, 2011, Ventouras и др., 2010). Но въпреки споровете в научната общност за сложността на видовете тестове и ефективността на начина на оценяване, е безспорно, че провеждането на компютърно генерирани тестове с множествен избор има потенциал да елиминира в голяма степен субективния фактор при оценяването. Степента на обективност зависи от това каква част от процеса е автоматизирана. При частично автоматизираните тестове изборът на въпроси, които да бъдат включени в теста, и/или оценяването на резултатите се извършва от човек. За да бъде сведен до минимум субективният фактор е необходимо и създаването на тестове и оценяването на резултатите да бъдат автоматизирани.

Разработеният алгоритъм за генериране на компютърни тестове е част от цялостна тестова система, която свежда участието на преподавателите в създаването на изпита до въвеждане на банка с тестови въпроси и избор на критерии за генериране на тест. За всеки от въпросите преподавателите задават теглови коефициент (в интервала 1 до 5), който позволява прецизно определяне на общата сложност на теста. След приключване на изпит системата извежда справка с информация за най-често грешените и правилно решаваните въпроси, която може да бъде използвана като база за редактиране на техните теглови коефициенти. След като бъде определена сложността на всеки от въпросите, системата ще разполага с достатъчно разнообразна база от данни, която може да бъде използвана за добавяне на опция за генериране на адаптивни тестове. При тях на всеки от въпросите е зададено ниво на сложност и следващия въпрос от теста зависи от отговорите на предишните (Ескенази и др., 1993). При една от разновидностите на адаптивните, тестове ако оценяваният отговори правилно – следващият въпрос ще бъде с по-висока сложност, в противен случай – с по-ниска. Според някои изследователи, (Wainer и др., 2014) адаптивните тестове предоставят точен измерител на знанията за студенти с различни нива на подготовка. Според същото изследване, класическият тест с множествен избор оценява точно само студентите със средно ниво на знания, но ощетява тези с по-високи и по-ниски нива на знание.

Въпреки посочените предимства на компютърно генерираните тестове, намаляването на експертното участие в етапите на създаване на вариантите и оценяването им предполага разработване на специализиран и достатъчно гъвкав алгоритъм, който да позволява генериране на тестове по предварително зададени критерии. За да бъде създаден качествен модул, следва да се изследват изискванията на всеки от потребителите.

Действието на алгоритъма за генериране на тестове може да бъде описано на концептуално ниво на фиг. 1, където са идентифицирани следните ползватели на функциите му – преподаватели, администратори на системата, студенти/кандидат-студенти, програмисти.



**Фиг. 1. Взаимодействие с алгоритъма за генериране на тестове**

Параметрите на теста се задават от преподавателите и въвеждат от администраторите на системата. Времетраенето, сложността на теста (сума от тегловите коефициенти на въпросите, включени в него), студентите за които е активен и времеви интервал, в който да бъде достъпен, са характеристики на всеки от създаваните тестове. Освен тях са налични следните варианти:

- А) генериране на тест по зададен общ брой въпроси;
- Б) генериране на тест по зададен общ брой въпроси от точно определени теми;
- В) генериране на тест чрез избор на точно определен брой въпроси от всяка тема.

Анализът на изискванията сочи, че, от гледна точка на преподавателите системата следва да генерира тест с въпроси, избрани на случаен принцип, удовлетворяващи посочените критерии.

Следващите действащи лица в процеса на създаване на тестове са администраторите. Те са потребителите, които въвеждат заложените от преподавателите изисквания и проследяват за успешното изпълнение на алгоритъма. Като основни параметри за качествена работа на системата администраторите посочват бързо изпълнение на зададените условия, както и високо ниво на диалог със системата – съобщенията при некоректен вход показват не само причината, а и начина на отстраняването ѝ.

След генериране на теста, той е достъпен за изпитваните в посочения времеви интервал. Студентите са пасивни потребители на алгоритъма – не участват в неговото изпълнение и изискванията им за справедливо разпределение на въпросите зависят от критериите, зададени от преподавателя при генериране на теста.

