ITHEA

# International Journal INFORMATION MODELS & ANALYSES

2020 Volume 9 Number 1

**International Journal**
**INFORMATION MODELS AND ANALYSES**
**Volume 9 / 2020, Number 1**



**IJ IMA is official publisher of the scientific papers of the members of the ITHEA® International Scientific Society**

IJ IMA rules for preparing the manuscripts are compulsory.
The **rules for the papers** for ITHEA International Journals are given on www.ithea.org .
The camera-ready copy of the paper should be received by ITHEA® Submission system http://ij.ithea.org .
Responsibility for papers published in IJ IMA belongs to authors.

**International Journal "INFORMATION MODELS AND ANALYSES" Volume 9, Number 1, 2020**

Edited by the **Institute of Information Theories and Applications FOI ITHEA**, Bulgaria, in collaboration with:
University of Telecommunications and Posts, Bulgaria,
V.M.Glushkov Institute of Cybernetics of NAS, Ukraine,
Universidad Politécnica de Madrid, Spain,
Hasselt University, Belgium, University of Perugia, Italy,
Institute for Informatics and Automation Problems, NAS of the Republic of Armenia
St. Petersburg Institute of Informatics, RAS, Russia,

Publisher: **ITHEA®** Sofia, 1000, P.O.B. 775, Bulgaria. www.ithea.org , e-mail: office@ithea.org
Technical editor: **Ina Markova**

**Printed in Bulgaria**



**ISSN 1314-6416 (printed)** **ISSN 1314-6432 (Online)**

# QUALITY OF EXPERIENCE MODELING OF MULTIMEDIA ON-LINE SERVICES

**Zlatinka Kovacheva, Stoyan Poryazov, Emiliya Saranova**

***Abstract**: This paper is focusing on some important aspects of QoE modeling of multimedia on-line services as influence factors and parameters, assessments and models. On the base of this survey we are going to propose a conceptual model for QoE prediction and to apply it for multimedia on-line services.*

***Keywords**: quality of experience, quality of service, modeling, prediction*

***ITHEA Keywords**: I.6.5. Model Development*

## Introduction

During the past few decades service quality has become a major area of attention to practitioners, managers and researchers owing to its strong impact on business performance, lower costs, customer satisfaction, customer loyalty and profitability [Seth 2005].

Quality of experience (QoE) of the multimedia on-line services is a subjective measure which depends on a variety of factors as network quality of service (QoS), users' experience, interest and expectations, cognitive and behavioural states, costs, etc. [Mitra 2018]. The term QoS refers to the ability of the network to achieve a more deterministic behaviour, so data can be transported with a minimum packet loss, delay and maximum bandwidth but QoS does not consider the user's perception [Alreshoodi 2013]. Obviously, the QoS is a key factor for the QoE.

QoE is a multidisciplinary and a multidimensional concept. We may outline the following main challenges to investigate and predict the QoE:

- To build a **conceptual model**, including a variety of interrelated parameters with nonlinear relations;
- To use relevant **methods** for modeling in order to apply the model for QoE measuring and prediction;
- The model should be applicable in a dynamic environment for a **long time period.**

The aim of this paper is to focus on the following important aspects of QoE modeling of multimedia on-line services:

- QoE influence factors and parameters**;**
- QoE assessments;
- QoE models.

## 1. QoE influence factors and parameters

In the Qualinet White Paper on Definitions of Quality of Experience [Callet 2012], the main QoE influence factors are grouped in the following categories:

- **Human factors** (age, gender, education, background, etc.);
- **System factors** (bandwidth, security, resolution, etc.);
- **Context factors** (location, movements, costs, etc.)

Context can be static and dynamic. Static context may include user's application preferences, their security requirements and cost. Dynamic context can change in a very short period of time and it is uncertain. The timely collection and processing of context may be crucial as it may loose its accuracy. Dynamic context may include user location, velocity, network load, etc.

The QoE of the multimedia on-line services depends on many parameters which may be classified as context and additional.

