# Algebraic Bayesian Networks: Probabilistic-Logic Inference Algorithms and Storage Structures

Ekaterina Malchevskaya, Nikita Kharitonov
Theoretical and Interdisciplinary Computer Science Lab, SPIIRAS
St. Petersburg, Russia
katerina.malch@gmail.com, nikspb95@gmail.com

Andrey Zolotin, Anastasia Birillo
St. Petersburg State University
St. Petersburg, Russia
andrey.zolotin@gmail.com, nbirillo@mail.ru

***Abstract*—This article presents math library and relational database, being components of software complex, that implements latest theoretical results in the field of Algebraic Bayesian Networks storage structures and probabilistic-logic inference algorithms. A review of existing software implementations of given algorithms is performed and a number of deficiencies that are present in them is identified. The description of probabilistic-logic inference tasks and the corresponding public methods of the mathematical library is proposed in the article. The structure of the math library is represented by a class diagram and supplemented by a propositional formula parser algorithm description. A comparison of existing database solutions based on their domain area is conducted and the substantiation of the choice is provided. The structure of the database is described in detail and a parallel between the theoretical objects and database tables is drawn. Moreover problems that have arised during the development are considered and an appropriate solution is provided.**

## I. Introduction

Algebraic Bayesian networks (ABN) are a class of probabilistic graphical models (PGM), which are in turn a representation of knowledge bases with uncertainty. Another well-known representative of PGM, Bayesian belief network, is widely used in a number of areas - pattern recognition systems [1], some areas of medical research [2], [3], ecological forecasting [4], [5] and decision support systems [6].

Modern methods of collecting information allow to obtain a huge amount of data at a time due to the increased capacity of storages, the data transmission rate as well as the widespread of information networks. However, the amount of data received and present uncertainty makes it sometimes impossible to perform a quick analysis of the whole dataset. This flaw of existing models boosts the development of new knowledge representation models and algorithms for their processing.

## II. Existing solutions

Introduced by professor V.I. Gorodetski in 1993 ABN allow to store and process huge volumes of data with uncertainty due to the decomposition of data into small bundles composed of closely related facts - knowledge patterns (KP) [7]. However, the theory development and formalization of local and global ABN structures as well as probabilistic logic inference machine optimization led to the need to progress the development of the software package containing all mathematical results that would allow researchers to carry out the computational experiments and analyze the results.

Let us consider several software solutions for computer-aided presentation of algorithms and structures of ABN theory. In 2008 professor Tulupyev A.L. has implemented software package in Java, including a database storing ABN knowledge patterns with their elements probability estimates and other meta-information [8]. However later in 2009 Sirotkin A.V. has introduced basic matrix-vector equations in probabilistic-logic inference that caused a massive reconstruction of software complex taking improved theory into account. The proposed improvements were implemented by Sirotkin A.V. in the software package in C ++ that worked with the same type of database [9].

However, the ABN theory development and final rendering of inference algorithms to matrix-vector equations boosts the creation of a new software complex that implements the latest algorithms in order to reduce the processing time and increase the results accuracy [10]. In recent years, several studies have been conducted in the area of KP local structure, that resulted into the new models representing KP in ABN, such as the ideal of disjuncts and set of quanta propositions [11]. These results allowed to increase the theoretical base by adapting probabilistic-logic inference algorithms for alternative KP thus increasing the attractiveness of ABN as a model to process big data [11], [12]. On the other hand the increased variety of KP alternative models lead to the need to expand the database structure and to rework the data access layer, as everything developed previously was based on the ABN model containing only KP over ideals of conjuncts and proposition quanta.

This paper proposes an approach to the design and implementation of the software package that takes all the latest studies in the ABN theory into account. To achieve this goal we compare different databases and related technologies and choose the best one taking into account the specifics of the domain as well as design and implement the mathematical library class structure, including algorithms for probabilistic-logic inference in ABN.

