

UDC 378:62

Viktor E. Bondarenko

Doctor of Technical Sciences, Associate Professor, Professor of the Department of Computer Sciences
 Kyiv International University, Kyiv, Ukraine
 ORCID ID 0000-0001-5591-1504
 victorbondarenko@ukr.net

DESIGNING OF COMPUTERIZED ADAPTIVE TESTS IN THE ABSENCE OF TESTING STATISTICS

Abstract: A Computerized Adaptive Test proposes items according to the student's knowledge level. Therefore, the number of items, which are given to students, is reduced. Besides, the ending of such test is determined by the student's knowledge level, which allows an instructor to reduce testing time. As usual, construction of such tests is based on the Item Response Theory (IRT). This theory gives models which use statistical data about the student's knowledge level and difficulty of items. We do not have such statistics for new tests. In such cases, this paper proposes to estimate the complexity of items on the basis of the experts' conclusions. These conclusions are based on the analytic hierarchy process (AHP) which was modified. The modification allows experts to estimate the complexity of items with the help of the collection of the items characteristics. This modification can remove the expert's inadequate estimates of items or their characteristics. This method allows experts to classify all items in clusters according to their complexity in the first stage of the testing when statistics of items use is absent. A test constructor, on the basis of a decision tables network, realizes the algorithm of the items' selection from different clusters. In the future, tutors will have tested a sufficient number of students' groups. They record statistics of the test using. A test constructor receives such statistics, which will allow them to use the models of the Item Response Theory for estimation of the test items' complexity.

The assessment of the knowledge level of students is made with the help of an adaptive test, which is based on a network of decision tables. This network determines the algorithm of using items from different clusters for the testing.

The adaptive test is built on the basis of the network of decision tables as a computer system. This system is constructed on the Java platform with the help of the programming environment Android Studio. It has the interface suitable for students as well as for a constructor, which allows the constructor to change the algorithm of using items if received statistics of items use shows such necessity.

Keywords: computerized adaptive test; analytic hierarchy process; experts' conclusion; decision tables; a network of decision tables.

1. INTRODUCTION

The problem statement. A Computerized Adaptive Test is a test which gives each item according to the skills of a student. The finalization of the test is determined according to the student's skills. Thus, these tests are adapted to the characteristics of each student. All the items are proposed according to the estimated student's knowledge level. As a result, the number of items which are posed to students is reduced. Thus, the test estimates the student's knowledge level faster, giving them adequate items.

The Item Response Theory (IRT) is a base for the design and analysis of tests. It allows to choose the test items and to find the finalization criterion of the test.

Unfortunately, this theory uses statistical data about the student's testing and difficulty of used items.

According to the IRT, the conditional probability of the item execution is a function of the latent variable θ (level of a student's knowledge) and the difficulty (β) of items.

The relation between a student's knowledge and the difficulty of the items is expressed by the Rasch model or more complex models as was shown [1], [2].

Variables θ and β are found from statistics. As usual, we do not have such statistics for new tests.

So, here we face the main question. How can we estimate the difficulty of the items for a new adaptive test, if statistics of items use is absent? An answer to this question was not found in the literature.

The article's goal. In view of the aforesaid, this paper describes a way of an adaptive test design without statistics of the test items use.

Such design of adaptive test is made with the help of an expert group. In the future, the test will be tested on big groups of students and we will receive statistics of its using. In such case, it is possible to correct such test according to the IRT theory.

This paper is organized as follows:

Section 1 is an introduction to the problem of adaptive testing. Section 2 has main results of the paper. Subsection 2.1 describes the general problems of the estimation of the item complexity of adaptive tests with the help of the analytic hierarchy process (AHP). Subsection 2.2 discusses building and concordance of pairwise comparisons matrices. Such matrices are used in the AHP method, which determines the complexity of test items. Besides, it describes the modification of the AHP method for classification of test items. This modification forms the concordance of matrices of the pairwise comparisons. Subsection 2.3 deals with the receiving of values of pairwise comparisons matrices with the help of a group of experts. Subsection 2.4 forms item clusters according to the item complexity. Subsection 2.5 shows the adaptive estimation of the knowledge level with the help of a network of decision tables which use items from different clusters. Subsection 2.6 describes the software implementation of the decision support system for adaptive testing. Such implementation is made on the basis of a network of decision tables. Section 3 consists of conclusion and main results of the paper.

2. THE RESULTS AND DISCUSSION

2.1. The method of estimation of item complexity

For the designing of the adaptive test, we should do the classification of items.

It is necessary to classify all items in classes according to their complexity. The complexity is defined by the set of characteristics of the items.

It is possible to use different characteristics of items for their classification. For example, we use such characteristics of items: item difficulty, item form, content validity, information capacity.

It is possible to use other characteristics, for example, the simplicity of questions formulating, inability to guess the answer and so on.

We do not have statistics on the items execution for the new tests. Therefore, such classification is done with the help of experts.