The context parameters may be grouped in the following context classes [Mitra 2018]:

- User and user environment (location, social context, age, gender, background, etc.);
- Tools/device/object (design layout, resolutions, input/output methods, usability, etc.);
- Application (type, requirements);
- Network (bandwidth, delay, jitter, packet loss, protocols used, received signal strength, etc.).

The additional parameters include users' satisfaction, technology acceptance, enjoyment, efficiency, accuracy, perceived ease-of-use, etc. Studying and modelling these parameters is a challenging task.

In [Perkis 2006] the authors classified the technology and user related parameters as either quantifiable or unquantifiable. Quantifiable parameters include bandwidth, delay and jitter. Parameters such as expectation, attitude, ease-of-use are related to user and are deemed to be unquantifiable.

In reality, the **parameters are not independent** of each other. There can be inter-dependencies and non-linear relationships between them. For example, the parameter "user satisfaction" may affect the parameter "technology acceptance". In the model of Gong at al. [Gong 2009] each QoE parameter is a function (linear or ratio) of one or more QoS parameters. For example, "service integrity" is a function of "delay", "jitter" and "packet loss" ratio.

Further, some parameters may be hidden. i.e. they may not be observed directly. These parameters may be hard to measure and quantify.

QoE modelling and measurement require a combination of different kinds of parameters to determine overall QoE.

## 2. QoE assessments

There are two main quality assessment methodologies, namely **subjective** and **objective** assessment. The most commonly used subjective method for quality measurement is the Mean Opinion Score (MOS). MOS is standardized in the ITU-T recommendations [ITU-T Recommendation 2003], and it is defined as a numeric value going from 1 to 5 (i.e. poor to excellent). Usually user surveys are conducted to gather the subjective evaluation of a given service. A variety of demographics and context characteristics should be considered. The main drawbacks of this approach are: it is high in cost, time consuming, cannot be used in real time and lacks repeatability. These limitations have motivated the development of objective tools that predict subjective quality solely from physical characteristics.

The objective approach is based on mathematical and/or comparative techniques that generate a quantitative measure. This approach is useful for in-service quality monitoring or the design of networks/terminals, as well as in codec optimization and selection. The development, deployment, and modeling of the objective methods are difficult processes due to their large space parameters.

The objective approach is more reproducible, more predictable and more suitable for in-service usage for real-time service monitoring and adaptation, however very often it is less accurate than subjective methods.

Many of the objective methods convert the final results to the MOS scale. A combination of the objective and subjective approaches can be performed to overcome the shortcomings of each individual technique [Alreshoodi 2013].

The different parameters which define the overall QoE may be measured by different scales. E.g., the “user’s satisfaction” may be measured by the scale 1 to 5 but the “technology acceptance” may be measured by “yes/no”.

Mitra et al [Mitra 2018] proposed to use a bipolar interval scale to map users’ ratings into an interval scale. For example, a 5-point ordinal scale is calibrated in such a manner that the best alternative, for example, “excellent” is assigned a

maximum value, ’1’; the worst alternative on the other hand is assigned the lowest value, ’0’. The mid-point is also used for calibration. For example, “good” is assigned a value of 0.50. This means that values lower than 0.50 are less favourable compared to values higher than 0.50. For example, a value between 0.8750 and 1 is considered to be “excellent” while a value in the range of 0 and 0.1250 is considered as “poor”. This way normalized values can be used to determine a QoE rating. Thus, a bipolar scale enables an expert to perform mathematical operations such as computing mean and standard deviation and the application of parametric statistical models.

In [Brooks 2010] the existing approaches of measuring network service quality from a user perspective are classified into three categories:

- Testing User-perceived QoS (TUQ) - e.g. MOS;
- Surveying Subjective QoE (SSQ) – e.g. questionnaires;
- Modelling Media Quality (MMQ) – e.g. perceptual evaluation of speech quality.

The first two approaches collect subjective data from users, whereas the third approach is based on objective technical measurements.