Developed software allows one to conduct computational experiments necessary to test new assumptions as well as to write educational papers. In addition, the implementation of algorithms, taking into account the principles of OOP, makes it possible in the future to use the operating data as a plug-in.

## III. MATH LIBRARY TASKS

Let us fix a finite set of atomic propositional formulae (ordered alphabet of atoms). Atoms in alphabet are considered to be elementary statements given by experts about domain, where validity of each statement can be estimated. Each atom can be set to true or false.

Conjunct is a conjunction of some set of atoms where atoms come from a given alphabet. Ideal of conjuncts is a set of all possible conjuncts over the given alphabet.

Algebraic Bayesian networks are represented by an undirected graph with ideals of conjuncts in nodes. However there are alternative ways to presenting knowledge patterns in ABN theory. They are knowledge patterns under ideal of disjuncts or quanta propositions. Quant is a chain of all atoms in given alphabet, where atoms can be with negation. Disjunct is a disjunction of atoms of a given alphabet.

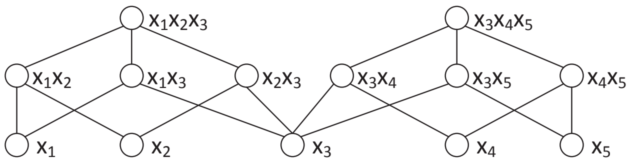


Fig. 1. Algebraic Bayesian network

Elements of KP can have either scalar or interval probability estimates. Elements of ideal of conjuncts together with assigned probability estimates are named KP. Previously, it was proved that all three models are closely related to each other [11] and a transition formula for alternative KP model was proposed. This formula allows to switch from one model to another, as well as to perform operations of PLI regardless of a given model.

It is possible to perform a number of operations with ABN that all together are named probabilistic-logical inference. There are several kinds of probabilistic-logic inference, in particular the global and the local inference. During the global inference all operations are performed on the entire network, while in the local inference we operate only on one specific KP from the entire ABS.

The designed math library currently allows to perform only local PLI over a single KP.

The PLI tasks also include the consistency check of the local probability estimates for the selected KP and local priori and a posteriori inferences tasks.

The priori inference task is to calculate the probability of an arbitrary propositional formula, given on the same alphabet as the original KP, using KP elements probability estimates.

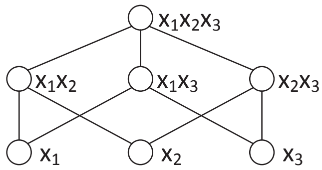


Fig. 2. Knowledge pattern over three atoms

In case of posteriori inference some new information that is called an evidence is transmitted into the KP. The first task of posteriori inference is to evaluate the probability of the received evidence using KP elements probability estimates, while the second task is to recalculate the probability estimates in given KP taking into account the new information received within evidence.

One of the important tasks which is solved within PLI is parsing of random propositional formula that is specified over the same alphabet as ABN. This problem appears when one need to perform a PLI with a formula that is not specified over a set of quanta propositions. As far as mathematical library should work with this data representation the mentioned parser was created to handle the evidences. It converts string containing variables and boolean operations into characteristic vector of set of quanta propositions.

The main idea of a parser is to transform a logical statement into a disjunctive normal form (DNF) over a set of quanta propositions. The DNF is formed in 2 steps. First the string parse tree is constructed, where the nodes contain logic operations, and leaves contain variables. A modified reverse Polish notation algorithm is used to construct this tree. Secondly deep depth-first search is used to get all sets of variables for which the formula is true. Finally to obtain the DNF variables or their negation in each such set are connected by logical multiplication operations and the resulting quants connected by disjunctions.

Characteristic vector of quants is built on the basis of the PDNF using the following principle: if the considered quant is included in DNF then appropriate element of characteristic vector is set to 1, otherwise it is set to 0. To obtain the resultant conjuncts characteristic vector the quants characteristic vector is multiplied on the left by the matrix **I** [13].

