www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issues 8 Aug 2016, Page No. 17555-17561

A New Approximated To the Natural Intelligence Decision Tree

Olga Popova [1], Dmitry Romanov[2], Marina Evseeva [3]

[1] Kuban State Technological University, Institute of Computer Systems and Information Security
2, street Moscow, Krasnodar, Russia
popova_ob@mail.ru

[2] Kuban State Technological University, Institute of Computer Systems and Information Security
2, street Moscow, Krasnodar, Russia
romanovda1@rambler.ru

[3]Kuban State University, Department of Romano-Germanic Philology,
149, street Stavropolskaya, Krasnodar, Russia
khizova2004@mail.ru

Abstract: ***The problem of adjustment of modern intelligence enhancement methods and automated data analysis methods to the problems that are still handled manually is fairly topical. For the solution of such problems, this study suggests a new DT representation which uses approximated to the NI knowledge structuring. The structuring is implemented by the authors' question-answer binary tree. This is a new DT with only most optimal decisions for all known situations excluding non-efficient cases. A set of 'the most effective' solutions are leaves of the tree. This new approach can be applied in intelligent decision support systems (IDSS) which enhance the natural intelligence of the scientist in the exploratory research. This tree was tested on the problem of selecting 'the most suitable' optimization method out of all known ones. First, detailed material on the main optimization methods was selected. The material was processed by new rules of deriving tree elements. The resulted tree has 127 nodes, 64 leaves are optimization methods (solution options). 63 intermediary nodes form a unique path from root to leaf, showing the progress to the most suitable method. Also, an IDSS was implemented in C#. The paper dwells on all stages of the DT construction with detailed illustrations, including video. The suggested DT allowed: simplify knowledge base designing; reduce system designing time; simplify decision search algorithm in the knowledge base; refer to the expert in case of contributing one's own developed knowledge to the subject field in the tree; obtain a new way of meta-knowledge representation.***

Keywords: intelligence enhancement, exploratory research, decision tree, Data Mining, natural intelligence, knowledge structuring

1. Introduction

Nowadays, information collecting, processing and storing methods are rapidly developing enabling collecting and storing huge data massifs. The volumes of such data massifs are enormous; therefore, automated methods of data studying – Data Mining [1, 24] – have become so topical. One of such methods is a well-known Decision Tree (DT) [2-7, 9, 21]. It is easy to understand and interpret, does not require a special data preparation, is reliable and manages huge massifs with no special preparation procedures.

However, this method has a number of disadvantages failing to solve some today's tasks which are still handled manually, or are not settled efficiently or precisely enough. Among them is the exploratory research task of finding the optimal solution method for a problem in the given scientific field.

Here are some major disadvantages of this decision method [4, 7, 8, 10]:

- the task of finding the optimal DT even for a simple problem is NP-complete, where the only optimal decision exists only locally at each node, therefore modern algorithms of DTs cannot ensure optimality of the whole tree;
- due to possible 'overfitting' the method of tree depth control has to be used;
- if the description of concept is complex, then the DT will be extremely large; consequently, additional approaches with their algorithms have to be used to solve the problem;
- for different data special techniques of attribute information value distribution have to be applied;
- today there are no heuristic rules that would be of a greater practical value, as many of them are only applicable in some special cases;
- all existing rules for growing a DT are designed for numerical sets with further application of criterion method or binary method to obtain the value of choice function, but they are completely unsuitable for non-numerical sets [13, 17-20];
- one DT is grown for all cases of one situation, therefore as many DTs have to be grown as many situations there are, which prevents from solving problems optimally with finding optimal solutions in the end.

The above-mentioned disadvantages are connected with the traditional way of growing a DT. Thus, a data set is written as follows:

$$(x, Y) = (x_1, x_2, x_3, \cdots, x_k, Y),$$

where Y is a target variable to analyze, classify and generalize. This variable depends on vector $\vec{x}$ which comprises input variables $x_1, x_2, x_3, \cdots, x_k$. Obviously, the solution to this situation will be the case with its set of input data that presents the most suitable solution Y. Among the given data, this may be the right solution but not perfect, as such data set corresponds to certain experience which may be incomplete and require further training/fitting.

This paper suggests a new way of constructing a DT which is approximated to the natural intelligence (NI) and based on the new intelligence enhancement method (IEM) suggested by

DOI: 10.18535/ijecs/v5i8.22

authors. This does not include the whole path of NI training required for the experience acquisition. Into account is only taken the way the NI structures perfect knowledge so as to arrive at the most optimal decision.

2. Data and Method. Theory.

To improve the efficiency of Data Mining [10] only one type of DT used to be constructed, which contains all ideal decisions for all possible situations. Thus, it does not cover other inefficient, non-productive and other cases.

