

A. CHUKHRAY, O. HAVRYLENKO*National Aerospace University “Kharkiv Aviation Institute”, Ukraine*

THE ENGINEERING SKILLS TRAINING PROCESS MODELING USING DYNAMIC BAYESIAN NETS

The **subject** of research in the article is the process of intelligent computer training in engineering skills. The **aim** is to model the process of teaching engineering skills in intelligent computer training programs through dynamic Bayesian networks. **Objectives:** To propose an approach to modeling the process of teaching engineering skills. To assess the student competence level by considering the algorithms development skills in engineering tasks and the algorithms implementation ability. To create a dynamic Bayesian network structure for the learning process. To select values for conditional probability tables. To solve the problems of filtering, forecasting, and retrospective analysis. To simulate the developed dynamic Bayesian network using a special Genie 2.0-environment. The **methods** used are probability theory and inference methods in Bayesian networks. The following **results** are obtained: the development of a dynamic Bayesian network for the educational process based on the solution of engineering problems is presented. Mathematical calculations for probabilistic inference problems such as filtering, forecasting, and smoothing are considered. The solution of the filtering problem makes it possible to assess the current level of the student's competence after obtaining the latest probabilities of the development of the algorithm and its numerical calculations of the task. The probability distribution of the learning process model is predicted. The number of additional iterations required to achieve the required competence level was estimated. The retrospective analysis allows getting a smoothed assessment of the competence level, which was obtained after the task's previous instance completion and after the computation of new additional probabilities characterizing the two checkpoints implementation. The solution of the described probabilistic inference problems makes it possible to provide correct information about the learning process for intelligent computer training systems. It helps to get proper feedback and to track the student's competence level. The developed technique of the kernel of probabilistic inference can be used as the decision-making model basis for an automated training process. The **scientific novelty** lies in the fact that dynamic Bayesian networks are applied to a new class of problems related to the simulation of engineering skills training in the process of performing algorithmic tasks.

Keywords: dynamic Bayesian network; modeling; engineering skills; forecasting; filtering; smoothing.

Introduction

Intelligent computer tutoring based on the adaptive algorithms for effective individual studying is a state of art approach to both distance and ordinary learning in the complex engineering domains. Adaptability of such programs is achieved due to advanced decision making algorithms for next task and next step choices [1]. These algorithms must implement basic pedagogical principles and give a feedback based upon adequate information about learning process [2-4]. Current research proposes to utilize dynamic Bayesian network (DBN) for tutoring process modeling. To evaluate the model variables it is offered to adapt probabilistic inference methods.

This is not the first time DBN has been used for ITS purposes. For example, in publication [5] DBNs represent several skills of students together within one model and an optimization algorithm with constraints is introduced for parametric training of such models. The prediction accuracy of the approach is evaluated on five

large-scale datasets in various fields of study such as mathematics, orthography, and physics.

An article [6] proposes an intelligent tutoring system that draws up a curriculum of prompts and tasks to teach the student certain skills. A template is provided for constructing a multilayer dynamic Bayesian network to model this problem and describe how to find out the model parameters from the data. Testing was carried out in the domains of finite field arithmetic and artificial language.

Formulation of the problem

In difference to the approaches described above, this article focuses on BSN based modelling research deal with engineering skills training. The issue of teaching engineering skills, concretely, the implementation of algorithmic tasks, when it is necessary not only to obtain an answer for specific problems, but also to compose an algorithm for its solution, has not been consid-

ered. In addition, the consequences of the modeling results, namely, the solution of forecasting, filtering, smoothing problems, are not provided.

Therefore, to raise efficiency of existed methods and to adapt them to the high-qualified engineers' tutoring following problems should be solved:

1. To propose a general approach to modeling of engineering skills occupying process.
2. To evaluate level of the student competence by considering of task solving algorithm knowledge and ability to apply it.
3. To create the structure of dynamic Bayesian network of tutoring process.
4. To develop conditional probabilities tables.
5. To solve the problems of filtering, prediction and smoothing.
6. To perform modeling of the developed DBN using special editor Genie 2.0.

1. Tutoring process model development

A tutoring process is considered as sequential solving by student tasks of a certain class with corresponding response on each of his actions. DBN of learning process model includes nodes of current competence level evaluation and nodes of decision-making on pedagogical intervention for student's competence level improvement [7].

Decisions on a kind of pedagogical actions are made after passing each of two breakpoints: algorithm construction for task solving and numerical task solution based on it. Competence level and providing hints for student on the current stage influences on competence level evaluation on the next stage. Thus, developing DBN includes two conditional connections of the first order: between nodes of student competence level, and also between node of providing hint and competence node.

The structure of created DBN in collapsed and expanded forms is shown in fig.1, where t is a time moment number.

"Skills Level" is general node for simulation of student competence level. This node has four discrete conditions: Excellent, Good, Satisfactory and Bad. On a current stage competence level influences the skill of algorithm development ("Algorithm Created" node). In its turn, the skill of algorithm development (ability to reproduce it) influences on ability to use it (node "Answer Calculated"). Two last mentioned nodes are characterized by three inner states: WithoutError (solution without errors), WithOneError (solution has one error) and WithMoreThanOneError (solution has more than one error). Depending on success of algorithm developing and the numerical solution of a task, the decision about providing hint to the student is made ("Hint" node). "Hint" node has two discrete states: Yes (hint is provided) and No (there are no any hints). Competence level of "Skills Level" and fact of hint showing at the current moment determine competence level at the next moment.

Each node has conditional probabilities table (CPT). CPT for "Skills Level" is shown in fig. 2:

$$P(\text{SkL}_t = X) = F(P(\text{SkL}_{t-1} = Y), P(\text{Hint}_{t-1} = Z)).$$

Initially there is complete uncertainty of competence level:

$$\begin{aligned} P(\text{SkL}_0 = \text{Excellent}) &= P(\text{SkL}_0 = \text{Good}) = \\ &= P(\text{SkL}_0 = \text{Satisfactory}) = P(\text{SkL}_0 = \text{Bad}) = 0.25. \end{aligned}$$

For relations $\text{SkL}_t > \text{AlgC}_t$ and $\text{AlgC}_t > \text{AnsC}_t$ corresponding CPTs are presented in fig. 3.