It is possible to measure and quantify the QoE and subsequently derive a mapping correlating the QoS parameters with the measured QoE metrics. A number of objective models have been devised for estimating QoE [Alreshoodi 2013]. The International Telecommunication Union (ITU) has developed a classification to standardize these models based on a focus of each model type. Generally, the objective quality assessment methodologies can be categorized into five types [Takahashi 2008]:

- Parametric packet-layer model predicts QoE from packet-header information, without handling the media signal itself. It does not look at the payload information; therefore it has difficulty in evaluating the content dependence of QoE;

- Parametric planning model takes quality planning parameters for networks and terminals as its input. This type of model requires a priori information about the system under testing;
- Media layer model predicts the QoE by analysing the media signal via HVS. However, if media signals are not available, this type of model cannot be used;
- Bit-stream model is a new concept. Its position is in between the parametric packet-layer model and the media-layer model. It derives the quality by extracting and analysing content characteristics from the coded bit-stream;
- Hybrid model is a combination of some or all of these models. It is an effective model in terms of exploiting as much information as possible to predict the QoE.

Since subjective scores and objective quality indices typically have different ranges, a meaningful mapping function is required to map the objective video quality (VQ) into the predicted subjective score (MOS). Mapping functions can be categorised into linear and non-linear. The linear mapping function can be used when both objective and subjective scores are scaled uniformly, i.e. an equal numerical difference corresponds to an equal perceived quality difference over the whole range [Korhonen 2012].

## 3. QoE modelling

Quality of Experience prediction models can be **intrusive** and **non-intrusive**, where intrusive models predict QoE by extracting features from the output signal, either on its own or by comparing it with the input signal while non-intrusive models rely on network and application parameters.

There are a variety of methods for modeling the QoE which may be classified into two main groups **– statistical** methods and methods based on **artificial intelligence and machine learning**.

### 3.1. Statistical methods

The statistical methods include linear and nonlinear regression and correlation analysis. These methods involve mathematical operations such as computing average, variance and standard deviation of users' ratings.

Khan et al. [Khan 2009] proposed a model for QoE estimation based on content clustering and linear regression. The prediction focuses mainly on video attributes and the video content type is extracted with content clustering. Linear regression is used to design an equation which calculates MOS. According to the presented result, video content type has a significant effect on QoE [Tsaregorodtseva 2019].

Fiedler, Hossfeld and Tran-Gia [Fiedler 2010] presented a quantitative mapping between QoS and QoE using their IQX hypothesis. It is based on exponential relationship between QoS and QoE parameters. The IQX hypothesis takes as an input QoS parameters such as packet loss and jitter (in the form of p-ordered ratio) to determine QoE in the form of PESQ MOS for VoIP applications. The authors show that the derived non-linear regression equation can provide an excellent mapping between QoS parameters and MOS for VoIP application. The authors also tested their hypothesis for QoE related to web browsing by considering weighted session time and delivered bandwidth.

The main drawback of IQX hypothesis is that it only considers one QoS parameter to predict the corresponding QoE value. The authors did not consider the problem of integrating additional context and QoE parameters to predict the overall QoE. [Mirta 2018]

Chen et al. [CHEN 2009] presented OneClick to measure and predict QoE regarding multimedia applications such as VoIP, video streaming and gaming. The authors developed a Possion regression equation to predict users' QoE based on user click rates. The user click rate is computed when the users click the keys on their keyboard corresponding to network QoS conditions. The experimental analysis comprised of VoIP and video streaming applications but considered only three human subjects.

Kim et al. [Kim 2008] proposed a method for QoE prediction based on a function of QoS parameters such as delay, jitter, packet loss and bandwidth. Firstly, a normalized QoS value is computed based on the linear weighted sum of QoS parameters. Once the QoS value is computed, it is then used to determine QoE on the scale of one to five based on another QoE function. However, the authors do not discuss in detail how the weights of each QoS parameters can be computed. Further, their method is limited to QoS parameters and treats each parameter independently.

The main drawback of such methods is that in case of subjective tests, the normality of collected data (users ratings) cannot be verified.