In this case, the set of 'the most effective' decisions

$$(x_i, Y_i) = (x_{1i}, x_{2i}, x_{3i}, \cdots, x_{ki}, Y_i)$$

will be the leaves of such DT, where $\vec{x}_i$ is the set of the most correct cognitive representations of all situations with the only right and 'most suitable' decisions. Each correct cognitive situation representation has its 'most efficient' decision, e.g.

$$(x_2, Y_2) = (x_{12}, x_{22}, x_{32}, \cdots, x_{k2}, Y_2).$$

Thus, all tree paths, from root to leaves, are the most correct representations of all situations, where the number of decisions equals the number of known situations and the number of all possible paths from its root to each leaf. The introduction of new elements into this tree is possible only on condition of appearing new, 'ideal', solution methods to completely new situations.

This DT differs from the previously known one in the fact that for each DT there used to be a set of cognitive representations of one situation $\vec{x}_i$ among which only one was correct and became the solution to the objective function Y. The DT fitting and completion by new, more accurate, solutions took place in the course of the analysis of that situation.

The individual cognitive 'power' of situation representation depends on many factors, such as: experience, amount of knowledge acquired from learning, the 'force/power' of NI, the research of which has started recently; therefore, this approach has been unknown before. Methods of intelligence enhancement have appeared recently, as well as external ways of the NI enhancement, becoming more and more popular. Yet, they still have not been integrated in Data Mining, nor intelligent DSSs and other ITs.

Therefore, the situation model was usually described by different cases of cognitive perception from the worst to the best, where the latter corresponded to the individual subjective experience, which in reality was not the best and needed different fitting algorithms. The model was described as part of the standard idea of artificial intelligence which was not approximated to the NI, but succeeded in doing well-formalized problems by the known mathematical methods [2-6, 9, 10, 21]. The situation data are presented in massifs, binary relations, graphs, neural networks, demanding various algorithms operating great volumes of data.

So, any situation can be described by different people by a certain set $\vec{x}_i$ where various combinations of input elements describing this situation are possible. These combinations can differ in values, different sequences of input elements, and their numbers. Another option can be, when some input data are split into their component parts which in turn can differ in sequence and values.

Shall we consider one of such situations:

$$\vec{x}_i = \begin{cases} (x_5, x_1, \cdots, x_{12}) \to \text{'worst cognitive representation'}; \\ (x_5, x_2, x_{1_1}, x_{1_2}, x_{1_3}, x_{1_4}, \cdots, x_{12}), \text{ where } x_1 = x_{1_1} + x_{1_2} + x_{1_3} + x_{1_4}; \\ (x_{1_1}, x_5, x_2, x_{12}, x_{1_2}, x_9, x_{1_3}, x_{1_4}, \cdots, x_4); \\ \cdots \\ (x_7, x_3, x_2, x_{15}, x_1, x_{11}, \cdots, x_{10}) \to \text{'best cognitive representation'}. \end{cases}$$

If the NI had to deal with such situation, out of many cases it would memorise only the one that corresponded to the best cognitive representation. As well as it would memorise one right solution corresponding to that situation to apply it when facing similar situations. In the course of life, the number of right solutions is growing and forming experience where the best cognitive representations are mainly stored.

Obviously, the most experienced individual, expert, is the one whose NI has accumulated the biggest number of right solutions for different situations which it can use at any moment infallibly. They may represent a special value if a person seeks a solution, not a way out of the situation he/she marred himself/herself by making a wrong decision. In this case, one has to remember the whole experience and all solutions – both good and bad.

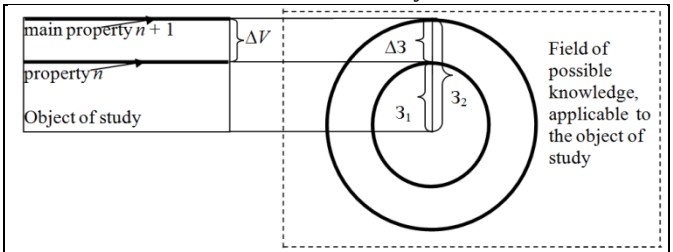
This new way of deriving DTs adopts another approach to situation modeling. The DT uses cognitive representations of the most experienced expert where each situation is instantly solved by 'ad hoc most suitable' tool. This approach leads to an optimal DT which is easily interpreted.

Moreover, this approach visualizes very well all knowledge available to the most experienced expert at one tree. This corresponds to the most productive way of information structuring and knowledge representation employed by the NI (Urok 3. Strukturirovanie informacii, 2016). This approach is known and applied as a technique which improves cognitive abilities of the NI and allows it to structure acquired knowledge precisely.

This new DT adopts its own rules of deriving root, intermediary nodes and leaves [15, 16, 19], different from the known ones.

Rule 1: If the scientific and technical progress (STP) made the object of study change its *n* property to *n* + 1property (see Fig. 1), the question '**Is it a problem with *n* property?**' will be the root of question-answer binary tree (see Fig. 2).