Алгоритъмът за генериране на тестове е част от цялата система на тестовия център и като такава част взаимодейства с други нейни модули, като модулите за разбъркване на въпроси и отговори, които са част от модул за визуализиране на теста и т.н. Изискванията, наложени от другите програмисти, включват бързодействие и унифициран, предвидим изходен резултат от изпълнението, който да може да бъде използван в зависимите модули.

Основен проблем при разработката на системата е създаването на алгоритъм, който да работи ефективно, както с големи, така и с малки бази от данни. Основните фактори, влияещи на работата, са: брой въпроси в теста, брой въпроси в базата от данни, брой въпроси в базата от данни, които удовлетворяват изискванията на тегловите коефициенти. Броят на последните зависи до голяма степен и от интервала за сложност на крайния тест при задаването на критериите – колкото е по-голям този интервал, толкова повече ще бъдат и въпросите, способни да удовлетворят зададения критерий. От тази гледна точка са възможни няколко комбинации на разпределение на въпросите в базата от данни:

- к1) базата от данни съдържа много въпроси с разнообразни теглови коефициенти, а потребителят изисква създаване на тест с малко на брой въпроси. Например, избор на тест от 10 въпроса от 200 налични в базата от данни, разпределени в 10 теми по 20;
- к2) базата от данни съдържа много въпроси, потребителят изисква голям тест, но броят на въпросите в теста отново е малък процент от наличните в базата. Генериране на тест от 100 въпроса при наличие на 10 теми с по 70 въпроса в базата данни;
- к3) броят на въпросите в базата с теглови коефициенти, удовлетворяващи критериите на потребителите е близък до броя въпроси, необходими за създаване на теста;
- К3.1) създаване на голям тест от 150 въпроса при налични 20 теми по 10 въпроса;
- К3.2) създаване на малък тест от 15 въпроса от тип текущ контрол при малка база от данни, състояща се от 10 теми с по 2 въпроса във всяка.

Като се анализират изложените примери и факти, могат да бъдат изведени следните основни проблеми:

1. Силно неравномерно разпределение на въпроси. При използване на вариант А) за генериране на тест съществува вероятност случайно избраните въпроси да бъдат осезаемо неравномерно разпределени по теми – в т.ч. тестът да се състои от въпроси само от една тема и/или някои теми изобщо да не бъдат засегнати.

Нека имаме банка от  $u$  на брой тестови теми, всяка от които съдържа по  $v$  на брой въпроса и нека да извлечем случайно  $w$  от тях. Тъй като общият брой на въпросите е равен на  $u \cdot v$ , то броят на всички възможни комбинации на изтегляне на въпросите е

$$n = C_{(u \cdot v)}^w = \frac{(u \cdot v)!}{w!(u \cdot v - w)!} \quad (1)$$

Нека сега да изчислим вероятността изтеглените въпроси да са разпределени в точно  $k$  теми. Очевидно  $k \leq \min(u, v)$ .

Номерираме в произволен ред темите от 1 до  $u$ . Тогава всяко възможно разпределение на въпросите в отделните теми се описва с вектор с размер  $u$  теми с  $k$  строго положителни компоненти и  $u - k$  нулеви компоненти:

$$X = (x_1, x_2, \dots, x_u), \quad (2)$$

за който са изпълнени условията:

$$\sum_{i=1}^u x_i = w \text{ и } x_i \leq v \quad (i = \overline{1..u}).$$

Ако компонентите на вектора (2) са подредени в намаляващ порядък, то той ще има вида

$$X_{k, s \in S}^* = (x_1, x_2, \dots, x_k, 0, \dots, 0), \text{ като } x_1 \geq x_2 \geq \dots \geq x_k \geq 1, \quad (3)$$

където  $S = \{X_s^*\}$  е крайно множество от вектори.