So, it is necessary to classify the n items with the help of the m characteristics. For this purpose, we should find methods which will classify the necessary items. We use, for the classification of items, the analytic hierarchy process (AHP), which was introduced by Saaty [3]. This method uses pairwise comparisons matrices of characteristics and items. Such matrices are built with the help of experts.

However, the use of the AHP meets some difficulties. Unfortunately, the AHP has the possibility of the poor concordance of pairwise comparisons matrices. Such poor concordance is determined by the expert's inadequate estimates of items or their characteristics.

Saaty [3] recommends removing the poor concordance of a pairwise comparisons matrix in such way: 1) to change experts; 2) to find more data; 3) to solve the problem by another method.

The aim of the next two sections is the modification of the AHP for the classification of items. This modification allows us to form the pairwise comparisons matrices by the experts' group that should improve the quality of these matrices. Besides, we propose the method which allows us to check the concordance of pairwise comparisons matrices.

2.2. Building and concordance of pairwise comparisons matrices

Based on the AHP, experts construct a matrix of the pairwise comparisons of characteristics. In such matrix, each row and column is labelled by the characteristics.

The matrix is filled with the help of assessment of the comparison of two characteristics. The assessment is based on a scale of importance. This scale has the values from 1 ("equal preference") to 9 ("A very significant advantage") (Table 1).

Table 1

The scale of importance

<i>Values</i>	<i>Description of importance</i>
1	Equal preference.
3	Moderate advantage.
5	Essential advantage.
7	Important advantage.
9	Very important advantage.
2,4,6,8	Intermediate values.

If the characteristic "Item difficulty" has a significant advantage over "Information capacity", we put the value 8 at the intersection of the row, labelled by the characteristic "Item difficulty" and the column, labelled by the characteristic "Information capacity".

And we will put the value 1/8 at the intersection of the column, labelled by the characteristic "Item difficulty" and the row, labelled by the characteristic "Information capacity". Thus, a group of experts forms the matrix of pairwise comparisons of the characteristics of the test. An example of such matrix is given in table 2.

Table 2

The non-concordant matrix of pairwise comparisons of test characteristics

		1	2	3	4
		Item difficulty	Form	Content validity	Information capacity
1	Item difficulty	1	3	5	8
2	Form	1/3	1	1/3	6
3	Content validity	1/5	3	1	9
4	Information capacity	1/8	1/6	1/9	1

The matrix of pairwise comparisons defines relations between characteristics of items or between items. For such relations transitivity of the matrix values should be performed.

The transitivity can check the logic of the expert thinking. The transitivity means for factors (characteristics of items or items) that if factor A exceeds the factor B and the factor B, exceeds the factor C, then the factor A should exceed the factor C. It is performed inequality $A > B > C$, where the symbol ">" means exceed. Also, the numerical evaluation of the transitivity of relations should be performed.

Let the factor A be 2 times more than factor B and the factor B 3 times more than the factor C. From this, it follows that the factor A should be $m = 2 \times 3 = 6$ times more than the factor C.

Lack of concordance can be a serious limiting factor for the use of the AHP method.

For investigating the transitivity of relations it is proposed to use oriented graphs. The pairwise comparisons matrix is rated as the adjacency matrix of the graph. The values of the matrix will be interpreted as the weight of the edges of the graph.

The concordance of the pairwise comparisons matrix demands the same weight from all paths between two nodes of the graph.

The node $w \in V$ attainable from the node $v \in V$, if $v = w$, or graph $G = (V, E)$ has a path from v to w , where the V is a set of nodes, the E is a set of edges (relations) between nodes.

The matrix of pairwise comparisons is constructed in such manner that does not need to use the relations for which $a_{ij} < 1$ at $i \neq j$, and $a_{ii} = 1$ for the adjacency matrix of the graph.

The matrix of pairwise comparisons, which is presented in table 2 shows some characteristics that characterize the items. We use this matrix as the adjacency matrix of the graph to analyze the concordance (Fig.1,a.). To simplify the picture, branches which have the weight less than or equal to 1 do not show on the graph.