When a new root of the tree is derived (see Fig. 3), the old root is considered to be an intermediary node.



3_1 – a field of knowledge applicable to the object of study when *n* property was the main one;

3_2 – a new field of knowledge applicable to the modified by STP object of study with the main property *n*;

$\Delta 3 = 3_1 - 3_2$ – a new body of knowledge from the field of knowledge correspondent to the object of study to address the new volume of scientific problems ΔV.

Figure 1: The change of ratio of scientific problem volume to the knowledge from the field of knowledge when the object main property *n* is replaced by *n* + 1

DOI: 10.18535/ijecs/v5i8.22

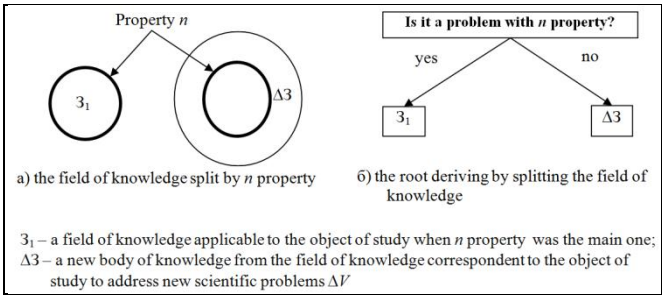


Figure 2: The deriving of the root of question-answer binary tree

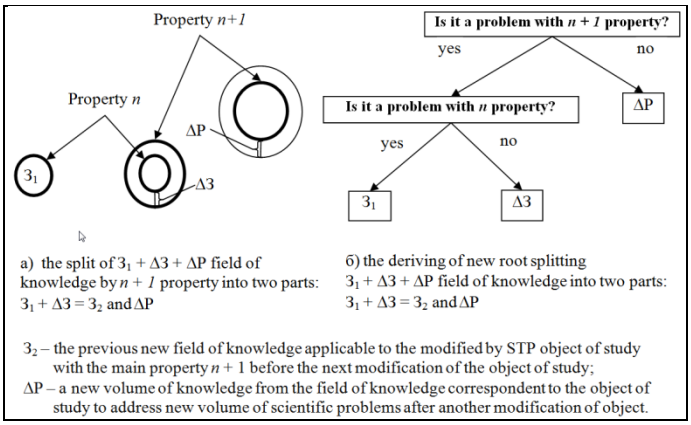


Figure 3: The deriving of new root of question-answer binary tree

Rule 2: With the help of Rule 1, each knowledge field is split into two parts until in each part there is only one group of knowledge united by one common property (see Fig. 4). Each identified root here is an intermediary node of question-answer binary tree, except the one with the main property of the first part. It is the root of question-answer binary tree.

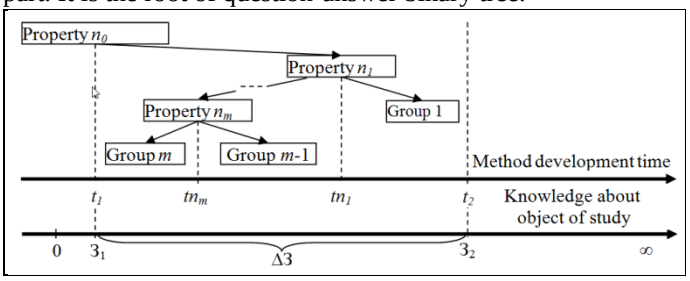


Figure 4: Property identification *n1*, …, *nm*, dividing Δ3 into *m* groups of methods – Group 1, …, Group *m*-1 and Group *m* in chronological order influenced by the STP

Rule 3: With the help of Rule 1, each knowledge group is split into two parts so as the second part would have only one piece of knowledge. The first part is divided into parts until each part has only one piece of knowledge. The identified root of each first part will be an intermediary node of question-answer binary tree, the split knowledge – the leaves of this tree. The knowledge should go down gradually from more complex to less complex in levels (see Fig. 5).

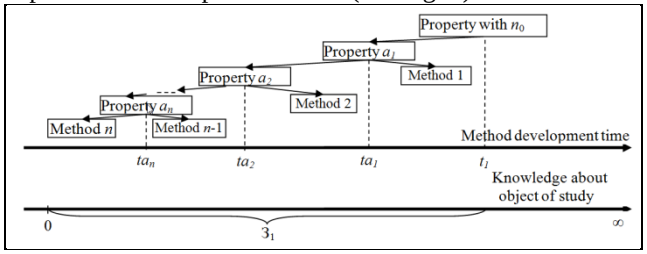


Figure 5: Property identification *a1*, *a2*, …, *an*, dividing a group of methods with property *n*0 into methods going from more to less complex

The above-mentioned rules reflect an algorithm of constructing the DT (Popova et al, 2013 c), if they are consecutively applied for deriving all intermediary nodes and leaves.