For relation $\text{AlgC}_t, \text{AnsC}_t \rightarrow \text{Hint}_t$ CPT is determined and defines hint absence when both breakpoints were passed without errors (fig. 4).

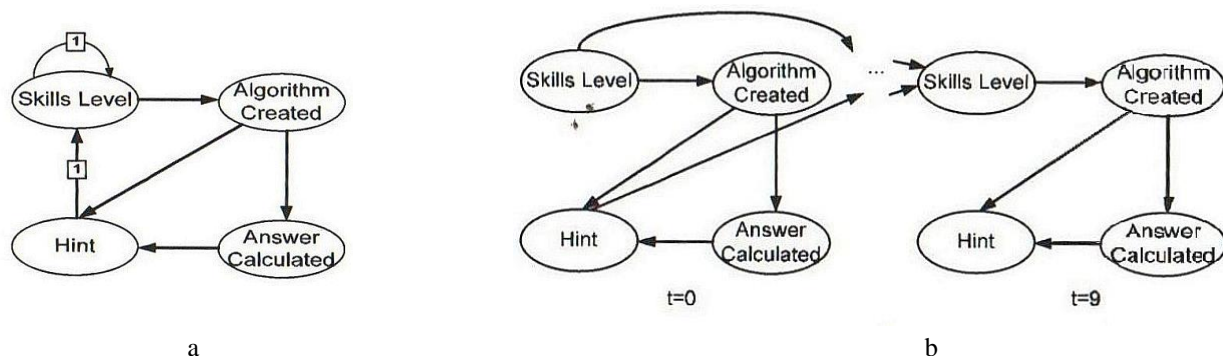


Fig. 1. Dynamic Bayesian network structure: a – in collapsed form; b – in expanded form

Hint	Yes				No			
	Excellent	Good	Satisfactory	Bad	Excellent	Good	Satisfactory	Bad
Excellent	1	0.5	0.1	0	1	0	0	0
Good	0	0.5	0.4	0.2	0	1	0	0
Satisfactory	0	0	0.5	0.3	0	0	1	0
Bad	0	0	0	0.5	0	0	0	1

Fig. 2. Conditional probabilities table for “Skills Level” node

Skill level	Excellent	Good	Satisfactory	Bad	Algorithm Created	withoutErr	withOneErr	withMoreThan1Err
withoutErr	0.9	0.55	0	0	withoutErr	0.9	0.55	0
withOneErr	0.1	0.9	0.1	0.01	withOneErr	0.1	0.9	0.1
withMoreThan1Err	0	0.05	0.9	0.99	withMoreThan1Err	0	0.05	0.9

Fig. 3. Conditional probabilities tables: a – “Algorithm Created” node; b – “Answer Calculated” node

Algorithm Created	withoutErr			withOneErr			withMoreThan1Err		
Answer Calculated	withoutErr	withOneErr	withMore...	withoutErr	withOneErr	withMore...	withoutErr	withOneErr	withMore...
Yes	○	●	●	●	●	●	●	●	●
No	●	○	○	○	○	○	○	○	○

Fig. 4. Conditional probabilities table for “Hint” node

2. Solving of dynamic model probabilistic inference problems

After building of tutoring process dynamic model structure there is an opportunity to formulate the main goals of probabilistic inference which should be solved by means of this model. The filtration, prediction and smoothing concern such problems. Solving of the given problems is directed on obtaining of the network variables evaluation that will allow to provide timely formation of system's feedbacks to the student. Let's consider in details an essence and purposes of probabilistic problems solving, as well as their mathematical aspect

2.1. Filtration

The objective of filtration is a “Skills Level” node evaluation at the moment of time $t = k$ for given model if all estimates “Algorithm Created” and “Answer Calculated” from the beginning up to the present moment are known. The calculations can be considered as the following: firstly, distribution of probabilities for current condition “Skills Level” is projected forward from t to $t+1$, then it is updated with new probabilities $AlgC_{t+1}$ and $AnsC_{t+1}$

$$\begin{aligned}
 P(SkL_{t+1} | AlgC_{1:t+1}, AnsC_{1:t+1}) &= \\
 &= \alpha \cdot P(AlgC_{t+1} | SkL_{t+1}) \cdot \\
 &\cdot P(SkL_{t+1} | AlgC_{1:t}, AnsC_{1:t}),
 \end{aligned}
 \tag{1}$$

where SkL – student competence level (node “Skills Level”); $AlgC$ – “Algorithm Created” node value; $AnsC$ – “Answer Calculated” node value, α – normalization constant introduced so that the sum of values in the $P(Y | X)$ distribution is equal to 1.

Second term $P(SkL_{t+1} | AlgC_{1:t}, AnsC_{1:t})$ represents a single-step prediction of next condition, and the first term updates it by new probabilities. Let's transform the second term, which is single-step prediction for a current condition by probabilities values updating of SkL_t

$$\begin{aligned}
 P(SkL_{t+1} | AlgC_{1:t+1}, AnsC_{1:t+1}) &= \\
 &= \alpha \cdot P(AlgC_{t+1} | SkL_{t+1}) \sum_{Hint_t} P(Hint_t | AlgC_t, AnsC_t) \cdot \\
 &\cdot \sum_{SkL_t} P(SkL_{t+1}, | SkL_t, Hint_t) \cdot P(SkL_t | AlgC_{1:t}, AnsC_{1:t}).
 \end{aligned}
 \tag{2}$$

Let's define updating of node “Hint” probabilities distribution according to probabilities $AlgC_t$ and $AnsC_t$:

$$\begin{aligned}
 P(Hint_t | AlgC_t, AnsC_t) &= \\
 &= \sum_{AlgC_t} P(AlgC_t) \cdot \\
 &\cdot \sum_{AnsC_t} P(AnsC_t | AlgC_t) \cdot P(Hint_t | AnsC_t, AlgC_t).
 \end{aligned}
 \tag{3}$$

Example. Let there are probabilities AnsC_0 and AlgC_0 for zero time slice. Find probabilities distribution for “Skills Level” node for the first time slice if probabilities AnsC_1 and AlgC_1 are known.