Множеството на векторите (2) включва всички възможни пермутации от елементите на всички вектори (3). Тъй като това са пермутации с повторения, броят им се намира по формулата

$$\bar{P}_{k, s \in S} = \frac{u!}{(u-k)! \cdot k_{s,1}! \cdot k_{s,2}! \cdot \dots \cdot k_{s,t}!}, \quad (4)$$

където:  $(u-k)$  е брой на нулевите елементи на (4);

$$k_{s,j}, (j = \overline{1 \dots t}) - \text{брой на елементите, равни на } 1, 2, \dots, t, \sum_{j=1}^t k_{s,j} = w.$$

Броят на различните варианти за извличане на въпросите от темите за всеки вектор (3) ще бъде равен на

$$\left[ \prod_{i=1}^u \frac{v!}{x_i!(v-x_i)!} \right]_{s \in S} \quad (5)$$

Векторите (3) представляват неподредени разбивания на числото  $w$  на точно  $k$  положителни числа. Броят им се определя с рекурентни равенства (Холл, 1970). Алгоритми за генериране на разбивания са дадени от (Кнут, 2013).

Като вземем предвид (4) и (5), то броят на всички възможни разпределения на  $w$  въпроса в точно  $k$  тестови секции е:

$$m_k = \sum_{s \in S} \frac{u!}{(u-k)!k_{s,1}!k_{s,2}! \dots k_{s,t}!} \cdot \left[ \prod_{i=1}^u C_x^v \right]_s. \quad (6)$$

Тогава вероятността изтеглените въпроси да са разпределени в точно  $k$  теми ще бъде

$$P_k = \frac{m_k}{n}. \quad (7)$$

Нека, например, имаме банка от 10 теми, всяка от които съдържа по 20 тестови въпроса. Тестът съдържа 10 въпроса, които се извличат по случаен начин от цялата съвкупност. Ще изчислим вероятността въпросите да са разпределени в седем теми. Векторите  $X_{7,s \in S}^*$  са:

$$X_{7,1}^* = (4,1,1,1,1,1,0,0,0);$$

$$X_{7,2}^* = (3,2,1,1,1,1,0,0,0);$$

$$X_{7,3}^* = (2,2,2,1,1,1,0,0,0).$$

Като използваме (1), (6) и (7), определяме:

$$n = 22\,451\,004\,309\,013\,280,$$

$$m_7 = 8\,363\,040\,000\,000\,000,$$

$$P_7 \approx 0.37250182\,151729.$$

Аналогично определяме (табл. 1)

Таблица 1

**Вероятности за неравномерно разпределение на въпроси**

Брой теми, от които са избрани въпроси ( $k$ )	Брой реализации ( $m_k$ )	Вероятност ( $P_k$ )
1 тема	1847560	0.00000000008229
2 теми	65439360720	0.00000291476318
3 теми	8742192030000	0.00038938979787
4 теми	283500257040000	0.01262750891398
5 теми	2475475002000000	0.11026121450639
6 теми	7435792728000000	0.33120089532096
7 теми	8363040000000000	0.37250182151729
8 теми	34364160000000000	0.15306290768562
9 теми	43776000000000000	0.01949845957778
10 теми	10240000000000000	0.00045610431760

Резултатите в таблицата показват, че средно 33 от 100 генерирани теста биха включвали въпроси само от 6 теми от конспекта, което би изключило 40% от изучавания материал. Въпреки малката вероятност, е възможно и всички въпроси да бъдат избрани от една и съща тема. Това налага извода, че е необходимо да се приложи механизъм, който да регулира броя избрани въпроси за всяка тема при използване на вариант А) за генериране на тест.

2. Невъзможност да бъдат обходени всички комбинации. Общият брой възможни комбинации от въпроси расте паралелно с размера на базата от данни. Изчисляването на всяка от тях би довело до огромен разход на изчислителни ресурси и време. Резултатите от изчисленията за брой комбинации на посочените примери са представени в таблица<sup>1</sup>.