The structure of this DT makes it possible not only to store knowledge, but search for a solution, following the tree, answering the questions that help to form the correct cognitive representation of the current situation. In such tree, situations of one type are represented by the correspondent set of properties ($\vec{x}_i$), i.e. by the sequence of questions resulting in one correct answer – 'the most suitable' solution.

The suggested tree is different from others in the fact that at intermediary nodes there are questions (attributes) containing complex properties (not necessarily numerical), separating a group of solutions from the whole set of all solutions. This tree does not have 'branches' and its attributes do not have information values as the general approach has changed. Each question has only two answers 'yes' and 'no' the choice of which makes a transition to the next level. This means that each node at splitting produces only two descendants. Such split is characteristic of natural structuring.

Such tree refers to binary trees. Among all known algorithms of DT construction, CART (Classification and Regression Tree) algorithm is the closest to the suggested. It is used to obtain a dichotomic classification model, as well as for solving classification and regression problems.

3. Results and Discussion

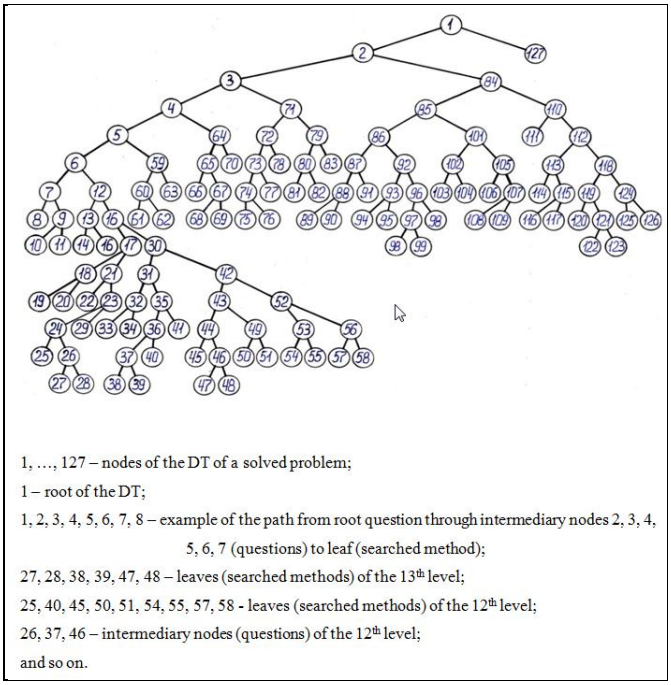
The suggested new way of constructing a DT approximated to the NI was approbated on doing a choice problem – selecting 'the most suitable' optimization method out of the set of all known optimization methods [11-14, 18].

First, quality material on the main optimization methods has been selected. It is a book 'Methods of Optimization' published by one of the outstanding Russian Schools headed by Prof. F.P. Vassil'ev [22]. This book gives a very detailed description of each method, providing a good basis for identifying common properties of optimization methods groups, as well as properties differentiating each method in its group.

In total, the resulting DT turned out to have 127 nodes (see Figs. 6 - 13). 64 leaves of which are optimization methods, i.e. solution options. The rest 63 nodes are intermediary nodes, giving their unique from root to leaf paths and showing the course of finding a solution – selecting the most suitable method (see Fig. 6).

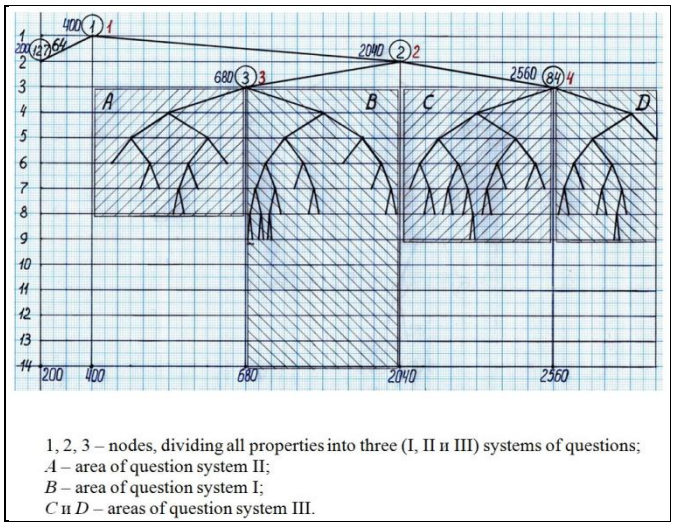
To research and implement this DT in an intelligent DSS, the complete DT with questions at nodes was replaced by a simplified binary tree (Fig. 6). To be able to use this tree (рис. 6) for an efficient search for the best solution, as well as for the search with input without tree restructuring, it was transformed into the type with tagged nodes in increments of 20 (see Figs. 7-13).