Initial conditions:

$$P(\text{SkL}_0) = \langle 0.25, 0.25, 0.25, 0.25 \rangle,$$

$$P(\text{AlgC}_0 = \text{withOneError}) = 1,$$

$$P(\text{AnsC}_0 = \text{withoutErrors}) = 1,$$

$$P(\text{AlgC}_0 = \text{withMoreOneError}) = 1,$$

$$P(\text{AnsC}_0 = \text{withOneError}) = 1.$$

$$\text{Find: } P(\text{SkL}_1 | \text{AlgC}_{0,1}, \text{AnsC}_{0,1}) = ?$$

Let's find out posterior probabilities:

$$P(\text{AlgC}_0) = \langle 0, 1, 0 \rangle; P(\text{AnsC}_0) = \langle 1, 0, 0 \rangle;$$

$$P(\text{AlgC}_1) = \langle 0, 0, 1 \rangle; P(\text{AnsC}_1) = \langle 0, 1, 0 \rangle.$$

$$P(\text{Hint}_0) = \sum_{\text{AlgC}_0} (\text{AlgC}_0).$$

$$\cdot \sum_{\text{AnsC}_0} P(\text{AnsC}_0 | \text{AlgC}_0) \cdot P(\text{Hint}_0 | \text{AnsC}_0, \text{AlgC}_0) =$$

$$= 0 \cdot (1 \cdot \langle 0, 1 \rangle) + 0 \cdot \langle 1, 0 \rangle + 0 \cdot \langle 1, 0 \rangle +$$

$$+ 1 \cdot (1 \cdot \langle 1, 0 \rangle + 0 \cdot \langle 1, 0 \rangle + 0 \cdot \langle 1, 0 \rangle) +$$

$$+ 0 \cdot (1 \cdot \langle 1, 0 \rangle + 0 \cdot \langle 1, 0 \rangle + 0 \cdot \langle 1, 0 \rangle) = \langle 1, 0 \rangle.$$

Having node AlgC_0 probability SkL_0 node evaluation could be updated:

$$P(\text{SkL}_0 | \text{AlgC}_0) = \alpha \cdot P(\text{SkL}_0) \cdot P(\text{AlgC}_0 | \text{SkL}_0) =$$

$$= \alpha \langle 0.25, 0.25, 0.25, 0.25 \rangle \cdot \langle 0.1, 0.9, 0.1, 0.01 \rangle =$$

$$= \alpha \langle 0.025, 0.225, 0.025, 0.0025 \rangle =$$

$$= \langle 0.09, 0.81, 0.09, 0.01 \rangle.$$

After SkL_0 node updating posterior probabilities distribution of SkL_1 node and its updating according to the AlgC_1 value could be done:

$$P(\text{SkL}_1 | \text{AlgC}_{0,1}, \text{AnsC}_{0,1}) =$$

$$= \alpha \cdot P(\text{AlgC}_1 | \text{SkL}_1) \cdot \sum_{\text{Hint}_0} P(\text{Hint}_0) \cdot$$

$$\cdot \sum_{\text{SkL}_0} (P(\text{SkL}_0) \cdot P(\text{SkL}_1 | \text{Hint}_0, \text{SkL}_0)) =$$

$$= \alpha \cdot \langle 0, 0.05, 0.9, 0.99 \rangle + 1 \cdot (0.09 \cdot \langle 1, 0, 0, 0 \rangle) +$$

$$+ 0.81 \cdot \langle 0.5, 0.5, 0, 0 \rangle + 0.09 \cdot \langle 0.1, 0.4, 0.5, 0 \rangle +$$

$$+ 0.01 \cdot \langle 0, 0.2, 0.3, 0.8 \rangle + 0 \cdot (0.09 \cdot \langle 1, 0, 0, 0 \rangle) +$$

$$+ 0.81 \cdot \langle 0, 1, 0, 0 \rangle + 0.09 \cdot \langle 0, 0, 1, 0 \rangle +$$

$$+ 0.01 \cdot \langle 0, 0, 0, 1 \rangle) = \langle 0, 0.318, 0.618, 0.064 \rangle.$$

Thus, there has been obtained “Skills Level” node probabilities distribution for the first time slice. Whereas $P(\text{AlgC}_1) = \langle 0, 0, 1 \rangle$, i.e. in the first time slice the algorithm of problem decision has been developed more than with one error, competence level evaluation was decreased from 80 percentage (good level) to 60 percentage (satisfactory level) that proves filtration calculations.

Filtration implementation for the variables of tutoring process dynamic model allows to define the current competence level. It has been accumulated as a result of all the previous tasks solving updated with the results of the two control points of last task. The evaluation of current competence level could be a basis for decision-making on the tutoring program performance termination because of achievement of the required skills indicator. As well it could be utilized for choice of the additional emergency pedagogical interventions for improvement of the given parameter.