In addition to performing the basic tasks, the parser searches ABN alphabet for the variables from the input string that in total allows us to control the correctness of input data. Moreover it handles different types of notations of the same logical operations that makes the library more flexible.

## IV. MATH LIBRARY METHODS AND STRUCTURE DIAGRAM

Math library is implemented in C# programming language. This enabled us to use the existing tools adapted for C#, such as lp_solve55 library [14], for solving the linear programming problems that appear during the interval KP consistency checking and maintenance as well as in priori and posteriori inference tasks. In addition, the choice of programming language was caused by the necessity of further integration with the project, aimed at the visualization and research of primary, secondary and other global structures of ABN, that is also implemented in C #.

Structure of the math library consists of several modules: module that allows to create knowledge patterns and evidences of different types assigning scalar and interval probability estimates to their elements; services to solve the P-LI tasks one of which performs the KP elements probability estimates consistency check and maintenance as well as solves the prior inference task for the specified propositional equations, while another one solves both tasks of posteriori inference and finally the facade module that provides interfaces to make the interactions with library more clear.

Below one a facade module class diagram can be found. It covers all basic functionality provided by library for KP processing including priori inference, posteriori inference as well as consistency check and maintenance. Diagram is presented in Fig. [3].

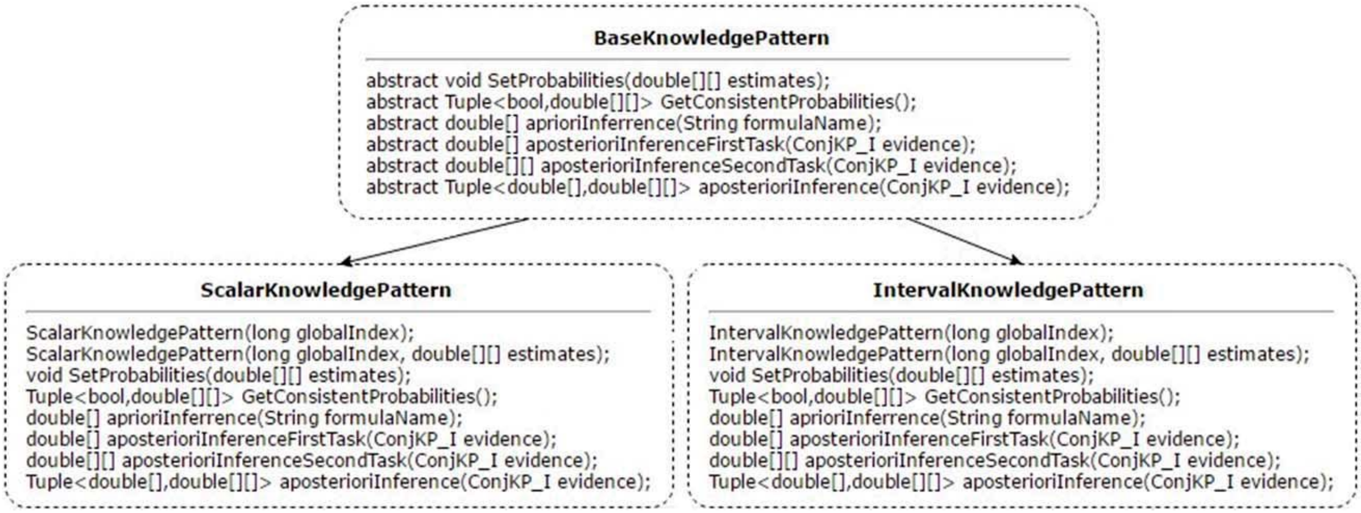


Fig. 3. Class diagram for the facade module of library

In this structure, the basic principles of OOP are used, in particular polymorphism. Thus, the abstract class BaseKnowledgePattern contains the description of methods that are common to both derived classes - ScalarKnowledgePatern implementing KP model with scalar probability estimates and IntervalknowledgePattern implementing KP model with interval probability estimates where all abstract methods are implemented in an appropriate way.