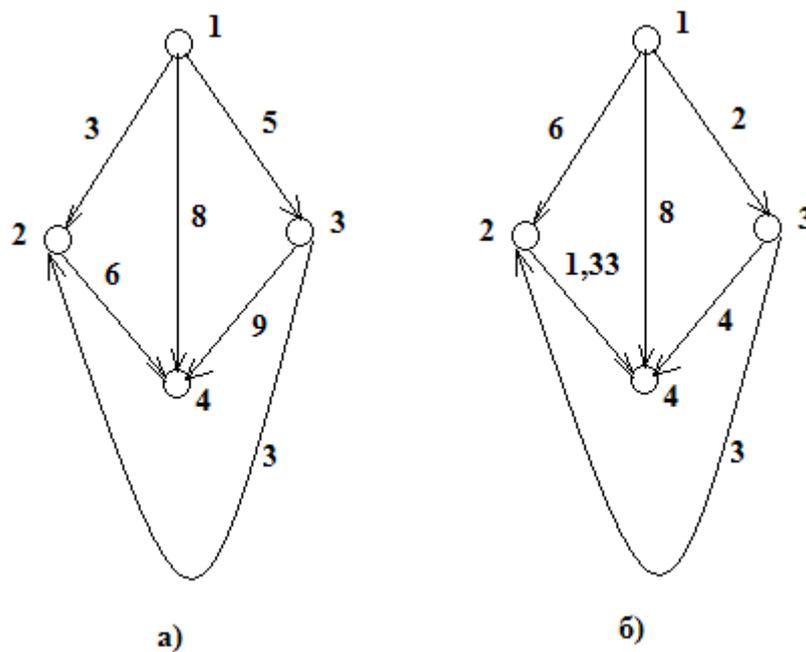


Figure 1. The graph of the matrix of pairwise comparisons. a) The non-concordant matrix of pairwise comparisons. b) The concordant matrix of pairwise comparisons

The analysis of the graph (Figure 1,a) shows that the matrix of pairwise comparisons (Table 2) is non-concordance.

The way 1-2-4 has the weight of 18 and the way 1-4 has the weight of 8. Way 1-3-2 has the weight of 15 and way 1-2 has the weight of 3. The way 1-3-4 has the weight of 45 and the way 1-4 has the weight 8.

Figure 1,b shows the graph which is based on the work of experts in the concordance of the matrix of the pairwise comparisons. The concordance of the matrix of the pairwise comparisons is presented in table 3.

Table 3

The concordant matrix of the pairwise comparisons

	Subject difficulty	Form	Content validity	Information capacity
Subject difficulty	1	6	2	8
Form	1/6	1	1/3	1.33
Content validity	1/2	3	1	4
Information capacity	1/8	1/1.33	1/4	1

The concordance of a matrix of pairwise comparisons is performed by a group of experts in the iterative mode.

2.3. The receiving of values of pairwise comparisons matrices with the help of a group of experts

The AHP method uses matrixes of pairwise comparisons of the characteristics of items and, in the same way, the pairwise comparisons of the items.

It is possible to use a group of experts to improve the quality of matrices of pairwise comparisons. The method of such evaluation is given below.

A group of the m experts estimates the pairwise comparison (a_{ij}) of the i th and the j th characteristics of the items.

Similarly, a group of the m experts estimates the pairwise comparison of the i th and the j th items. It is performed with the help of the following algorithm. It is possible to use T stages of assessment with the aim of improving its quality.

Algorithm 1.

1. Evaluation of the a_{ij} is computed as a weighted average, according to the formula

$$a_{ij} = \frac{\sum_{k=1}^m \rho_k S_{ij}^k}{\sum_{k=1}^m \rho_k},$$

where ρ_k is the coefficient of the confidence of the k th expert. On the first stage of estimation, the coefficients of the confidence of all experts are the same and equal to 1. S_{ij}^k is the evaluation of the pairwise comparison of the i th and the j th characteristics or items. Such evaluation is defined by the k th expert. m is the number of experts.

Each expert makes the assessment of the pairwise comparisons based on a scale of the importance. The values of this scale are defined in the range from 1 to 9 (see table 1).

2. Calculate the coefficient of confidence of the k th expert by the formula

$$\rho_k = \frac{1}{T} \sum_{t=1}^T \rho_k^t,$$

where T is the number of stages of assessment, ρ_k^t is the coefficient of confidence of the k^{th} expert in the t^{th} stage of the assessment;

$$\rho_k^t = \exp\left(-\frac{(G_{ij} - S_{ij}^k)^2}{2\sigma_k}\right),$$

G_{ij} is the estimation of the i^{th} characteristic or item in the j^{th} measure. This estimation is defined after testing the matrix of pairwise comparisons from the point of view its concordance. σ_k is the coefficient of forgetting of the k^{th} expert.

3. A test of the concordance of a pairwise comparisons matrix a_{ij} and its correction by experts. As a result, we receive the new pairwise comparisons matrix with elements a_{ij}^{new} .

4. Steps 1-3 are repeated until the change of elements a_{ij} after corrections is lower than the predetermined value ε .

$$\sum_{i=1}^m \sum_{j=1}^m \left| a_{ij}^{\text{new}} - a_{ij} \right|^2 < \varepsilon.$$

2.4. The forming of item clusters according to the item complexity

This section describes the forming of item clusters with the help of the AHP method.

Step 1. According to the AHP method, we calculate the vector Y of local priorities as follows:

1. Calculate the elements R_i of the vector R .

$$R_i = (W_{i1} \times W_{i2} \times W_{i3} \dots W_{in})^{1/n},$$

where W_{ij} are characteristics of the items the i^{th} row and the j^{th} column of pairwise comparisons matrix, n is the size of the matrix.