Таблица 2

### Брой възможни тестове

База данни и тест	Брой комбинации
Тест от 10 въпроса избрани от 200	254.31
Тест от 100 въпроса избрани от 700	2409.63
Тест от 150 въпроса избрани от 200	2158.31
Тест от 15 въпроса избрани от 20	15504

При генериране на всяка от комбинациите следва да се извърши проверка дали сумата от теглата на въпросите в нея отговаря на зададеното изискване за обща сложност на теста. Търсенето на правилните комбинации може да бъде сравнено с дешифриране на кодирано съобщение. Така например, за разбиване на (читания за неразбиваем в миналото) хеш алгоритъм за криптиране SHA-1 са достатъчни  $2^{69}$  операции (Schneier, 2005).

Въпреки напредващите възможности на компютърния хардуер, извършването на такъв обем изчисления е трудно постижим. Разработен през 2013 г. китайски суперкомпютър извършва по 33, 86 трилиона операции в секунда. Ако се използва неговата изчислителна мощ, времето за проверка на първите три комбинации ще бъде както следва:

Таблица 3

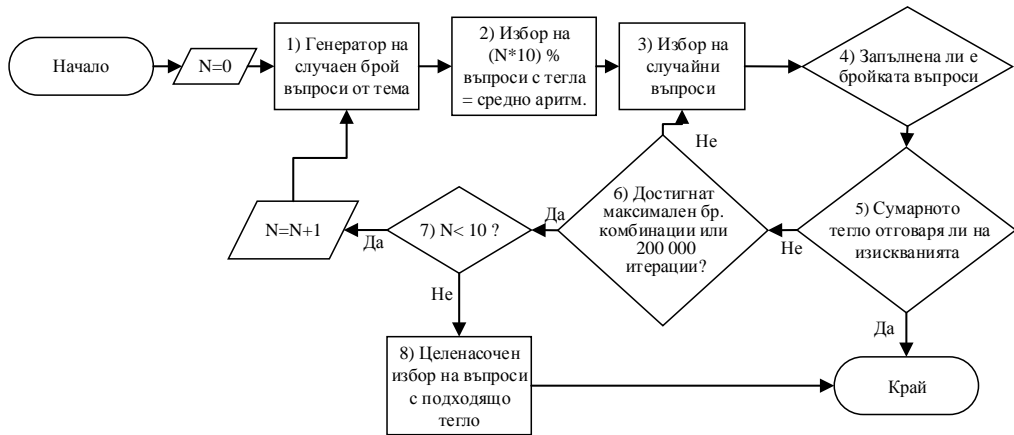
### Време за проверка на генерираните тестове

База данни и тест	Време за обработка на всички комбинации
Тест от 10 въпроса избрани от 200	11.05 минути
Тест от 100 въпроса избрани от 700	$2^{377.78}$ години
Тест от 150 въпроса избрани от 200	$2^{126.46}$ години

Невъзможността да бъдат обходени всички възможни комбинации създава проблем в случаите, в които много малка част от тях удовлетворяват зададените изисквания.

## 2. Описание и тестване на алгоритъм за генериране на компютърна тестова комбинация

В опит да бъдат разрешени посочените проблеми е създаден алгоритъм със следната структура (фиг. 2 )



**Фиг. 2. Алгоритъм за генериране на тестове**

Вероятността избраните въпроси да бъдат концентрирани в определени теми се елиминира чрез генериране на комбинация на случаен брой въпроси за всеки модул, като минимумът е 1 (стъпка 1). Поради естеството на случайния избор е възможно в базата данни за избраната тема да няма въпроси, които да имат подходящи тегла, за да удовлетворят изискванията на потребителя за обща сложност на теста. Например, ако в темата има само въпроси с тежест 3, а изискването на потребителя е за тест с максимална сложност – т.е. само въпроси с тежест 5. Въпреки това, не е направено изключване на неудовлетворимите комбинации, защото това кореспондира с проблема, свързан с невъзможността да бъдат обходени всички варианти.