There are several methods that are common for all classes implementing KP models: method setting probability estimates for KP elements (SetProbabilities), that receives the input array with scalar estimates or two arrays with lower and upper boundaries of interval probability estimates respectively; second method that maintains consistency of the KP probability estimates and then returns the estimates or errors in the case of the KP was initially inconsistent (GetConsistentProbabilities); third method that solves the priori inference task (AprioriInferrence) and the last 2 methods solving 2 problems of posteriori inference (AposterioriInferenceFirstTaskи and AposterioriInferenceSecondTask respectively). The method that solves priori inference task accepts propositional formula represented as a string on input, while methods that solve posteriori inference tasks accept evidences.

Method AposterioriInference accepts the evidence on an input, and, in turn, solves both problems of a posteriori inference. As a result this method returns a couple formed of 2 results - one for every problem.

Besides this, derivative classes have two constructors that allow one to create a KP of required type with the specified global index and elements probability estimates.

It should be noted that all the functionality of the described facade module is covered by unit tests.

## V. Database structure

### *A. The choice of technologies*

The relational Microsoft SQL Server database is used in the considered project [15]. For several reasons listed below the relational database management system (RDBMS) has been chosen instead of NoSQL.

Firstly, comprehensive project works with such objects as ABN, KP, evidence and propositional formula that have a complex and strict structure. It is convenient to store such structures in tables, since this type of storage gives the possibility to obtain objects partially and perform further actions with the obtained data (for example get a specific KP from ABN). The developed database presumes a storage of a large amount of data (due to the definition of the presented structures and the specifics of use). But at the same time, complex analysis will be performed on the data, which can be achieved by means of good relational database query optimizers. In addition, modern relational databases, such as SQL Server [15] support the work with big data and provide efficient storage of a large amount of information.

Secondly, the application being developed is designed to be used during a long period of time and should be able to store the intermediate result in the system. At the same time it is necessary to guarantee the fulfillment of transactions in order to avoid internal errors and conflicts. NoSQL technology is relatively young, so the operations with transactions are not stable enough there. For example, the currently widespread NoSQL database monoDB [16] from Oracle supports atomicity of operations only on one level (in one document). But the simultaneous modification of several documents is autonomous, which is not suitable for the above described structure of ABN.

Moreover, the chosen technology supports high-speed data write and update operations that are undoubtedly important when working with large amounts of tabular data. Furthermore, relational databases are fairly well-studied technology, so there are now many documented both paid and free supported solutions.

Currently there are several popular RDBMS such as MySQL [17], SQL Server [15] and PostgreSQL [18]. The main choice had to be made between SQL Server and PostgreSQL, since MySQL is currently not supported. Comparison of SQL Server and PostgreSQL showed that both systems have high reliability, security and performance. Both mentioned RDBMS allow us to work with the database using the .NET framework language, that is important, since the implementation of the mathematical library is written in C#. The main difference between these technologies is a price of usage - SQL Server provides the free license supported by the student's DreamSpark program, supplemented by a free hosting on Microsoft Azure server, while PostgreSQL provides open source code even for commercial purposes, but does not provide an opportunity to host a database for free. Since both RDBMS meet the necessary requirements for the project, but the SQL Server provided us with a free hosting, the last was chosen for the project needs.

*B. Problems and solutions*

As mentioned in the second chapter, ABN consists of a KP set that may be constructed above an ideal of conjuncts, ideal of disjuncts or a set of quanta propositions. KP elements probability estimates could be either interval or scalar, but in all cases the common structure of ABN remains the same, while the data processing algorithms may vary. To solve the problem of type unification the KP type was moved out to the ABNTypeEnum and KPTypeEnum tables that store information about the KP type on the ABN level. In this case there is no information duplication because the theory assumes that a single ABN cannot contain KP of different types.