In our example $n=4$. The elements R_i of the vector R of the characteristics of the items of a test is calculated as the expression.

$$R_i = (W_{i1} \times W_{i2} \times W_{i3} \times W_{i4})^{1/4}.$$

After substituting the numerical values from table 3 we get:

$$\begin{aligned} R_1 &= (1 \times 6 \times 2 \times 8)^{1/4} = 3.13, \\ R_2 &= (1/6 \times 1 \times 1/3 \times 1/1.33)^{1/4} = 0.45, \\ R_3 &= (1/2 \times 3 \times 1 \times 4)^{1/4} = 1.56, \\ R_4 &= (1/8 \times 1/1.33 \times 1/4 \times 1)^{1/4} = 0.39. \end{aligned}$$

2. Calculation of the components X_i of the normalized vector X of the characteristics of the items with the help of such expression.

$$X_i = \frac{R_i}{\sum_{i=1}^n R_i},$$

where n is the number of rows of the matrix of pairwise comparisons.

We find the components X_i of the normalized vector X after substituting the numerical values.

$$\begin{aligned} X_1 &= 3.13/(0.313+0.45+1.56+0.39) = 1.15, \\ X_2 &= 0.45/(0.313+0.45+1.56+0.39) = 0.17, \\ X_3 &= 1.56/(0.313+0.45+1.56+0.39) = 0.58, \\ X_4 &= 0.39/(0.313+0.45+1.56+0.39) = 0.14. \end{aligned}$$

3. Computation of the local priorities vector Y of characteristics of items is made with the help of such expression.

$$\begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{pmatrix} = \begin{pmatrix} W_{11} & W_{12} & W_{13} & W_{14} \\ W_{21} & W_{22} & W_{23} & W_{24} \\ W_{31} & W_{32} & W_{33} & W_{34} \\ W_{41} & W_{42} & W_{43} & W_{44} \end{pmatrix} \times \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{pmatrix} .$$

We obtain a vector of local priorities Y for characteristics of the items after substituting the numerical values.

$$\begin{pmatrix} 4.46 \\ 0.74 \\ 2.23 \\ 0.56 \end{pmatrix} = \begin{pmatrix} 1 & 6 & 2 & 8 \\ 1/6 & 1 & 1/3 & 1.33 \\ 1/2 & 3 & 1 & 4 \\ 1/8 & 1/1.33 & 1/4 & 1 \end{pmatrix} \times \begin{pmatrix} 1.15 \\ 0.17 \\ 0.58 \\ 0.14 \end{pmatrix} .$$

Step 2. The matrix of pairwise comparisons of the test items (A, B, C, D, E) is based on expert assessments. The matrix is filled with values estimated pairwise comparisons of the items. Such matrix should be built for each of the characteristics. The filling of the matrix is the same as in points 1-3 of step 1.

For each matrix, we build the vector of local priorities as in step 1. For our task, four such matrices will be built.

We determine the vector of local priorities for test items based on each of the characteristics as Z_{ij} , where i is the number of the characteristic of the item, j is the number of the item of the test. These vectors are summarized in tables 4-7.

Table 4

The vector of local priorities for items based on the parameter "Subject difficulty"

Subject difficulty	A	B	C	D	E	R_i	X_i	Z_{1j}
A	1	2	1	4	2	1.74	0.31	1.54
B	1/2	1	1/2	2	1	0.87	0.15	0.77
C	1	2	1	4	2	1.74	0.31	1.54
D	1/4	1/2	1/4	1	1/2	0.43	0.076	0.38
E	1/2	1	1/2	2	1	0.87	0.15	0.77

Table 5

The vector of local priorities for items based on the parameter "Form"

Form	A	B	C	D	E	R_i	X_i	Z_{2j}
A	1	2	2	2	2	1.74	0.33	1.17
B	1/2	1	1	1	1	0.87	0.17	0.83
C	1/2	1	1	1	1	0.87	0.17	0.83
D	1/2	1	1	1	1	0.87	0.17	0.83
E	1/2	1	1	1	1	0.87	0.17	0.83

Table 6

The vector of local priorities for items based on the parameter "Content validity"

Content validity	A	B	C	D	E	R_i	X_i	Z_{3j}
A	1	1	1	1	1	1	0.2	1
B	1	1	1	1	1	1	0.2	1
C	1	1	1	1	1	1	0.2	1
D	1	1	1	1	1	1	0.2	1
E	1	1	1	1	1	1	0.2	1

Table 7

The vector of local priorities for items based on the parameter "Information capacity"

Information capacity	A	B	C	D	E	R_i	X_i	Z_{4j}
A	1	4	8	4	2	3.03	0.47	2.42
B	1/4	1	2	2	1	1	0.15	0.82
C	1/8	1/2	1	1/2	1/4	0.38	0.058	0.30
D	1/4	1	2	1	1/2	0.76	0.12	0.61
E	1/2	1	4	2	1	1.32	0.20	1.06

Step 3: Calculation of the vector P of the global priority of items is performed with the help of such expression.