In an effort to reduce the need for subjective studies, the authors in [Agboma 2008] present a method that only relies on limited subjective testing. The viewers marked the point at which the change of quality became noticeable by using the method of limits. Discriminate Analysis (DA) was used to predict group memberships from a set of quantitative variables. The group memberships were separated by mathematical equations and then derived. The derived equations are known as discriminant functions, which are used for prediction purposes. In this study, two video parameter shave been used namely the bitrate and the frame rate for three different terminals and six types of video content. The authors explain that involving other factors related to the video content and coding parameters can maximize the user perceived quality and achieve efficient network utilization. The accuracy of the developed model validated for each terminal was Mobile phones: 76.9%, PDAs: 86.6% and Laptops: 83.9%. However, this approach suffers from a limited accuracy due to the statistical method used to build the prediction models. Moreover, no specific implementation was considered for the QoS parameters at the Network Layer [Alreshoodi 2013].

### 3.2. Artificial Iintelligence and Machine Learning methods

The second group of methods include decision trees, fuzzy logic, artificial neural networks, hidden Markov models, Naïve Bayes, k-Nearest Neighbours (k-NN), Random forest, etc. [Mirta 2018]. These methods are more flexible and more adaptive in a dynamical environment, with many unknowns and missing data.

An experiment for QoE prediction of video streams by 4 methods is presented in [Tsaregorodtseva 2019]. The methods are Support Vector Machines, Random Forest, Gradient Boosted Trees, and Neural Networks.

Support Vector Machines (SVM) are maximum margin classifiers. In particular, linear SVMs seek to find a hyperplane in the dataspace that separates the data into its respective classes and maximizes the distance between the data points of different classes that are closest to this separating hyperplane. For example, in the case of two classes and two-dimensional data, it consists of finding a line which separates the data into two classes, and the two vectors of different classes closest to the line are furthest away from each other (See Figure 1).

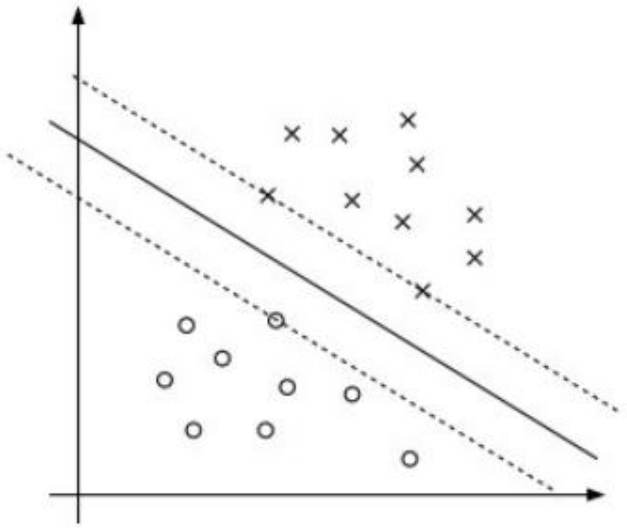

Figure 1. Linear SVM

However, most real word data sets cannot be linearly separated. The data can be projected into a higher dimensional space and the separating hyperplane can be learned in this space. A separating hyperplane can be efficiently learned with so called kernel trick – a function of two vectors k(x,y) and a measure of distance between two vectors must be specified [Scholkopf 1997].

Random forest is a supervised learning algorithm where the model is created by an ensemble of Decision Trees. A Decision Tree works by formulating a set of rules to use for prediction from the features and labels of the training data set. It can be described as a flowchart of ‘yes” or “no” questions that eventually lead to a predicted class or continuous value. Most commonly in cases of classification, the splits of nodes are chosen to maximize the reduction in Gini Impurity of their answers. Gini Impurity is a mathematical concept that represents the probability of a randomly chosen element of the set being incorrectly labeled if it was labeled by a distribution of samples in the set [Safavian 1991].