2.2. Prediction

One of the filtration problem stages is a prediction of competence level node probabilities distribution at a current time slice before its updating by the obtained probabilities. Let's consider another challenge – prediction of DBN probabilities distribution at the moment of time $t+k$ ($k > 0$) if all the probabilities before the moment of time t are known. The algorithm might be described as following:

1) initially, set a counter of time slice number $i = t + 1$;

2) perform prediction of DBN nodes probabilities distribution at i -th time moment:

- SkL_i node probabilities distribution:

$$P(\text{SkL}_i) = \sum_{\text{Hint}_{i-1}} P(\text{Hint}_{i-1}) \cdot$$

$$\cdot \sum_{\text{SkL}_{i-1}} P(\text{SkL}_{i-1}) \cdot P(\text{SkL}_i | \text{Hint}_{i-1}, \text{SkL}_{i-1}), \quad (4)$$

- AlgC_i node evaluation:

$$P(\text{AlgC}_i) = \sum_{\text{SkL}_i} P(\text{SkL}_i) \cdot P(\text{AlgC}_i | \text{SkL}_i), \quad (5)$$

- AnsC_i node evaluation:

$$P(\text{AnsC}_i) = \sum_{\text{AlgC}_i} P(\text{AlgC}_i) \sum_{\text{AnsC}_i} P(\text{AnsC}_i | \text{AlgC}_i), \quad (6)$$

- Hint_i node evaluation:

$$P(\text{Hint}_i) = \sum_{\text{AlgC}_i} P(\text{AlgC}_i) \cdot \sum_{\text{AnsC}_i} P(\text{AnsC}_i | \text{AlgC}_i) \cdot P(\text{Hint}_i | \text{AnsC}_i, \text{AlgC}_i); \quad (7)$$

3) increase the counter *i* by one;

4) repeat steps 2-3 until reach the required moment *t + k*.

As a result of calculations prediction of network nodes posterior probabilities distribution from *t + 1* to *t + k* moment has been obtained.

Prediction of learning process model variables probabilities distribution allows to evaluate quantity of additional iterations of tutoring for achievement of required competence level. In a case if needed quantity of additional iterations exceeds a limit, feedback strategy correction should be done in order to increase the level of student's competence [8-10].

2.3. Smoothing

Smoothing problem (retrospective analysis) consists of a posterior variables values probabilities calculation. These variables cover previous state if all probabilities before current state are known. For the current learning process model it could be defined as calculation of student competence level evaluation at a moment *k*, $1 < k < t$. It would be done only if all probabilities AlgC and AnsC for every moment until *t* are obtained. Retrospective analysis provides better information evaluation about some state in the past, which was available up to that time (because it includes greater number of probabilities).

Having probabilities at moments $1:t$, when it is required to get evaluation of some node at time moment *k*, $1 < k < t$, it is convenient to make calculations of smoothing dividing task in two parts: before the moment *k* and from *k + 1* to *t*:

$$P(\text{SkL}_k | \text{AlgC}_{1:t}, \text{AnsC}_{1:t}) = \alpha \cdot P(\text{SkL}_k | \text{AlgC}_{1:k}, \text{AnsC}_{1:k}) \cdot P(\text{AlgC}_{k+1:t}, \text{AnsC}_{k+1:t} | \text{SkL}_k) = \alpha \cdot f_{1:k} \cdot b_{k+1:t}, \quad (8)$$

where $b_{k+1:t} = P(\text{AlgC}_{k+1:t}, \text{AnsC}_{k+1:t} | \text{SkL}_k)$, $f_{1:k}$ can be calculated by filtering in forward direction from 1 to *k* according to the equation (1).

Reverse expression $b_{k+1:t}$ calculated through recursive process in backward direction from *t*:

$$P(\text{AlgC}_{k+1:t}, \text{AnsC}_{k+1:t} | \text{SkL}_k) = \sum_{\text{SkL}_{k+1}} P(\text{AlgC}_{k+1} | \text{SkL}_{k+1}) \cdot P(\text{AlgC}_{k+2:t}, \text{AnsC}_{k+2:t} | \text{SkL}_{k+1}) \cdot P(\text{SkL}_{k+1} | \text{SkL}_k, \text{Hint}_k). \quad (9)$$

Example. Let's utilize the algorithm described above for calculation of competence level smoothing evaluation at the first time moment, if all the probabilities AlgC_{1:3} and AnsC_{1:3} are known.

Posterior probabilities are:

$$P(\text{AlgC}_1) = \langle 0, 0, 1 \rangle; \quad P(\text{AnsC}_1) = \langle 1, 0, 0 \rangle; \\ P(\text{AlgC}_2) = \langle 0, 1, 0 \rangle; \quad P(\text{AnsC}_2) = \langle 1, 0, 0 \rangle; \\ P(\text{AlgC}_3) = \langle 1, 0, 0 \rangle; \quad P(\text{AnsC}_3) = \langle 0, 0, 1 \rangle.$$

Initially, it is total uncertainty about competence level:

$$P(\text{SkL}_1) = \langle 0, 25, 0, 25, 0, 25, 0, 25 \rangle.$$

As a result of the described algorithm of filtering:

$$P(\text{SkL}_1 | \text{AlgC}_1, \text{AnsC}_1) = \langle 0, 0, 0258, 0, 4639, 0, 51031 \rangle.$$

The general equation for calculation of smoothing evaluation is:

$$P(\text{SkL}_1 | \text{AlgC}_{1:3}, \text{AnsC}_{1:3}) = \alpha \cdot P(\text{SkL}_1 | \text{AlgC}_1, \text{AnsC}_1) \cdot P(\text{AlgC}_{2:3}, \text{AnsC}_{2:3} | \text{SkL}_1).$$

The second multiplier is reverse expression which can be calculated by recursive procedure from $t = 3$ to $t = 1$:

$$P(\text{AlgC}_{2:3}, \text{AnsC}_{2:3} | \text{SkL}_1) = \sum_{\text{SkL}_2} P(\text{AlgC}_2 | \text{SkL}_2) \cdot P(\text{AlgC}_3, \text{AnsC}_3 | \text{SkL}_2) \cdot P(\text{SkL}_2 | \text{SkL}_1, \text{Hint}_1).$$

After insertion of calculated $AlgC_2$ node value to the formula following expression could be obtained:

$$\begin{aligned} P(AlgC_{2,3}, AnsC_{2,3} | SkL_1) &= \\ &= P(AnsC_2 | AlgC_2) \cdot \\ &\cdot \sum_{SkL_2} P(AlgC_2, | SkL_2) \cdot P(AlgC_3, AnsC_3 | SkL_2) \cdot \\ &\cdot P(SkL_2 | SkL_1, Hint_1). \end{aligned}$$

Reverse expression:

$$\begin{aligned} P(AlgC_3, AnsC_3 | SkL_2) &= \\ &= P(AnsC_3 | AlgC_3) \sum_{SkL_3} P(AlgC_3 | SkL_3) \cdot \\ &\cdot P(| SkL_4) \cdot P(SkL_3 | SkL_2, Hint_2). \end{aligned}$$

Thus, as a result of all functional components calculations for SkL_1 node smoothed evaluation has been obtained.