$$\begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{pmatrix} = \begin{pmatrix} Z_{11} & Z_{21} & Z_{31} & Z_{41} \\ Z_{12} & Z_{22} & Z_{32} & Z_{42} \\ Z_{13} & Z_{23} & Z_{33} & Z_{43} \\ Z_{14} & Z_{24} & Z_{34} & Z_{44} \\ Z_{15} & Z_{25} & Z_{35} & Z_{45} \end{pmatrix} \times \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{pmatrix}$$

We get the vector of global priority of items (P) after substituting the numerical values.

$$\begin{pmatrix} 19.05 \\ 14.47 \\ 17.61 \\ 12.61 \\ 14.61 \end{pmatrix} = \begin{pmatrix} 1.54 & 1.17 & 1 & 2.42 \\ 0.77 & 0.83 & 1 & 0.82 \\ 1.54 & 0.83 & 1 & 0.3 \\ 0.38 & 0.83 & 1 & 0.61 \\ 0.77 & 0.83 & 1 & 1.06 \end{pmatrix} \times \begin{pmatrix} 4.46 \\ 0.74 \\ 2.23 \\ 0.56 \end{pmatrix}$$

With the help of the vector P experts can build several clusters of items of different complexity.

For example, in our case the experts construct two clusters. The first cluster consists of the items, for which the values of the vector P do not exceed 15. The second cluster includes items, for which the values of the vector P are greater than 15. In this case, the first cluster consists of the items B, D, E, and the second cluster consists of the items A, C.

2.5. Adaptive estimation of the knowledge level with the help of a network of decision tables

The implementation of different clusters of items in the test is based on a decision support system. Such system helps to choose the strategy of the items use.

The decision support system is based on the methods which allow checking students' knowledge. Such evaluation will be made with the help of needed items. The method based on the use of decision tables is often used to design decision support systems.

The programming language which uses decision tables was proposed by Lombardi [4]. Further development of the use of decision tables is presented in [5], [6]. The decision table is a model of the decision-making process. It is a means of the compact description of logical rules that are used to define actions.

These actions are performed by the conditions for resolving problems.

Decision tables help to understand the described problems. The use of decision tables usually requires no explanation. They are formulated in the form accessible to perception and understandable to all.

Interpretation of decision tables proposed in [5], [6] restricts their potential. These decision tables do not allow the binding subset of conditions with the conjunction "or". The modification of decision tables allows to link conditions with the conjunction "or". Such decision tables have more adaptable possibility to describe the item use.

Besides, we develop theoretical principles that allow constructing networks from decision tables.

A decision table is a means of the description of logical rules. Such logical rules are used to define actions which are performed by the conditions.

A standard decision table has the form which is shown in Figure 2.

Condition 1	Yes	No	Yes	No
Condition 2	No	Yes	No	Yes
...	...			
Condition n	Yes	Ind	No	Yes
Action 1	*		*	
Action 2	*			*
...			*	*
Action m		*	*	

Figure 2. General view of the decision table

In the upper left quarter of this table, we have conditions. The truth of these conditions is compared with the truth of each of Boolean vectors that are in the upper right quarter of the table. The elements of a Boolean vector can have three values - "Yes", "No", "Indifferently".

If the truth of a Boolean vector coincides with the truth of a set of conditions, then the subset of actions that are in the bottom left quarter of the table are performed. The symbols "*" of a vector-pointer from the right lower quarter of the table show actions which should be executed. This vector-pointer is situated under the Boolean vector, which has the same truth as the set of conditions.

Actions that are in the bottom left quarter of the table can be complex and include expert systems, optimization problems, simulations and other methods.

The paper proposes the extension of decision table that allows binding conditions with the conjunction "or".

In the case of this modification, the table will look similar to the table presented in Figure 3.

As we can see, the decision table (Figure 3) has a subset of the set of conditions consisting of 2, 3, 4 and conditions 6,7, which are in the same cells. Such conditions are bound by the conjunction "or".

When at least one condition of a subset is true, the whole a subset is true. A subset of the conditions is not true if all the conditions of a subset are not true. It should be noted that the elements of a Boolean vector can be, depending on the needs, fuzzy, many-valued and have other features that are necessary to describe the system. In the general case, the truth of conditions can use the degree of confidence, which is the number of the interval [0,1], where 0 is complete distrust, 1 is complete confidence.

Condition 1	Yes	No	Yes	No
Condition 2 Condition 3 Condition 4	No	Yes	No	Yes
Condition 5	Yes			
Condition 6 Condition 7	Yes			
...	...			
Condition n	Yes	Ind	No	Yes
Action 1	*		*	
Action 2	*			*
...			*	*
Action m		*	*	

Figure 3. General view of the modified decision table

Thus, based on the foregoing, decision table T can be represented as a set

$$T = \langle A, B, P, V, \varphi: I(\mathcal{P}(A)) \rightarrow P, S \subset P \times V, F: S \rightarrow \mathcal{P}(B) \rangle,$$

where A is a set of conditions, B is a set of actions, P is a set of Boolean vectors, V is a set of vector-pointers, $\varphi: I(\mathcal{P}(A)) \rightarrow P$ is the surjection reflecting the truth of all subsets $\square(A)$ of the set of conditions A on the set of Boolean vector P , $S \subset P \times V$ is a set of related pairs of Boolean vectors and vector-pointers, $F: S \rightarrow \mathcal{P}(B)$ is the injective mapping of related pairs "Boolean vector – vector-pointer" in the set of all subsets $\mathcal{P}(B)$ of the actions B .