In a Decision Tree, at each node the algorithm searches through all of the possible features to find one which would result in the greatest Gini Impurity or MSE reduction, and then chooses it to split on. This splitting procedure is repeated reqursively until the tree reaches maximum depth. An issue with decision trees is that they are high variance methods and can fit noise in the dataset well, resulting in very different trees being learned for moderately different splits in the dataset. This results in severe overfitting to the training data and poor generalization performance. An approach to countering overfitting for high variance machine learning models is bagging, when an ensemble of models are trained on different random samples of the dataset. Random Forest is the application of bagging to decision trees. The algorithm selects a random subset of training data for each Decision Tree. and selects a random subset of features for splitting nodes. When a tree in a Random Forest picks a random sample of training data points they are drawn with replacement, which is known as bootstrapping, and the predictions of each tree in the random forest are averaged at test time. This procedure is known as bagging. An illustration of Random Forest with two estimators is shown in Figure 2 [Donges 2018].

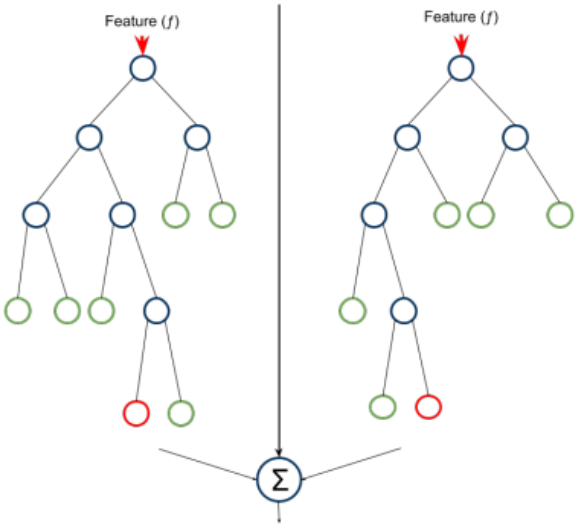


Figure 2. Random Forest with two estimators

Gradient boosting is a general technique similar to bagging that can be used to create an ensemble of models. While bagging is used to reduce overfitting of high variance models, Gradient Boosting is used to increase the power of high bias i.e. weak models that fail to fit the data well when used individually. Unlike in bagging, for Gradient Boosting the ensemble of models is trained sequentially rather than in parallel. In the case of Gradient Boosted Trees, which is the algorithm used in this work, Decision Trees are used as the weak model [Elith 2008] .

What sets Gradient Boosted Trees apart from the Random Forest algorithm is that trees are not random and independent of each other, but they are built sequentially, and each new tree attempts to minimize the loss function of all the trees combined. It is often the case that individual models in the ensemble become good at explaining data in a particular subspace of the data space and a good fit to the full dataspace can be achieved by combining of all these specialized models. Gradinet Boosted trees are quite efficient and do not use a lot of memory.

An artificial neural network (ANN), also called a simulated neural network (SNN) or commonly just neural network (NN) is an interconnected group of artificial

neurons that uses a mathematical or computational model for information processing based on a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network.

The classical architecture of artificial neural network consists of three groups, or layers, of units: a layer of "input" units is connected to a layer of "hidden" units, which is connected to a layer of "output" units. If the signals are travelling only in one direction – from input to output layers, the network is called feedforward neural network. The input layer does not perform any computation and just passes the information to the hidden layer. The inputs have associated weights that represent their importance comparing to other inputs. Hidden layers and output layers do perform computation, and the last hidden layer's nodes pass their outputs to the output layer, which produces the final result value. Figure 3 illustrates a feedforward neural network [Upadhyay 2019]. Feedback networks can have signals travelling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point.