DOI: 10.18535/ijecs/v5i8.22



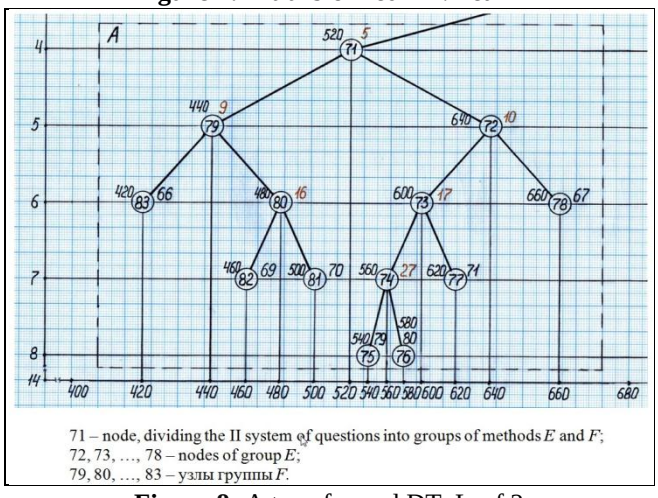
1, …, 127 – nodes of the DT of a solved problem;
1 – root of the DT;
1, 2, 3, 4, 5, 6, 7, 8 – example of the path from root question through intermediary nodes 2, 3, 4, 5, 6, 7 (questions) to leaf (searched method);
27, 28, 38, 39, 47, 48 – leaves (searched methods) of the 13th level;
25, 40, 45, 50, 51, 54, 55, 57, 58 - leaves (searched methods) of the 12th level;
26, 37, 46 – intermediary nodes (questions) of the 12th level;
and so on.

Figure 6: A simplified binary tree based on the complete DT to search for the optimization method out of all possible methods



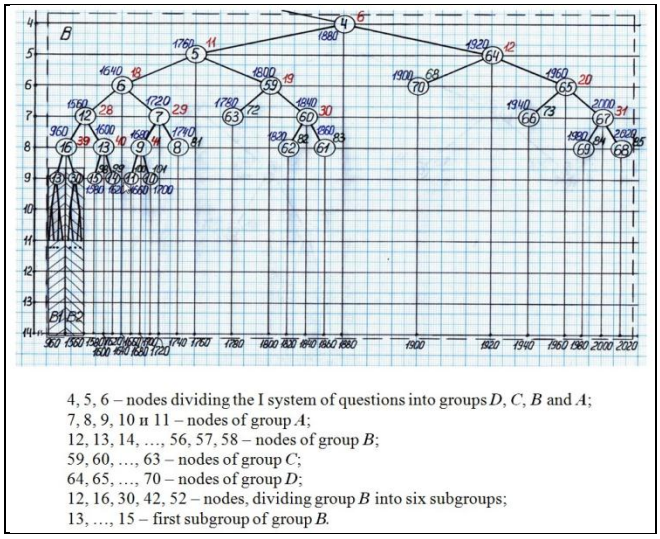
1, 2, 3 – nodes, dividing all properties into three (I, II и III) systems of questions;
A – area of question system II;
B – area of question system I;
C и *D* – areas of question system III.

Figure 7: A transformed DT. Leaf 1



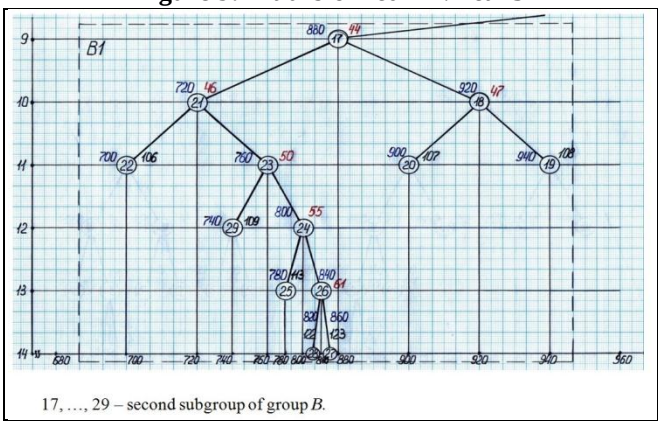
71 – node, dividing the II system of questions into groups of methods *E* and *F*;
72, 73, …, 78 – nodes of group *E*;
79, 80, …, 83 – узлы группы *F*.

Figure 8: A transformed DT. Leaf 2



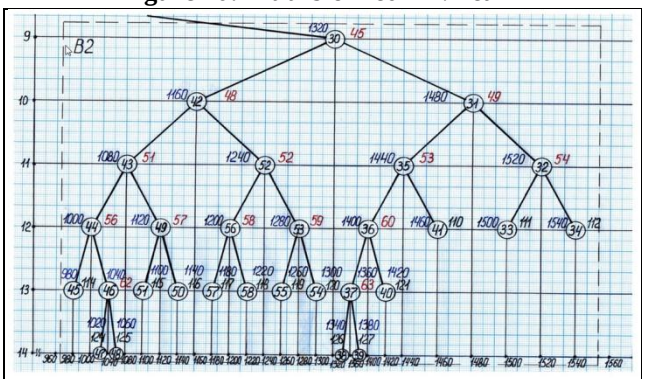
4, 5, 6 – nodes dividing the I system of questions into groups *D*, *C*, *B* and *A*;
7, 8, 9, 10 и 11 – nodes of group *A*;
12, 13, 14, …, 56, 57, 58 – nodes of group *B*;
59, 60, …, 63 – nodes of group *C*;
64, 65, …, 70 – nodes of group *D*;
12, 16, 30, 42, 52 – nodes, dividing group *B* into six subgroups;
13, …, 15 – first subgroup of group *B*.