To preserve the consistency of structure for KP with interval and scalar probability estimates it was decided not to separate these cases as the scalar probability estimates is a special case of interval estimates with lower and upper bounds being equal. To avoid a duplication of estimates in case of scalar estimates the upper bound stores an empty value (NULL).

Since an ABN consists of several KP connected with each other, where the intersection of KP (separator) is also a KP built over the elements belonging to both KP, there is a problem of duplication of elements belonging to the separator. To solve this problem, it was proposed not to store the KP themselves, but the elements of which they consist (conjuncts, disjuncts and quanta propositions).Thus, the KP contains references to the elements that form it and and the same time elements of the separators are referenced by multiple KP.

When storing the alphabet over that an ABN is built one must take into account the order of the elements for the correct operation of algorithms. It was decided not to introduce an additional field with a sequence number of the alphabet element, because each symbol has a unique identifier (id), which will keep the right sequence of elements of the alphabet, using ascending sort by the element id.

Propositional formulas necessary for a priori inference, are represented as PDNF and are formed of a set of elements similar to PatternItems. At the same time the probability estimates of PatternItem elements (scalar values in case of scalar KP and pairs of lower and upper limits in case of the interval KP) in propositional formula and in KP, where the inference is conducted should be the same. In this regard, an appropriate solution was not to duplicate the PatternItem data for every propositional formula but refer to the desired item in the appropriate table to save memory and simplify working with data.

In the posteriori inference tasks 3 types of evidences are used: deterministic, stochastic and interval. Deterministic evidence consists of the elements with the probability estimates equal to either 0 or 1, so this type of evidence can be conveniently represented with a binary mask, rather than stored as a full evidence object, that takes much more space. In order to improve system performance and minimize the memory size required to store such evidences the EvidenceTable table was extended with a LocalIndex field that holds a value - a binary mask or an empty value (NULL) in case of stochastic or interval evidence.

*C. Database schema*

Figure [4] shows the developed database structure that stores the information about ABN.

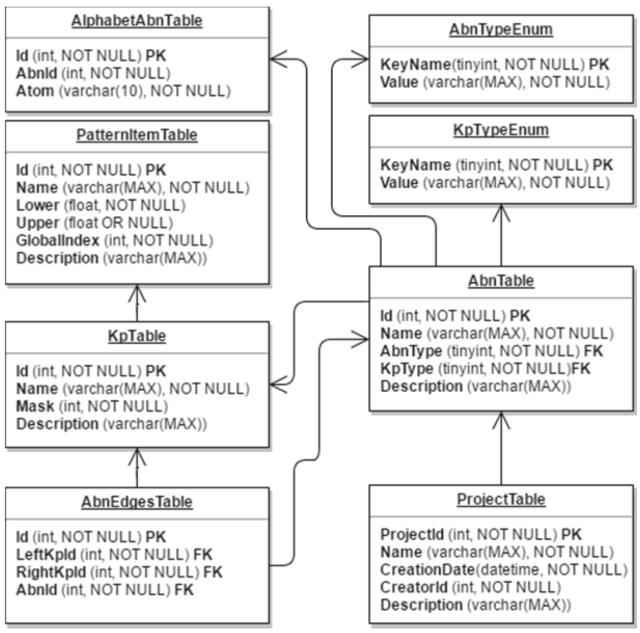


Fig. 4. Structure of algebraic Bayesian network

All ABN are combined into projects, where projects are presented in the table ProjectTable. Each project has a unique identifier (ProjectId), a name (Name), a creation date (CreationDate) and a project creator (CreatorId). The proposed grouping system will help us to achieve the best data representation when implementing user interface.

Alphabet (over which ABN is constructed) is stored in table AlphabetAbnTable element-by-element. Each atom of the alphabet has unique identifier (Id), unique identifier of ABN (AbnId) to which it belongs and name of the element (Atom) which length is restricted to 10 symbols.