Modification of decision tables allows us to have subset $U \subset A$ of the set of conditions A . Such subset U conditions are connected by conjunctions "OR".

For such subsets you can answer "Yes" (true), if at least one condition of this subset is true, "no" (false), when none of the conditions of a subset is true, "It does not matter" (indifferent), if the truth of all conditions of a subset does not matter.

Hence, we have:

$$\exists(u \in U \subset A) I(u) = true \rightarrow I(U \subset A) = true;$$

$$\forall(u \in U \subset A) I(u) = false(indifferent) \rightarrow I(U \subset A) = false(indifferent).$$

It is easy to use networks of decision tables for decision-making. We will use such network for test management.

The network of decisions tables M is defined as follows:

$$M = \langle \{T\}, \{B(T)\}, \psi: \{T\} \times \mathcal{P}(B(T)) \rightarrow \mathcal{P}(T) \rangle,$$

where $\{T\}$ is the set of decision tables, $\{B(T)\}$ is the set of actions of the decision table T , $\psi: \{T\} \times \mathcal{P}(B(T)) \rightarrow \mathcal{P}(T)$ is the surjection which displays all pairs "decision table - a subset of actions of decision table T " in a subset of decision tables for the execution of which it is possible to go from the decision table T .

We will consider some useful operations with a network of decision tables. The first operation is the insertion of a new decision table into a network. The second operation is the removing of a decision table from a network.

Let us assume that the decision table T_j must be inserted in the network M so that the transition to the table T_j is caused by the action $b \in B(T_k)$ of the table T_k . In other words, we perform the operation $\{T\} = \{T\} \cup \{T_j\}$. And we add to the set of actions of the table T_k the action that causes the transition to the table T_j . In other words, we perform the operation $B(T_k) = B(T_k) \cup \{\text{goto } T_j\}$. Besides, it is necessary to include a new Boolean vector $p \in P$ and new vector-pointer $v \in V$ which points to the execution of the action $\{\text{goto } T_j\}$.

Suppose we need to delete from the network M the decision table T_j , which is caused by the action of the table T_k . Then you need to remove from the network M the decision table T_j . In other words, we perform the operation $\{T\} = \{T\} \setminus \{T_j\}$. And we remove from the set of actions of table T_k action that causes the transition to the decision table T_j . In other words, we perform the operation $B(T_k) = B(T_k) \setminus \{\text{goto } T_j\}$. It is necessary to remove the Boolean vector $p \in P$ and vector-pointer $v \in V$ which points to the execution of the action $\{\text{goto } T_j\}$.

For example, we consider such simple adaptive test construction. Suppose we have simple items B, D, E and items of high complexity A, C, identified in the previous section. Let the constructor, that constructs the test, use such logic. If a student did not execute simple items B and E, there is no need to give them the item D and the more complex items A, C.

A student receives the rating "not satisfactory" (2) and stops testing. If the student has completed the item B or E, they receive the item D. If the student executes the item B, they

receive the item D. If they do not fulfill it, the student receives a rating of "not satisfactory" (2). The testing stops.

If a student does not execute the item B and executes the item E, the student gets the item D. If they execute this item, they receive the rating of "satisfactory" (3).

If a student does not execute item D, they receive the estimation "unsatisfactory" (2). The testing is terminated.

If the student has completed the items B, D, E, they receive the item A. And if they do not execute it, a student is graded "good" (4). The testing stops.

If the student has completed the items B, D, E, A, they receive the item C. If they execute it, the student receives the mark "excellent" (5). If the student does not execute the item C, they receive the mark "good" (4) and the testing stops.

The schematic representation of the network is presented in Figure 4. The decision tables of the network have such elements: conditions of the items B, D, E, A, C execution; actions of the assessment of the testing (2, 3, 4, 5); the actions of the transition to another table goto <table name>; Boolean vectors that define the execution of the items.