За да бъде подобро бързодействието и запазена вероятността по случаен път да се избере тест, броят на операциите по проверка на комбинации се свежда до по-малкото от двете числа – 200 000 или общия брой комбинации\*3 (стъпка 6). Както беше посочено по-горе, има вероятност в избраното по случаен начин множество от брой въпроси за тема да не съществува комбинация, отговаряща на изискванията. От друга страна е възможно такава комбинация да съществува, но дори след 200 000 разбърквания тя да не бъде открита. По тази причина към алгоритъма е добавена рекурсия (стъпка 7) и реципрочен коефициент на разбъркване (РКР) – N. РКР указва какъв процент въпроси с тегла, възможно най-близки до средното, да бъдат избрани преди разбъркването. Например, при изискване за генериране на тест със средно ниво на сложност да се избират въпроси със средна трудност. В началото на изпълнението този коефици-

циент е 0, а при всяка следваща рекурсия нараства с 10%, като по този начин намалява времето за изпълнение, защото колкото повече въпроси са избрани на база своите тегла, толкова по-малко възможни комбинации остават за произволен избор.

Въпреки близо 2-та милиона операции, все пак е възможно да не бъде открит подходящ набор от въпроси, удовлетворяващи изискванията. По тази причина е създадена функция (стъпка 8), която да търси целенасочено въпроси с подходяща сложност. Механизмите на търсене на определени въпроси до известна степен са в противоречие с изискването за случаен избор, но са единствената възможност в случаи, в които общият брой комбинации е невъзможно да бъде обходен.

Справките за функциониране на системата показват, че през последната година са проведени над 1000 теста. Създаването на тест отнема средно 0,0002 секунди, а максималното време за изпълнение е близо 0,001 секунди, което демонстрира добро бързодействие на алгоритъма. Въпреки сравнително големия брой тестове, вход в рекурсия не е извършен нито един път. Тези резултати са следствие от състоянието на базата от данни и изискванията към тестовете. Към момента има изключително малък процент въпроси с тегло, различно от 3, а заявките за генериране на тестове са с голям интервал за обща сложност на теста. При такива фактори, всяка изтеглена случайна комбинация от въпроси би удовлетворила зададените параметри.

Посочените резултати не са достатъчни, за да бъде определена ефективността на работа на предложения алгоритъм и по тази причина е създаден тестов модул, който да изследва по-обстойно възможностите му в различни ситуации. Използвана е следната хардуерна конфигурация – Intel Core i3-3130M CPU @ 2.60GHz, HDD - HGST 7200RPM SATA-III 500GB, RAM - 4 GB DDR3-1600(800MHz) SDRAM. Софтуерът за изпълнение на тестовете включва ОС Windows 7 Pro 32 bit с SP1, езикът PHP 5.5.6 и СУБД MySQL Community Server 5.6.14.

Като основни параметри на тестовия модул се задават четирите случая, изброени по-горе, а именно:

- 1) генериране на тест от 10 въпроса от 200 налични в базата данни;
- 2) генериране на тест от 100 въпроса от 700 налични в базата данни;
- 3.1) генериране на тест от 150 въпроса от 200 налични в базата данни;
- 3.2) генериране на тест от 15 въпроса от 20 налични в базата данни.

Въпросите са разпределени равномерно по теми. За проверка на ефективността на системата при различни теглови коефициенти се създават пет при-

мерни бази от данни с различни коефициенти. Сложността на въпросите в тях е произволно число между – 1 и 4; 2 и 5; 1 и 5; 2 и 4; всички въпроси са със сложност 3.

За всяка от комбинациите (общо 40) на посочените видове тестове и бази от данни с теглови коефициенти се задава условие за генериране на 4000 теста в рамките на максимум 10 минути. За да се симулират разнообразни изисквания за обща сложност на теста, всяка група от 20 теста получава изисквания от възможно най-широк диапазон (при вариант 1 общата сложност да е между 10 и 50) до възможно най-малък, като увеличаването на долната граница и намаляването на горната е през 2%. По този начин всеки първи генериран тест от групата от 20 е с най-лесни изисквания, а всеки 20-ти – с най-сложни.