Let's check formula validness for smoothed evaluation of competence level calculation derivation by means of probability distribution calculation:

$$P(| SkL_4) = \langle 1, 1, 1, 1 \rangle,$$

$$\begin{aligned} P(AlgC_3, AnsC_3 | SkL_2) &= \\ &= 0.9 \cdot (0.9 \cdot 1 \cdot \langle 1, 0.5, 0.1, 0 \rangle) + \\ &+ 0.05 \cdot 1 \cdot \langle 0, 0.5, 0.4, 0.2 \rangle + 0.1 \cdot \langle 0, 0, 0.5, 0.3 \rangle + \\ &+ 0.1 \cdot \langle 0, 0, 0, 0.5 \rangle = \langle 0.81, 0.4275, 0.099, 0.009 \rangle, \end{aligned}$$

$$\begin{aligned} P(AlgC_{2,3}, AnsC_{2,3} | SkL_1) &= \\ &= 0.45 \cdot (0.1 \cdot 0.81 \cdot \langle 1, 0.5, 0.1, 0 \rangle) + \\ &+ 0.9 \cdot 0.4275 \cdot \langle 0, 0.5, 0.4, 0.2 \rangle + 0.1 \cdot 0.099 \cdot \langle 0, 0, 0.5, 0.3 \rangle + \\ &+ 0.01 \cdot 0.009 \cdot \langle 0, 0, 0, 0.5 \rangle = \\ &= \langle 0.03645, 0.10479, 0.07513, 0.03598 \rangle \end{aligned}$$

$$\begin{aligned} P(SkL_1 | AlgC_{1,3}, AnsC_{1,3}) &= \\ &= \alpha \cdot \langle 0.03645, 0.10479, 0.07513, 0.03598 \rangle \cdot \\ &\cdot \langle 0, 0.0258, 0.4639, 0.5103 \rangle = \\ &= \langle 0, 0.04835, 0.62328, 0.32837 \rangle. \end{aligned}$$

If we have probabilities for three time slices, after calculation of SkL (smoothed evaluation of student competence level) confidence in good skills level remain less than 5%. But more significant thing is improvement of satisfactory and good competence levels states. It is 62 % and 33 % respectively when six probabilities are known. After first trying to complete tasks this values will be 46 % and 51 %.

Precession restriction for evaluation of student competence level, when probabilistic inference of filtering and prediction is used deal with received probabilities at the current moment. They are algorithm designing and numerical calculations probabilities for task instance. After obtaining of additional probabilities it is possible to make retrospective analysis of knowledge level for specific moment in past and get smoothed evaluation. Smoothed evaluation allows to solve following problems:

- 1) research of the competence level growth based upon got probabilities after task series completion;
- 2) evaluation of the competence level growth based upon got probabilities after task series completion by comparison of smoothed evaluation and results of current check;
- 3) decision making on change of training trajectory or on additional pedagogical interventions in a case of insufficient growth of competence level.

3. Computer simulation of DBN

In order to verify mathematical calculations of probabilistic inference of DBN, obtained results were compared with the results of machine computation in Genie 2.0 environment [11]. The discrepancy between the results is negligible. In fig. 5 the results of machine computation of competence level smoothed evaluation for initial state are shown with stated time moments t and probability levels in percent. It might be spotted that depending on errors detecting in calculations and algorithm implementation, the probabilities of student grades redistribute appropriately.

Fig. 6 shows a graph of Skills Level values versus time for correct input of the answer and creating an algorithm with more than one error at time $t = 1$, correct input of the answer and creating an algorithm with one error at time $t = 2$. The graph shows how the Excellent and Good scores change over time. This proves the effectiveness of the proposed approach, since knowledge and skills are modeled adequately.