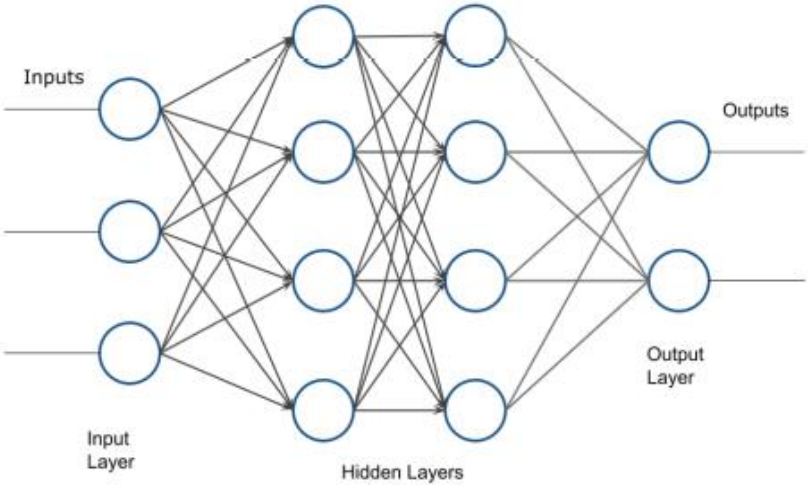


Figure 3. Feedforward Neural Network

In the standard back-propagation (BP) algorithm, the weights start off being random. Every input in the training data set is propagated through the NN, and the output is compared with the corresponding expected output. Then, based on the error, the weights are adjusted using the gradient descent optimization algorithm. This process repeats until the error is low enough, and after it terminates the NN has learned all of its weights and can be used for its intended purpose.

Artificial Neural Networks simulate the ability of the human brain for self-learning on the base of the collected information. They are used for building a self-organized architecture of the network and to define a self-learning algorithm for nonlinear systems modeling. Another important use of the ANN is in discovering theoretical links embedded in large chunks of data and the provision of richer interpretations of the interconnecting relationships existing between the variables. Since neural networks are best at identifying patterns or trends in data, they are well suited for prediction or forecasting needs.

The results in [Tsaregorodtseva 2019] show that the highest accuracy – 91.65% is achieved by Neural Networks (See Figure 4).

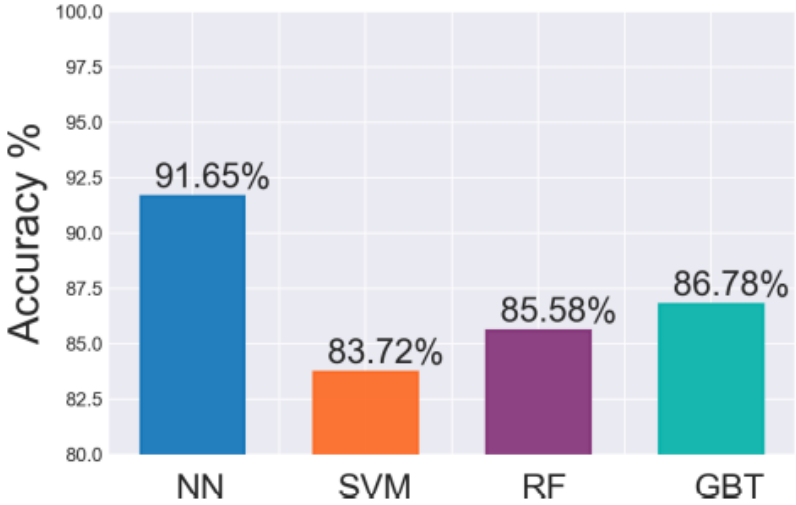


Figure 4. Accuracy of QoE prediction

Neural networks are widely used for time series prediction. They are also very suitable for modeling of the nonlinear relation between the QoE and QoS.

The study reported in [Machado 2011] proposed a method that connects the QoE metrics directly to QoS metrics according to the corresponding level of QoE. The QoE was estimated by employing a Multilayer ANN. The network QoS parameters were selected as the input layer, while the MOS, Peak Signal to Noise Ratio (PSNR) and Video Quality Metric (VQM) as the output layer. The ANN model was trained to get the correct weights. After training the ANN model, the relationship between the input layer and output layer was established. From the results, the proposed model gives acceptable prediction accuracy.

A model of QoE prediction for mobile 3D video streaming based on neural network is presented in [Almohammadi, 2019]. Neural networks are used for QoE prediction also in [Du, 2009], [Frank 2006] and many others.