Figure 9: A transformed DT. Leaf 3



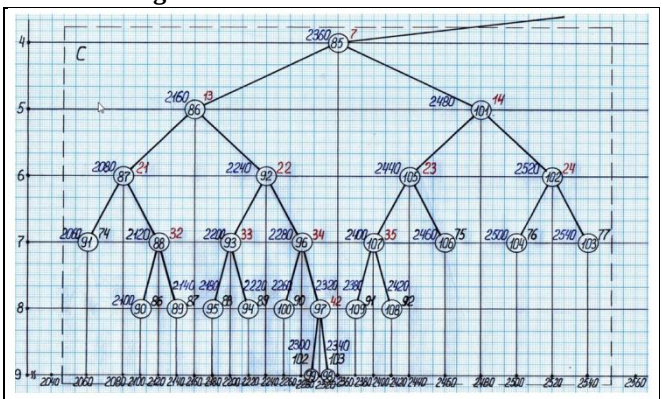
17, …, 29 – second subgroup of group *B*.

Figure 10: A transformed DT. Leaf 4



31, …, 41 – third subgroup of group *B*;
43, …, 51 – fourth subgroup of group *B*;
53, 54, 55 – fifth subgroup of group *B*;
56, 57, 58 – sixth subgroup of group *B*.

Figure 11: A transformed DT. Leaf 5



85, 86 – nodes, dividing the III system of questions into groups *J*, *H* и *G*;
101, 102, …, 109 – nodes of group *J*;
92, 93, …, 100 – nodes of group *H*;
87, 88, …, 91 – nodes of group *G*.

Figure 12: A transformed DT. Leaf 6

DOI: 10.18535/ijecs/v5i8.22

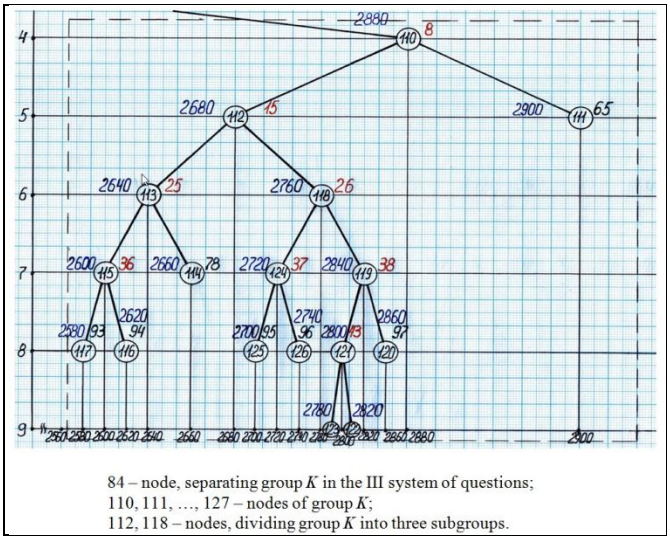


Figure 13: A transformed DT. Leaf 7

Such representation of the DT allows one to test it for errors in its structure and so avoid errors in making decisions. Especially, it will be important with increasing number of elements.

It is obvious (Figs. 7-13) that the input of elements is connected with appearing new optimization methods, which can be added to their group of methods without breaking the structure of the whole tree. It is possible due to the structure of binary tree and the search with input. This solution can significantly increase the efficiency of intelligent decision support system and its maintenance. This DT can be employed as an element of control at selecting a desired alternative (see Fig. 14), which was implemented in the intelligent information system of choice 'Optimel' (see Fig. 14) [18].