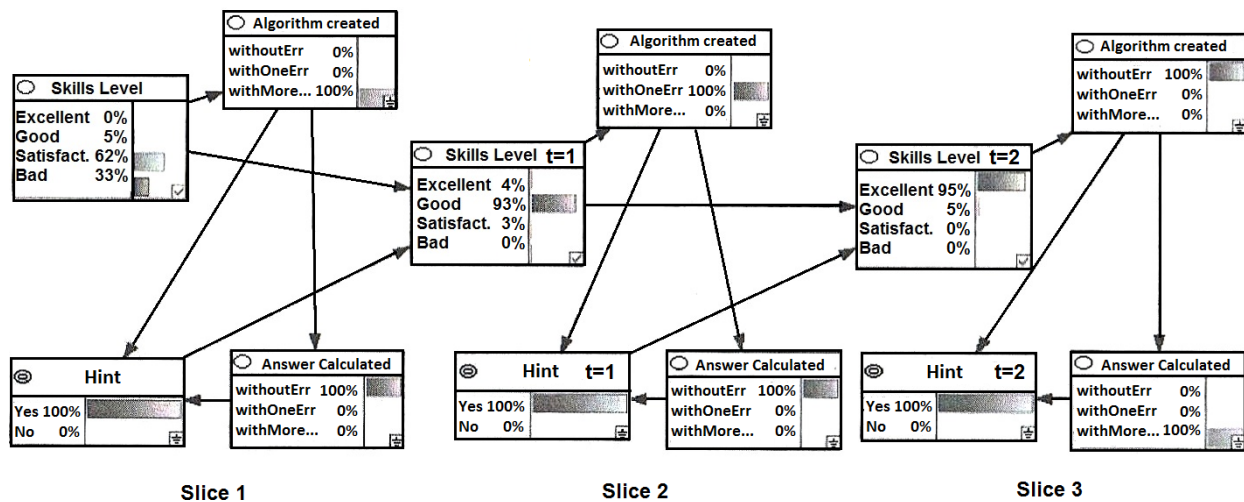


Fig. 5. Smoothing problem solution in Genie 2.0 environment

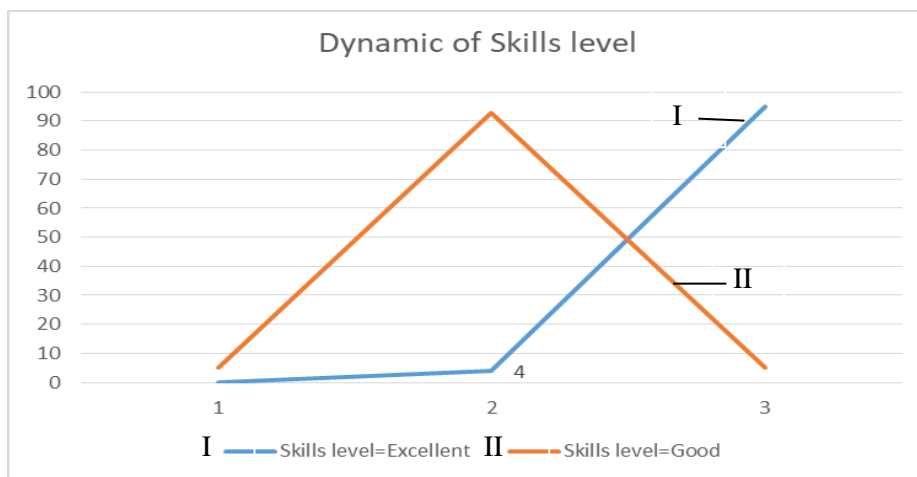


Fig. 6. Dynamic of skills level after student answer

Conclusions and perspectives

The DBN development for the tutoring process based on the engineering tasks solving is represented. The mathematic calculations for probabilistic inference problems such as filtering, prediction, smoothing have been considered. Filtering problem solution allows to evaluate current student competence level after obtaining of last probabilities for his algorithm development and his numerical calculations of assigned task. The probabilities distribution of tutoring process model was predicted. Evaluation of additional iterations number needed for achieving of required competence level is done. Retrospective analysis allows to obtain smoothed evaluation of competence level which was got after previous task instance completion and after having the new additional probabilities characterizing completion of two task breakpoints. Solution of described probabilistic

inference problems gives opportunity to provide correct information about tutoring process for the Intelligent Tutoring Systems. It helps to give proper feedbacks and to track students' competence level according to heuristic principles of self-organization in intelligence systems [12, 13].

Developed probabilistic inference kernel technique could be used as a basis for decision-making model for automated tutoring process [14, 15].

References (GOST 7.1:2006)

1. Murray, C. R. *Looking Ahead to Select Tutorial Actions: A Decision-Theoretic Approach* [Text] / C. R. Murray, K. VanLehn, J. Mostow // *International Journal of Artificial Intelligence in Education*. – 2004. – Vol. 14, No. 3, 4. – P. 235-278. HAL Id: hal-00197308.
2. *Open learner models in supporting self-regulated learning in higher education: A systematic*

literature review [Text] / D. Hooshyar et al. // *Computers & Education*. – 2020. – Vol. 154. – P. 103-108. – Article Id: 103878. DOI: /10.1016/j.compedu.2020.103878.

3. How Do Students Perceive Their Own and Their Peers' Progress in E-Learning? [Text] / S. Somyürek, et al // *International Journal of Information and Learning Technology*. – 2021. – Vol. 38, No. 1. – P. 49-74. DOI: 10.1108/IJILT-05-2020-0073.

4. Corbeil, J. P. Probabilistic Approaches to Detect Blocking States in Intelligent Tutoring System [Text] / J. P. Corbeil, M. Gagnon, P. R. Richard // *Proceedings on 16th International Conference on Intelligent Tutoring Systems, Athens, Greece, June 8-12, 2020*. – P. 79-88. DOI: /10.1007/978-3-030-49663-0_11.

5. Käser, T. Dynamic Bayesian Networks for Student Modeling [Text] / T. Käser, S. Klingler, A. G. Schwing, M. Gross // *IEEE Transactions on Learning Technologies*, 2017. – Vol. 10, No. 4. – P. 450-462. DOI: 10.1109/TLT.2017.2689017.

6. Learning a skill-teaching curriculum with dynamic bayes nets [Text] / D. T. Green, T. J. Walsh, P. R. Cohen, Y. H. Chang // *Proceedings of the 25th AAAI Conference on Artificial Intelligence and the 23rd Innovative Applications of Artificial Intelligence Conference, 2011*. – San Francisco, California, USA. – P. 1648-1654.

7. McNaught, K. Using Dynamic Bayesian Networks for Prognostic Modelling to Inform Maintenance Decision Making [Text] / K. McNaught, A. Zagorecki // *IEEE International Conference on Industrial Engineering and Engineering Management*. – 2009. – P. 1155-1159.

8. Alevan, V. Limitations of student control: Do students know when they need help? [Text] / V. Alevan, K. Koedinger // *Intelligent Tutoring Systems: 5th International Conference ITS, Montreal, Canada, 2000*. – P. 292-303.

9. Mitrovic, A. An intelligent SQL tutor on the web [Text] / A. Mitrovic // *International Journal of Artificial Intelligence in Education*. – 2003. – Vol. 13, No. 3-4. – P. 173-197.

10. Development of the automated laboratory practical work at the course «Modeling of systems» [Text] / A. Kulik, A. Chukhray, S. Pedan, P. Anzenberger // *Proceedings of the Interactive Computer Aided Learning Conference, Villach, Austria, September 24-26, 2008*. – P. 1-11.

11. Marek, J. D. GeNIe: A Development Environment for Graphical Decision-analytic Models [Text] / J. D. Marek // *Proceedings of the Annual Symposium of the American Medical Informatics Association*. – Washington, D.C., 1999. – Corpus ID: 29592997. – P. 1206.

12. Доценко, С. І. Інтелектуальні системи: принципи евристичної самоорганізації процесів смислового мислення та смислової діяльності [Текст] / С. І. Доценко // *Радіоелектронні і комп'ютерні системи*. – 2020. – № 2(94). – С. 4-22. DOI: 10.32620/reks.2020.2.01.