### 3. Резултати и заключения

Проведените тестове сочат, че средното време за генериране на един тест е 0,110 секунди при генерирани 52 118 теста.

Таблица 4

#### Резултати от проведените тестове

Тип тест	Тегла на въпросите в базата между	Среден брой итер.	Макс. брой итер.	Средно време за ген. (сек)	Макс. време за ген. (сек)	Общ брой тестове	Брой изп. на целенас. избор	Брой слушачи на вход в рек.	Среден брой рек.
3.2)	1-4	1,20	8	0,000	0,016	760	-	-	-
3.1)	1-4	70,27	987	0,660	199,475	779	-	10	0,0231
1)	1-4	2,47	9	0,0004	0,016	780	-	-	-
2)	1-4	4,35	9	0,002	0,312	780	-	-	-
3.2)	1-5	1,02	9	0,003	7,300	4000	2	2	0,01
3.1)	1-5	1,00	2	0,001	0,016	4000	-	-	-
1)	1-5	1,46	9	0,0003	0,016	4000	-	-	-
2)	1-5	1,00	1	0,001	0,016	4000	-	-	-
3.2)	2-4	1,00	1	0,000	0,016	4000	-	-	-
3.1)	2-4	1,00	1	0,001	0,016	4000	-	-	-

Ив. Куюмджиев. Разработване на алгоритъм за генериране на компютърна тестова комбинация

1)	2-4	1,17	9	0,0002	0,016	4000	-	-	-
2)	2-4	1,00	1	0,001	0,027	4000	-	-	-
3.2)	2-5	1,04	4	0,0002	0,016	240	-	-	-
3.1)	2-5	32,64	9	1,523	226,347	259	1	5	0,06
1)	2-5	2,39	9	0,0004	0,016	260	-	-	-
2)	2-5	1,43	9	0,001	0,016	260	-	-	-
3.2)	3	1,00	1	0,0002	0,003	4000	-	-	-
3.1)	3	1,00	1	0,001	0,020	4000	-	-	-
1)	3	1,00	1	0,0003	0,024	4000	-	-	-
2)	3	1,00	1	0,001	0,007	4000	-	-	-

С най-високо средно време за генериране (0,437 сек.) е тест от 150 въпроса, избрани от база от данни с 20 теми по 10 въпроса, а с най-ниско (0,0003 сек.) – тест от 10 при 10 теми по 20 въпроса. Резултатите показват, че времето за генериране на тестове е в пряка връзка с броя итерации и броя рекурсии, а те, от своя страна, зависят не само от размера на базата данни и размера на изисквания тест, а и от съответствието между изискването за обща сложност на теста и теглата на отделните въпроси в базата данни. С най-продължително средно време за изпълнение в повечето случаи са тестовете, в които въпросите в базата данни са с тегла между 2 и 5 или 1 и 4, т.е. бази от данни с по-ниска или по-висока от средната сложност. Петата база от данни, съставена само от въпроси с теглови коефициент 3, отговаря на състоянието на използваната на практика база от данни и доказва твърдението, че скоростта на изпълнение и малкият брой итерации са резултат от задаване на широки критерии за търсене и липсата на разнообразие в сложността на въпросите.

Резултатите сочат, че алгоритъмът за генериране на тестове изпълнява поставените изисквания. Работи със задоволителна скорост, както с големи, така и с малки бази от данни. Условия за подобряване на бързодействието му са задаването на набор от въпроси с разнообразна сложност и създаване на тестове с до няколко десетки на брой въпроси.