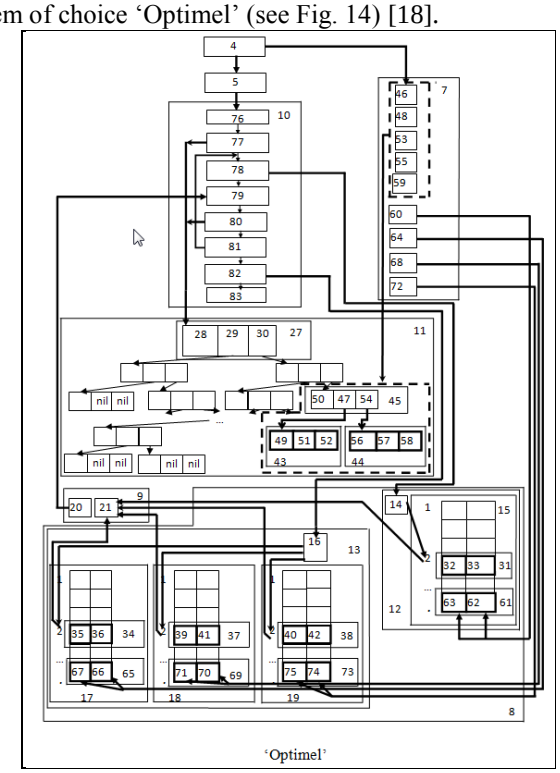


Figure 14: The intelligent information system of choice 'Optimel'

In Fig. 14, the suggested DT is block 11. Block 12 is a database of questions identifying properties. Block 13 is databases of answers and citations to the resources about the found solution methods and their applications to the problems of the given type.

The derived intelligent DSS can be applied for choosing the most suitable optimization method without additional fitting and saturation by optimization methods, as the very tree represents a system of ideal knowledge renewable by new methods. These potential for adding methods have not been obtained or publicized yet. This is especially important for pilot studies which help explore the subject field, identify existing solution methods for the given problem and make a decision in the course of study.

This principle of DT deriving can also be used for automating pilot studies that help identify the solution method for the problem of a definite scientific field. Here the DT represents all known solution methods for scientific problems of a definite scientific field applied for similar situations.

To this end, for construction of the required DT the suggested rules of deriving root and intermediary nodes should be used. To construct an intelligent DSS, different implementation techniques that help structure knowledge in the above-mentioned way can be used.

4. Conclusion

The suggested DT will allow simplifying designing knowledge base. It is possible due to combining some stages (choice of representation of knowledge base, designing knowledge base structure and developing algorithm of knowledge search) and excluding others from the designing process (developing fitting algorithm for knowledge base and its filling).

Roles of the developer of IDSS and of the expert in designing such system with the new DT can be combined, which considerably saves time on development, reduces errors, simplifies testing process of the obtained knowledge base and decreases expenses on the development of IDSS.

The suggested way of information structuring for the DT allows simplifying the decision search algorithm in the knowledge base. This will allow using a variety of ready-made intelligent systems with their databases to produce the IDSS, where knowledge base can be formed by the power of an intelligent system with the help of suggested structuring principles, e.g. the Moodle learning environment [19, 20].

The designed IDSS with such DT enables the user to directly address the knowledge base without turning to the expert. The user may refer to the expert in case of contributing the developed knowledge to the subject field in the tree.

The suggested way of the DT representation corresponds to the new way of meta-knowledge representation, i.e. knowledge about knowledge in the knowledge base determining rules, precedents and subject knowledge.

5. Acknowledgements

Funding: This work was supported by the Russian Foundation for Humanities [grant number 16-03-00382 as of 18th February 2016 issue: 'Monitoring of research activities of educational institutions in the information society'].

References

[1] *M. Bramer, Principles of Data Mining, Springer-Verlag, London, 2007. doi:10.1007/978-1-84628-766-4.*