13. Доценко, С. І. Інтелектуальні системи: принцип евристичної самоорганізації [Текст] / С. І. Доценко // *Радіоелектронні і комп'ютерні системи*. – 2020. – № 1(93). – С. 4-16. DOI: 10.32620/reks.2020.1.01.

14. Structural diagnosis method for computer programs developed by trainees [Text] / D. Gaydachuk, O. Havrylenko, J. P. Martínez Bastida, A. Chukhray // *Proceedings of 15th International Conference on ICT in Education, Research, and Industrial Applications*. – Kherson, Ukraine, 2019. – P. 485-490.

15. Martínez Bastida, J. P. Developing a self-regulation environment in an open learning model with higher fidelity assessment [Text] / J. P. Martínez Bastida, O. Havrylenko, A. Chukhray // *Communications in Computer and Information Science*. – 2018. – Vol. 826. – P. 112-131. DOI: 10.1007/978-3-319-76168-8_6.

References (BSI)

1. Murray, C. R., VanLehn, K., Mostow, J. Looking Ahead to Select Tutorial Actions: A Decision-Theoretic Approach. *International Journal of Artificial Intelligence in Education*, 2004, vol. 14, no. 3,4, pp. 235-278. HAL Id: hal-00197308.

2. Hooshyar, D., Pedaste, M., Saks, K., Leijen, Ä., Bardone, E., Wang, M. Open learner models in supporting self-regulated learning in higher education: A systematic literature review. *Computers & Education*, 2020, vol. 154, pp. 103-108, Article Id: 103878. DOI: 10.1016/j.compedu.2020.103878.

3. Somyürek, S., Brusilovsky, P., Çebi, A., Akhüseyinoğlu, K., Güyer, T. How Do Students Perceive Their Own and Their Peers' Progress in E-Learning? *International Journal of Information and Learning Technology*, 2021, vol. 38, no. 1, pp. 49-74. DOI: 10.1108/IJILT-05-2020-0073.

4. Corbeil J. P., Gagnon, M., Richard, P. R. Probabilistic Approaches to Detect Blocking States in Intelligent Tutoring System. *Proceedings on 16th International Conference on Intelligent Tutoring Systems, Athens, Greece, June 8-12, 2020*, pp. 79-88. DOI: /10.1007/978-3-030-49663-0_11.

5. Käser, T., Klingler, S., Schwing, A. G., Gross, M. Dynamic Bayesian Networks for Student Modeling. *IEEE Transactions on Learning Technologies*, 2017, vol. 10, no. 4, pp. 450-462. DOI: 10.1109/TLT.2017.2689017.

6. Green, D. T., Walsh, T. J., Cohen, P. R., & Chang, Y. H. Learning a skill-teaching curriculum with dynamic bayes nets. *Proceedings of the 25th AAAI Conference on Artificial Intelligence and the 23rd Innovative Applications of Artificial Intelligence Conference, 2011*, San Francisco, California, USA, pp. 1648-1654.

7. McNaught, K., Zagorecki, A. Using Dynamic Bayesian Networks for Prognostic Modelling to Inform Maintenance Decision Making. *IEEE International Conference on Industrial Engineering and Engineering Management*, 2009, pp. 1155-1159.

8. Alevin, V., Koedinger, K. Limitations of student control: Do students know when they need help? *Intelligent Tutoring Systems: 5th International Conference ITS*, 2000, Montreal, Canada, pp. 292-303.

9. Mitrovic, A. An intelligent SQL tutor on the web. *International Journal of Artificial Intelligence in Education*, 2003, vol. 13, no. 3-4, pp. 173-197.

10. Kulik, A., Chukhray, A., Pedan S., Anzenberger, P. Development of the automated laboratory practical work at the course «Modeling of systems». *Proceedings of the Interactive Computer Aided Learning Conference*, September 24-26, 2008, Villach, Austria, pp. 1-11.

11. Marek, J. D. GeNIe: A Development Environment for Graphical Decision-analytic Models. *Proceedings of the Annual Symposium of the American Medical Informatics Association*, Washington, D.C., 1999, Corpus ID: 29592997, pp. 1206.

12. Dotsenko, S. I. Intel'ektual'ni systemy: pryntsyvy evrystychnoyi samoorganizatsiyi protsesiv smyslovoho myslennya ta smyslovoyi diyal'nosti [Intellectual systems: principles of the heuristic self-organization of the processes of sense thinking and

sense activity]. *Radioelektronni i komp'uterni sistemi – Radioelectronic and computer systems*, 2020, no. 2(94), pp. 4-22. DOI: 10.32620/reks.2020.2.01.

13. Dotsenko, S. I. Intel'ektual'ni systemy: pryntsyvy evrystychnoyi samoorganizatsiyi [Intellectual systems: principles of the heuristic self-organization]. *Radioelektronni i komp'uterni sistemi – Radioelectronic and computer systems*, 2020, no. 1(93), pp. 4-16. DOI: 10.32620/reks.2020.1.01.

14. Gaydachuk, D., Havrylenko, O., Martínez Bastida, J. P., Chukhray, A. Structural diagnosis method for computer programs developed by trainees. *Proceedings of 15th International Conference on ICT in Education, Research, and Industrial Applications*, 2019, Kherson, Ukraine, pp. 485-490.

15. Martínez Bastida, J. P., Havrylenko, O., Chukhray, A. Developing a self-regulation environment in an open learning model with higher fidelity assessment. *Communications in Computer and Information Science*, 2018, vol. 826, pp. 112-131. DOI: 10.1007/978-3-319-76168-8_6.