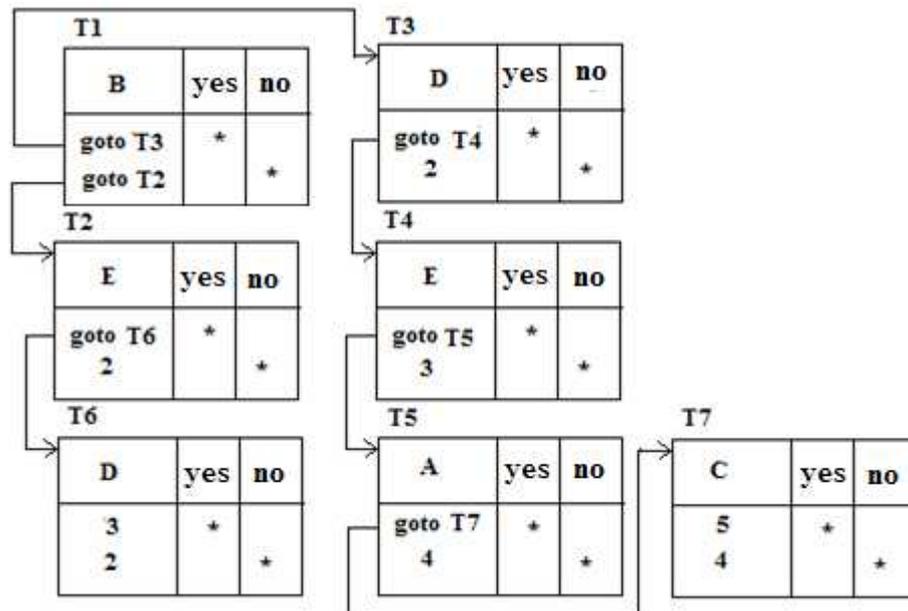


Figure 4. The example of the network of decision tables for testing

2.6. The algorithm and software for implementation of the decision support system

The process of working with the system can be divided into the following stages:

1. The group of experts creates and coordinates a list of characteristics of the items. Further, experts together conduct the pairwise comparisons of characteristics (Algorithm 1. Subsection 2.3). Next, they form the matrix of pairwise comparisons of characteristics (Subsection 2.2).

2. The experts mutually carry out the concordance of the matrix of pairwise comparisons of characteristics (Subsection 2.2).

3. The second phase of the process is creating a list of items. The next step is the joint execution by experts of pairwise comparisons of items for each characteristic (Algorithm 1. Subsection 2.3). They construct matrices of pairwise comparisons of items (Subsection 2.2).

4. The experts carry out the concordance of the matrices of pairwise comparisons of items for each of the characteristics (Subsection 2.2).

5. The last step is the calculation of ranking of items, which are based on their characteristics. Based on ranking of items, experts form the clusters of items (Subsection 2.4).

6. The design of the network of decision tables for the testing (Subsection 2.5).

The experimental variant of decision support system was developed on the basis of the network of the decision tables. It was worked out in the programming environment Android Studio.

The pilot operation of the system showed that the interface of the system is clear for the users. The system can be recommended for decision support in the construction of the adaptive tests. But it is a most versatile tool and can be used for decision-making in other areas of human activity.

3. CONCLUSION AND PROSPECTS FOR FURTHER RESEARCH

1. In the paper, the method of design of adaptive tests is presented. This method allows us to estimate the complexity of items of a new test when we do not have the statistics of its use. The items complexity of such tests is evaluated on the basis of modification of the analytic hierarchy process (AHP). This modification includes:

A) The calculation of the complexity of items based on the assessment of pairwise comparisons matrices by the group of experts. Evaluation of the items and their characteristics is calculated as a weighted average using a confidence factor for each expert. The confidence factor of an expert is based on the quality of their estimations in the past.

B) The analysis of graphs can execute the concordance of pairwise comparisons matrices. In that way, it is possible to verify transitivity of matrix relations. This verification allows analyzing and adjusting the logic of the experts' thinking.

2. The use of the modification of the AHP can estimate classes of items for adaptive testing.

3. The method of decision-making relative to using of test items is developed as a network of decision tables. Modification of decision tables is proposed. This modification allows us to include conditions, bound by the conjunction "or", in a decision table.

4. The proposed formalizing of the network of decision tables and operations on this network:

A) The inclusion of a new decision table in the network;

B) The extracting of a decision table from the network.

5. The adaptive test as a decision support system is built on the basis of the network of modified decision tables. This system is constructed on the Java platform with the help of the programming environment Android Studio.

6. The use of the system showed that it can be recommended for decision support in training. But it is a most versatile tool. And it can be used for decision-making in some other areas of human activity. For example, in banks reliability analysis or firms' analysis in public tender offers.

7. Dissemination of the described decision-making methods in analysis of firms, which participate in public tender offers, is a direction for further research.

REFERENCES (TRANSLATED AND TRANSLITERATED)

- [1] L. Crocker, J. Algina, Introduction to Classical and Modern Test Theory. Mason, Ohio, USA, Cengage Learning, 527 p, 2006. (in English).
- [2] Embretson S.E., Reise S.P. Item response theory for psychologists. London, LEA.370 p. 2013. (in English).
- [3] T. L. Saaty, Axiomatic foundation of the analytic hierarchy process. Management Science, Vol. 32, №7: p. 841-855, 1986. (in English).
- [4] L.A. Lombardi, A general business oriented language based on decision expression. Communications of the ACM, 7 (2): p. 104-111, 1964. (in English).
- [5] E. Humby, Programs from decision tables. London, Macdonald and Co.; New York, American Elsevier, 91 p., 1973. (in English).
- [6] R.A.Karayev, N.Y.Sadikhova, Production-Tabular Knowledge. Bases Tools for Assessing and Checking of Correctness, Middle-East Journal of Scientific Research, 21 (9): p.1659-1662, 2014. (in English).