Използването на рекурсия забавя изпълнението на скрипта, но води до генериране на тест с въпроси, избрани на напълно случаен принцип. Както беше посочено, въпреки големия брой разбърквания и рекурсии, в някои случаи се налага прилагане на функцията за целенасочено търсене на въпроси с подходящи теглови коефициенти – при проведените тестовете в три от над 52 000 случая.

Използването на разработения алгоритъм подпомага дейността на тестовия център към ИУ – Варна. Компютърно генерираните тестове от разработения модул намаляват субективния фактор и съдържат справедливо и случайно разпределение на въпросите по теми. Същевременно повишават и икономичността на провеждане на изпита като намаляват:

- разходите, необходими за размножаване (копиране) на един тест;
- разхода на време при провеждане на теста;
- времето, необходимо за проверка и оценка на резултатите.

Елиминира се нуждата от използването на софтуер, разработен от външна фирма и се създава възможност за бъдеща доработка – внедряване на система за адаптивни тестове. Като друга перспектива пред тестовия център може да бъде очертана възможността за провеждане на сертификационни изпити на външни организации. Безпроблемната работа на алгоритъма през последните 12 месеца доказва, че изпълнява наложените от потребителите изисквания. Прилагането му за целите на кандидат-студентския прием и проведените тестове показват способността му да работи и с по-големи бази от данни, без това да се отрази осезаемо на времето за изпълнение.

Бих искал да изкажа благодарности на колегите, без които тази статия и описания в нея софтуер нямаше да съществуват – Георги Зеленков, Деян Михайлов, Милен Маринов, Петър Димитров.

### **Бележки**

1. Изчислени са по формулата  $C_n^k = \frac{V_n^k}{P_k}$

### **Литература**

1. България. НАОА. (2011) Критериална система за програмна акредитация на професионални направления и специалности от регулираните професии. // Министерски съвет. София.
2. Гюрова, В. и др. (2006) Приключението учебен процес. // Агенция Европрес. София.
3. Десев, Л. и др. (1977) Педагогическа психология. София: Наука и изкуство.
4. Кнут, Д. (2013) Искусство програмирования. Том 4, вып. 3. Генерация всех М. Изд. дом «Вильямс», с. 53-55.
5. Холл, М. Комбинаторика. М. (1970) Изд. „Мир” с. 45-46.

6. Eskenasi, A., Vladimirova, T., Vassileva, J. (1993) Incorporating Student Models in Adaptive Testing Systems. *Innovations in Education & Training International*. 30( 2), p. 135-142.
7. Kastner, M., B. Stangla. (2011) Multiple Choice and Constructed Response Tests: Do Test Format and Scoring Matter? *Procedia - Social and Behavioral Sciences*. (12), p. 263-273.
8. Lesage, E., M. Valcke, E. Sabbe. (2013) Scoring methods for multiple choice assessment in higher education – Is it still a matter of number right scoring or negative marking? *Studies in Educational Evaluation*. 39 (3), p. 188-193.
9. Patrick, M., E. Taylor, K. Thomas. (2008) Summary of Assessment in Higher Education and the Management of Student-Learning Data. *Academy of Management Learning & Education*. 7 (2), p. 269-278.
10. Ventouras, E., D. Triantis, P. Tsiakas, C. (2010) Stergiopoulos. Comparison of examination methods based on multiple-choice questions and constructed-response questions using personal computers. *Computers & Education*. 54 (2), p. 455-461.
11. Wainer, H., N. Dorans, D. (2014) Thissen, R. J. et al. *Computerized Adaptive Testing: A Primer* 2nd Ed. Routledge.
12. Schneier, B. (2005) SHA-1 Broken. Available from:  
[https://www.schneier.com/blog/archives/2005/02/sha1\\_broken.html](https://www.schneier.com/blog/archives/2005/02/sha1_broken.html) [Accessed 30/01/2015]
13. Worlds Fastest Supercomputer. Available from:  
<http://www.telegraph.co.uk/technology/news/10129285/Chinese-supercomputer-is-worlds-fastest-at-33860-trillion-calculations-per-second.html>[Accessed 3