[2] *L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and regression trees, Wadsworth & Brooks / Cole Advanced Books & Software, Monterey, 1984.*
[3] *L. Breiman, "Bagging Predictors," Machine Learning, 24, pp. 123–140, 1996.*
[4] *H. Deng, G. Runger, E. Tuv, "Bias of importance measures for multi-valued attributes and solutions," Proceedings of the 21st International Conference on Artificial Neural Networks (ICANN), pp. 293 – 300, 2011.*
[5] *J.H. Friedman, Stochastic gradient boosting, Stanford University, 1999.*
[6] *T. Hastie, R. Tibshirani, J.H. Friedman, The elements of statistical learning: Data mining, inference, and prediction, Springer Verlag, New York, 2001.*
[7] *Tamás Horváth, Akihiro Yamamoto, "Inductive Logic Programming," Lecture Notes in Computer Science, p. 2835, 2003. doi:10.1007/b13700.*
[8] *L. Hyafil, R.L. Rivest, "Constructing Optimal Binary Decision Trees is NP-complete," Information Processing Letters, V (1), pp. 15-17, 1976. doi: 10.1016/0020-0190(76)90095-8.*
[9] *G.V. Kass, "An exploratory technique for investigating large quantities of categorical data," Applied Statistics, XXIX (2), pp. 119-127, 1980. doi: 10.2307/2986296.*
[10] *S. Murthy, "Automatic construction of decision trees from data: A multidisciplinary survey," Data Mining and Knowledge Discovery, II(4), pp. 345-389, 1998. doi: 10.1023/a:1009744630224.*
[11] *O.B. Popova, B.K. Popov, Fundamental'nye issledovanija (fundamental research), XI(5), p. 1201-1205, 2012. [Online]. Available: http://www.fundamental-research.ru/ru/article/view?id=30734*
[12] *O.B. Popova, B.K. Popov, Sovremennye problemy nauki i obrazovanija (Modern problems of science and education), 5, p. 132, 2012. [Online]. Available: http://www.rae.ru/meo/?section=content&op=show_article&article_id=4259*
[13] *O.B. Popova, B.K. Popov, V.I. Kljuchko, Sistemnyj analiz processa vybora metoda optimizacii informacionnoj sistemy: monografija (System analysis of the process of selecting a method of optimization of information systems: monograph), FGBOU VPO «KubGTU», OOO «Izdatel'skij Dom-Jug», Krasnodar, 2012.*
[14] *O.B. Popova, B.K. Popov, Svidetel'stvo o gosudarstvennoj registracii programmy (State registration certificate program), № 2012615868 ot 27.06.2012.*
[15] *O.B. Popova, B.K. Popov, V.I. Kljuchko, Sovremennye problemy nauki i obrazovanija (Modern problems of science and education), III, 2013. [Online]. Available: http://www.science-education.ru/109-9146.*
[16] *O.B. Popova, B.K. Popov, V.I. Kljuchko, Fundamental'nye issledovanija (fundamental research), VI(1), pp. 55-59, 2013. [Online]. Available: http://www.fundamentalresearch.ru/ru/article/view?id=31413.*
[17] *O.B. Popova, B.K. Popov, V.I. Kljuchko, Binarnoe derevo vybora znanija iz oblasti znanija, ispol'zuja sistemu voprosov i otvetov. Teorija i praktika: monografija (A binary tree of the knowledge of the selection field of knowledge, using a system of questions and answers. Theory and practice: a monograph), FGBOU VPO «KubGTU», OOO «Izdatel'skij Dom-Jug», Krasnodar, 2013.*
[18] *O.B. Popova, B.K. Popov, Patent na izobretenie (The patent for invention), RUS №2564641 ot 27.05.2014.*
[19] *O. Popova, B. Popov, V. Karandey, M. Evseeva, "Intelligence amplification via language of choice description as a mathematical object (binary tree of question-answer system)," Procedia – Social and Behavioral Sciences, 214, pp. 897–905, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877042815061030.*
[20] *O. Popova, B. Popov, V. Karandey, "Intelligence Amplification in Distance Learning through the Binary Tree of Question-answer System," Procedia – Social and Behavioral Sciences, 214, pp. 75–85, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877042815059522.*
[21] *J. R. Quinlan, "Induction of Decision Trees," Machine Learning, Kluwer Academic Publishers, 1, pp. 81–106, 1986.*
[22] *F.P. Vasilyev, Metody optimizacii (Optimization methods), Publishing house «Factorial Press», Moscow, 2002.*
[23] *Urok 3. Strukturirovanie informacii (Lesson 3. Structuring information). [Online]. Available: http://4brain.ru/memory/strukturirovanie.php, 09.01.2016.*
[24] *I.H. Witten, E. Frank, M.A. Hal, Data Mining: Practical Machine Learning Tools and Techniques (3rd Edition), Morgan Kaufmann, 2011.*

Author Profile

Olga Popova is an Assistant Professor in the Department of Information Systems & Programming at Kuban State Technological University in Krasnodar, where she has been a faculty member for 16 years. All this time, along with teaching, she has been doing extensive research and writing a Doctorate Thesis. Olga is the author of 90 publications on various issues in system analysis which are rated in Russian Science Citation Index (RSCI) where her h-index is 11. Completed her undergraduate studies at Kuban State Technological University with honors, receiving a diploma in electrical engineering, being the top graduate in her class, and in 2002 she received Ph.D. in technical sciences from the same university. Her thesis covered mathematical modeling and optimization of special electromechanical systems. Research interests lie in the area of complex system analysis with focus on Decision Support Systems (DSSs) and information structuring. In recent years, she has been exploring artificial intelligence enhancement methods approximated to those used by natural intelligence.

Marina Evseeva is a senior lecturer at the Department of Applied Linguistics & New Information Technologies at Kuban State University in Krasnodar, where she has been a faculty member for 5

years. She teaches Theoretical Phonetics and English to BA undergraduates. In 1996, she completed with honors her undergraduate studies in English philology at Kuban State University, and in 2005 received Ph.D. in Cultural Linguistics from Volgograd State Pedagogical University, thesis 'The Concept of 'Friendship' in the English and Russian Linguistic Cultures'. Marina's research interests are mainly in the area of cultural and applied linguistics, the ESL teaching methodology with focus on how to create a dynamic and interactive classroom applying appropriate teaching methods and strategies that stress learning beyond textbook. At present, as part of a research team, Marina Evseeva is involved in carrying out research on Decision Support Systems (DSSs) and artificial intelligence enhancement methods in education.