AbnTypeEnum and KpTypeEnum tables have identical structure (as well as all tables with prefix Enum). These tables are the dictionaries that associate every number of type integer to its text equivalent. Table AbnTypeEnum contains information on what type of elements the considered ABN is based on (ideal of conjuncts, ideal of disjuncts or set of quanta propositions). Table KpTypeEnum characterizes the type of vectors containing the probability estimates of KP elements (scalar or interval values).

In the database ABN is represented by the AbnTable table, that contains properties such as a unique identifier (Id), a name (Name) and references to the tables AbnTypeEnum and KpTypeEnum (AbnType and KpType respectively).

KP is represented by the KpTable table and has a unique identifier (Id), the name (Name) and global index - the mask (Mask), which allows us to determine the subset of ABN alphabet atoms on which this KP is built.

Elements of KP (conjuncts, disjuncts and quanta propositions) are stored in the PatternItemTable table since the general structure is as stated earlier in the section C of this chapter. They are characterized with a unique identifier (Id), a name (Name), lower and upper probability bounds (Lower and Upper accordingly) and a global index (GlobalIndex) which is an analog of the Mask property in KP. The GlobalIndex allows to define the subset of atoms from the KP alphabet over that the PatternItem is constructed.

ABN can also be represented in the form of secondary structure [19]. In paper [20] the software implementation of the developed algorithms to work with the ABN secondary structure is provided. The application can also generate a set of minimal joint graphs for the given secondary structure. Table AbnEdgesTable stores one of such structures according to the user's choice. Joint graphs in this case are represented as a set of edges, where each edge has a unique identifier (Id), references to the left and right vertex - KP that are connected to this edge (LeftKpId and RightKpId accordingly) and a unique identifier of ABN to which this joint graphs edge belongs (AbnId).

Fig. 5 shows structure of the evidence. Type of evidence (deterministic, stochastic or imprecise) is stored in the table EvidenceTypeEnum. Table EvidenceTable similarly contains basic information about the evidence: unique identifier (Id), unique identifier of ABN that it belongs to (AbnId), name of the evidence (Name), global index that allows us to define a subset of alphabet over which the evidence is constructed (GlobalIndex), reference to the table EvidenceTypeEnum that helps to identificate the type of evidence (Type) and local index (LocalIndex) introduced for the determined evidence. Stochastic and interval evidences being complete analogs of PatternItems consist of elements that are stored in EvidencePatternItemTable table.

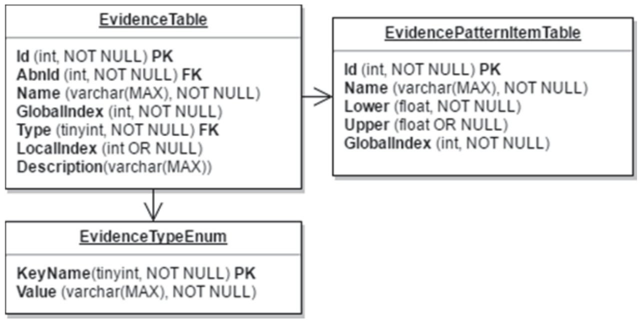


Fig. 5. Structure of evidence

Propositional formulae for priori inference in ABN are also stored in the database. Figure [6] shows the structure of the propositional formula. Table EquationTable contains information about the propositional formula. The formula is stored as a set of PatternItems into which it is separated by means of the developed parser described above. Propositional formula is characterized by unique identifier (Id), unique identifier of ABN which it belongs to (AbnId), name of formula (Name) and it also has the UserView field to store a string representation of a formula, that user has entered. It is necessary for inverse mapping of a formula to an original view, that it is impossible to determine a type of the initial formula by PDNF unequivocally.