Надійшла до редакції 15.04.2021, розглянута на редколегії 20.05.2021

МОДЕЛЮВАННЯ ПРОЦЕСУ НАВЧАННЯ ІНЖЕНЕРНИМ НАВИЧКАМ З ВИКОРИСТАННЯМ ДИНАМІЧНИХ БАЙЄСІВСЬКИХ МЕРЕЖ

А. Г. Чухрай, О. В. Гавриленко

Предметом дослідження в статті є процес інтелектуального комп'ютерного навчання інженерним навичкам. **Метою** є моделювання процесу навчання інженерним навичкам в інтелектуальних комп'ютерних навчальних програмах за допомогою динамічних байєсівських мереж. **Завдання:** запропонувати підхід до моделювання процесу навчання інженерним навичкам. Оцінити рівень студентських компетенцій шляхом розгляду умінь складати алгоритми виконання інженерних завдань і вмінь їх застосовувати. Створити структуру динамічної байєсівської мережі процесу навчання. Вибрати значення для таблиць умовних ймовірностей. Вирішити завдання фільтрації, прогнозування та ретроспективного аналізу. Виконати моделювання розробленої динамічної байєсівської мережі з використанням спеціального середовища Genie 2.0. Використовуваними **методами** є методи теорії ймовірностей і методи виведення в байєсівських мережах. Отримані наступні **результати:** представлена розробка динамічної байєсівської мережі для навчального процесу на основі рішення інженерних задач. Розглянуто математичні розрахунки для задач імовірнісного виведення, таких як фільтрація, прогнозування, згладжування. Рішення завдання фільтрації дозволяє оцінити поточний рівень компетентності студента після отримання останніх ймовірностей розробки алгоритму і його чисельних розрахунків поставленого завдання. Передбачено розподіл ймовірностей моделі процесу навчання. Зроблено оцінку кількості додаткових ітерацій, необхідних для досягнення необхідного рівня компетенції. Ретроспективний аналіз дозволяє отримати згладжену оцінку рівня компетентності, яка була отримана після виконання попереднього примірника завдання і після наявності нових додаткових можливостей. Рішення описаних задач імовірнісного виведення дає можливість надати правильну інформацію про процес навчання для інтелектуальних комп'ютерних навчальних систем. Це допомагає отримувати належний зворотний зв'язок і відслідковувати рівень компетентності студентів. Розроблена методика ядра імовірнісного виведення може бути використана в якості основи для моделі прийняття рішень для автоматизованого процесу навчання. **Висновки.** Наукова новизна полягає в тому, що динамічні байєсівські мережі застосовані до нового класу задач, пов'язаних з моделюванням навчання інженерним навичкам в процесі виконання алгоритмічних завдань.

Ключові слова: динамічна байєсівська мережа; моделювання; інженерні навички; прогнозування; фільтрація; згладжування.

МОДЕЛИРОВАНИЕ ПРОЦЕССА ОБУЧЕНИЯ ИНЖЕНЕРНЫМ НАВЫКАМ С ИСПОЛЬЗОВАНИЕМ ДИНАМИЧЕСКИХ БАЙЕСОВСКИХ СЕТЕЙ

А. Г. Чухрай, Е. В. Гавриленко

Предметом исследования в статье является процесс интеллектуального компьютерного обучения инженерным навыкам. **Целью** является моделирование процесса обучения инженерным навыкам в интеллектуальных компьютерных обучающих программах посредством динамических байесовских сетей. **Задачи:** предложить подход к моделированию процесса обучения инженерным навыкам. Оценить уровень студенческих компетенций путем рассмотрения умений составлять алгоритмы выполнения инженерных заданий и умений их применять. Создать структуру динамической байесовской сети процесса обучения. Выбрать значения для таблиц условных вероятностей. Решить задачи фильтрации, прогнозирования и ретроспективного анализа. Выполнить моделирование разработанной динамической байесовской сети с использованием специальной среды Genie 2.0. Используемыми **методами** являются методы теории вероятностей и методы вывода в байесовских сетях. Получены следующие **результаты:** представлена разработка динамической байесовской сети для учебного процесса на основе решения инженерных задач. Рассмотрены математические расчеты для задач вероятностного вывода, таких как фильтрация, прогнозирование, сглаживание. Решение задачи фильтрации позволяет оценить текущий уровень компетентности студента после получения последних вероятностей разработки алгоритма и его численных расчетов поставленной задачи. Предсказано распределение вероятностей модели процесса обучения. Произведена оценка количества дополнительных итераций, необходимых для достижения необходимого уровня компетенции. Ретроспективный анализ позволяет получить сглаженную оценку уровня компетентности, которая была получена после выполнения предыдущего экземпляра задачи и после наличия новых дополнительных вероятностей, характеризующих выполнение двух контрольных точек. Решение описанных задач вероятностного вывода дает возможность предоставить правильную информацию о процессе обучения для интеллектуальных компьютерных обучающих систем. Это помогает получать надлежащую обратную связь и отслеживать уровень компетентности студентов. Разработанная методика ядра вероятностного вывода может быть использована в качестве основы для модели принятия решений для автоматизированного процесса обучения. **Выводы.** Научная новизна заключается в том, что динамические байесовские сети применены к новому классу задач, связанных с моделированием обучения инженерным навыкам в процессе выполнения алгоритмических заданий.

Ключевые слова: динамическая байесовская сеть; моделирование; инженерные навыки; прогнозирование; фильтрация; сглаживание.

Чухрай Андрій Григорович – д-р техн. наук, доцент, зав. каф. математичного моделювання та штучного інтелекту, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», Харків, Україна.

Гавриленко Олена Володимирівна – канд. техн. наук, доцент каф. систем управління літальних апаратів, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», Харків, Україна.

Andrey Chukhray – Doctor of Technical Sciences, Associate Professor, Head of the Department mathematical modelling and artificial intelligence, National Aerospace University «Kharkiv Aviation Institute», Kharkiv, Ukraine,

e-mail: achukhray@gmail.com, ORCID: 0000-0002-8075-3664.

Olena Havrylenko – Candidate of Technical Science, Associate Professor of the Department «Fly Vehicle's Control Systems», National Aerospace University «Kharkiv Aviation Institute», Kharkiv, Ukraine,

e-mail: lm77191220@gmail.com, ORCID: 0000-0001-5227-9742.