The text of the article was accepted by Editorial Team 01.08.2018

ПРОЄКТУВАННЯ КОМП'ЮТЕРИЗОВАНИХ АДАПТИВНИХ ТЕСТІВ ЗА ВІДСУТНОСТІ СТАТИСТИКИ ТЕСТУВАННЯ

Бондаренко Віктор Євгенович

доктор технічних наук, доцент, професор кафедри комп'ютерних наук

Київський міжнародний університет, м. Київ, Україна

ORCID ID 0000-0001-5591-1504

victorbondarenko@ukr.net

Анотація. Комп'ютерний адаптивний тест пропонує завдання відповідно до рівня знань студента. Тому кількість завдань, які надаються студентам для тестування, зменшується. Крім того, закінчення такого тесту визначається рівнем знань студента, що дозволяє інструкторові скоротити час тестування. Зазвичай побудова таких тестів заснована на теорії педагогічних вимірювань (Item Response Theory - IRT). Ця теорія пропонує моделі, у яких використовуються статистичні дані про рівень знань студента і складність завдань.

Для нових тестів такої статистики немає. У таких випадках у роботі пропонується оцінити складність завдань на основі висновків експертів. Ці висновки засновані на методі аналізу ієрархій (Analytic Hierarchy Process - АНП), який був модифікований. Ця модифікація допоможе усунути некоректні оцінки експертами завдань і їх характеристик. Метод АНП дозволяє класифікувати всі завдання на кластери відповідно до їх складності на першому етапі тестування, коли статистика використання завдань відсутня. У майбутньому, коли буде протестовано велику кількість студентських груп, отримані статистичні дані тестування дозволять використовувати моделі IRT-теорії для уточнення оцінки складності тестових завдань.

Конструктор тесту, на основі запропонованої в роботі мережі таблиць рішень, формує алгоритм вибору завдань для тестування з різних кластерів складності.

Оцінка рівня знань студентів проводиться за допомогою програмно реалізованого адаптивного тесту, заснованого на мережі таблиць рішень.

Адаптивний тест як система підтримки прийняття рішень реалізований на платформі Java за допомогою середовища програмування Android Studio. Він має зручний як для студента, так і для конструктора інтерфейс, який дозволяє конструкторові змінювати алгоритм використання завдань, якщо отримана статистика використання завдань тесту в процесі тестування покаже таку необхідність.

Ключові слова: комп'ютеризований адаптивний тест; метод аналізу ієрархій; висновок експертів; таблиці рішень; мережа таблиць рішень.

ПРОЕКТИРОВАНИЕ КОМПЬЮТЕРИЗИРОВАННЫХ АДАПТИВНЫХ ТЕСТОВ ПРИ ОТСУТСТВИИ СТАТИСТИКИ ТЕСТИРОВАНИЯ

Бондаренко Виктор Евгеньевич

доктор технических наук, доцент, профессор кафедры компьютерных наук

Киевский международный университет, г. Киев, Украина

ORCID ID 0000-0001-5591-1504

victorbondarenko@ukr.net

Аннотация. Компьютерный адаптивный тест предлагает задания в соответствии с уровнем знаний студента. Поэтому количество заданий, которые предоставляются студентам для тестирования, уменьшается. Кроме того, окончание такого теста определяется уровнем знаний студента, что позволяет инструктору сократить время тестирования. Обычно построение таких тестов основано на теории педагогических измерений (Item Response Theory - IRT). Эта теория предлагает модели, в которых используются статистические данные об уровне знаний студента и сложности заданий.

Для новых тестов такой статистики нет. В таких случаях в работе предлагается оценить сложность заданий на основе выводов экспертов. Эти выводы основаны на методе анализа иерархий (Analytic Hierarchy Process - АНП), который был модифицирован. Эта модификация поможет устранить некорректные оценки экспертами заданий и их характеристик. Метод АНП позволяет классифицировать все задания на кластеры в соответствии с их сложностью на первом этапе тестирования, когда статистика используемых заданий отсутствует. В будущем, когда будет протестировано множество студенческих групп, полученные статистические данные тестирования позволят использовать модели IRT-теории для уточнения оценки сложности тестовых заданий.

Конструктор теста, на основе, предложенной в работе сети таблиц решений, формирует алгоритм выбора заданий для тестирования из разных кластеров сложности.

Оценка уровня знаний студентов производится с помощью программно реализованного адаптивного теста, основанного на сети таблиц решений.

Адаптивный тест как система поддержки принятия решений реализован на платформе Java с помощью среды программирования Android Studio. Он имеет удобный как для студента, так и для конструктора интерфейс, который позволяет конструктору изменять алгоритм использования заданий, если полученная статистика использования заданий теста в процессе тестирования покажет такую необходимость.

Ключевые слова: компьютеризированный адаптивный тест; метод анализа иерархий; заключение экспертов; таблицы решений; сеть таблиц решений.