Table EquationPatternItemTable contains the set of PatternItems into which propositional formula was partitioned. Due to the fact that the ABN theory implies identical values of probability estimates in same PatternItems in priori inference and PatternItems in KP of considered ABN, the table EquationPatternItemTable comprises references to the PatternItemTable and it has two properties: unique identifier (Id) and reference to the unique identifier of the necessary element (PatternItemId).

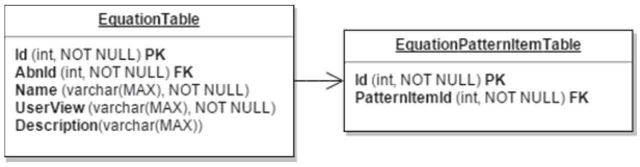


Fig. 6. Structure of propositional formula

"One-to-many" relations, for example, "set of KP at specific ABN" are realized through additional tables with prefix Link, that contain two fields - unique identifiers of the 2 linked objects. These tables excluded from the schemes represented at Fig. 4, 5, 6 to make the representation compact.

Objects such as ABN, KP, PatternItem, evidence, propositional formula and project have an optional description field for the comment.

## VI. Conclusion

Mathematical library that implements algorithms of probabilistic-logic inference in the ABN and the database that allows to store the main objects of the ABN theory are considered in the paper. The proposed software implementation of math library takes into account the latest results in the ABN theory, such as alternative models of knowledge patterns (disjuncts, quanta propositions) and matrix-vector operations that were introduced in probabilistic logic inference algorithms. The developed mathematical library can be used for numerical experiments and as a plug-in application module that can be accessed using public methods of the library. Through adherence to the basic principles of the object oriented programming (OOP) in the design and development of the library, it is easily extensible with new methods and classes [20]. We used unit tests to verify the correct functioning of the developed library. Tests cover all the functionality of the library.

Database that is presented in the paper is the elaboration of the existing solution with ABN theoretical apparatus development taken into account. It allows to conveniently store such objects as the KP, the sets of elements on that the KP is built (conjuncts ideal, the ideal of disjuncts and quanta propositions), the evidences for the posteriori inference in ABN as well as propositional formulae needed for the a priori inference. Also in the database secondary structure of ABN is considered being presented as the minimal joint graph. The developed database is flexible in terms of the types of stored objects are not tied to a specific type of KP that allows us to easily extend it as a part of the further development process in the area of ABN theory.

Mathematical library and database, described in this article, create a foundation for the development of the ambient software package as well as provide an opportunity for a larger number of numerical experiments, developing the algebraic Bayesian networks theory in particular and the area of artificial intelligence in general.

The developed mathematical library and database is currently implemented in a web project AlgBNWebApp, which allows working with ABN in online mode [22].

## Acknowledgment

The paper presents results of the project partially supported with RFBR grant 15-01-09001-a "Combined Probabilistic-Logic Graphical Approach to Representation and Processing of Uncertain Knowledge Systems: Algebraical Bayesian Networks and Related Model".

## References

[1] P. X. Peng, Y. H. Tian, Y. W. Wang, J. Li, T. J. Huang, "Robust multiple cameras pedestrian detection with multi-view Bayesian network", *Pattern Recognition*, 2015, № 42, pp. 1760-1772.

[2] X. Wu, X. Y. Wen, J. Li, L. Yao, "A new dynamic Bayesian network approach for determining effective connectivity from fMRI data", *Neural Computing & Applications*, 2014, vol. 24, no. 1, pp. 91–97.

[3] K. F. R. Liu, C. F. Lu, C. W. Chen, Y. S. Shen, "Applying Bayesian belief networks to health risk assessment", *Stochastic Environmental Research and Risk Assessment*, 2012, vol. 26, no. 3, pp. 451–465.

[4] M. A. E. Forio, D. Landuyt, E. Bennetsen, K. Lock, T. H. T. Nguyen, M. N. D. Ambarita, P. L. S. Musonge, P. Boets, G. Everaert, L. Dominguez-Granda, P. L. M. Goethals, "Bayesian belief network models to analyse and predict ecological water quality in rivers", *Ecological Modelling*, 2015, vol. 312, pp. 222–238.

[5] J. V. Murray, K. E. Stokes, R. D. van Klinken, "Predicting the potential distribution of a riparian invasive plant: the effects of changing climate, flood regimes and land-use patterns", *Global Change Biology*, 2012, vol. 18, no. 5, pp. 1738–1753.

[6] J. Borras, A. Moreno, A. Valls, "Intelligent tourism recommender systems: A survey", *Expert Systems with Applications*, 2014, vol. 41, no. 16, pp. 7370-7389.

[7] V. I. Gorodetski, "Algebraic Bayesian networks — a new paradigm of expert systems", *Anniversary collection of proceedings of Department of Informatics, Computer Science and Automation of RAS*, 1993, vol. 2, iss. 1, pp. 120–141. (In Russian)

[8] A. L. Tulupyev, "Algebraic Bayesian networks: an implementation of probabilistic-logic inference in a system of Java-programs", *SPIIRAS Proceedings*, 2009, vol. 8, p. 191-232. (In Russian)

[9] A. V. Sirotkin, "Program complex for probabilistic logic inference in knowledge pattern bases: knowledge pattern implementation", *SPIIRAS Proceedings*, 2013, vol. 25, no. 2, p. 204-220. (In Russian)

[10] A. L. Tulupyev, A. V. Sirotkin, A. A. Zolotin, "Matrix equations for normalizing factors in local a posteriori inference of truth estimates in algebraic Bayesian networks", *Vestnik of the St. Petersburg University: Mathematics*, 2015, vol. 48, no. 3, pp. 168-174. (In Russian)

[11] A. A. Zolotin, A. L. Tulupyev, A. V. Sirotkin, "Matrix-vector algorithms of local posteriori inference in algebraic bayesian networks on quanta propositions", *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2015, vol.15, no. 4, pp. 676–684.

[12] A. A. Zolotin, E. A. Malchecvskaya, "Matrix-vector algorithms of local posteriori inference in algebraic Bayesian networks on ideal of disjuncts", *XIX IEEE International Conference on Soft Computing and Measurements (SCM)*, 2016, pp. 31-34. (In Russian)

[13] A. L. Tulupyev, A.V. Sirotkin, S. I. Nikolenko, *Bayesian belief networks: probabilistic-logic inference in the acyclic directed graphs*. SPb: SPbGU, 2009, 400p.

[14] Sourceforge lp_solve reference guide, lp_solve 5.5, Web: http://lpsolve.sourceforge.net/.

[15] Microsoft SQL Server documentation, Web: https://msdn.microsoft.com/ruru/library/mt590198(v=sql.1).aspx.

[16] MonoDB documentation, Web: https://docs.mongodb.com.

[17] MySQL Documentation, Web: https://dev.mysql.com/doc/.

[18] PostgreSql Documentation, Web: https://www.postgresql.org/docs/.

[19] A. A. Filchenkov, "Hierarchy of algebraic bayesian network global structures as a system of graphs and hypergraphs", *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2013. № 1. pp. 75–81.

[20] A. I. Berezin, A. V. Romanov, M. A. Zotov, "Algebraic Bayesian Networks Secondary Structure Synthesis: All possible Minimal Join Graphs' Set Capacity Estimate", *International Conference on Soft Computing and Measurements (SCM)*, 2016, pp. 76-78.

[21] E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. USA.: Addison-Wesley, 1994. 395p.

[22] A. A. Zolotin, D. G. Levenets, E. A. Malchevskaya, M. A. Zotov, A. I. Birillo, A. I. Berezin, A. V. Ivanova and A. L. Tulupyev, Algorithms for processing and visualization of algebraic Bayesian networks.(In Russian) [in press]