Combined methods are also applied. E.g., a combined method for QoE prediction based on several Elman neural networks is presented in [Xu 2019]. Elman neural network is a type of locally recurrent network, which is considered as a special type of feedforward NN with additional memory neurons (context layer) and local feedback. As shown in Figure 5, the Elman NN consists of the context layer, input layer, hidden layer, and output layer. $W^1$ denotes the weight from the context layer to the hidden layer, $W^2$ denotes the weight from the input layer to the hidden layer, and $W^3$ denotes the weight from the hidden layer to the output layer. $U(t-1)$ denotes the network input vector at the $(t-1)$th iteration, $V(t)$ denotes the hidden layer output vector at the $t$-th iteration, and $Z(t)$ denotes the network output vector at the $t$-th iteration. The context layer retains the hidden layer output vector from the previous iteration; that is, $V^C(t)$ denotes the context layer output vector at the $t$-th iteration, and its value equals the hidden layer output vector at the $(t-1)$-th iteration. The transfer functions of the hidden layer and output layer units are $g(\cdot)$ and $h(\cdot)$, respectively.

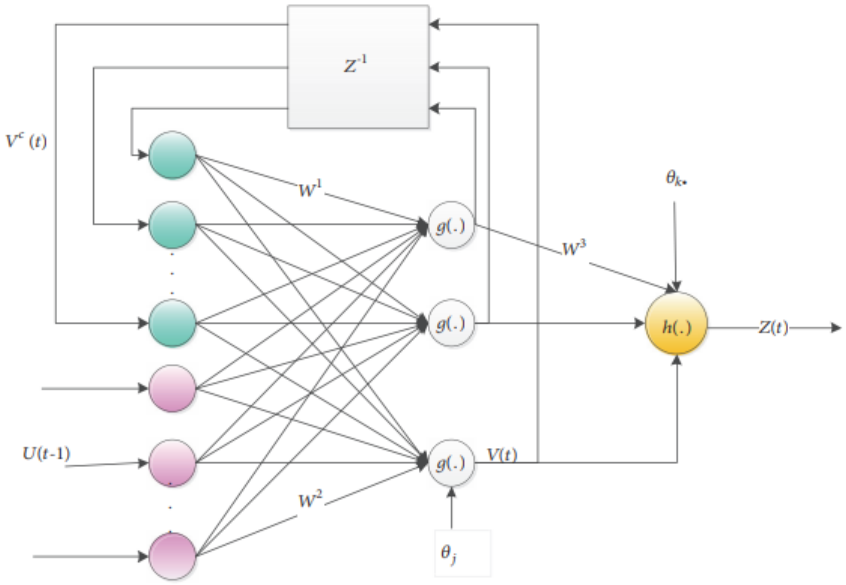


Figure 5. Elman neural network model

Because of its better learning efficiency, approximation ability, and memory ability than other neural network, the Elman NN can not only be used in time series prediction, but also in system identification and prediction. Combining the exits of the several Elman networks, the method has bigger generalizing ability and stability.

## Conclusion

Modeling the quality of experience of multimedia on-line services is a complex, multidisciplinary, multidimensional and challenging task. It involves concepts from several fields as computer networks, cognitive and behavioral science, human-computer interaction, economics, etc. It includes the definition of the set of context and additional parameters and the relations between them which are usually nonlinear dependences.

Even the best modern conceptual models cannot sufficiently solve the problems of measuring and prediction of the QoE for a long time period. It is recommended to include more parameters (context and additional) as well as to apply a more systematical and unified approach. It is necessary to use a bigger data base at the entrance of the machine learning models.

On the base of this survey we are going to build a conceptual model for QoE prediction and to apply it for multimedia on-line services.

## Acknowledgements

The work of Zlatinka Kovacheva is partially supported by the Task 1.2.5 of the Bulgarian National Scientific Program "ICT in Science, Education and Security", funded by the Ministry of Education and Science (MES) (Contract MES DOI-205/23.11.2018).

The work of S. Poryazov is partially supported by the joint research project "Symbolic-Numerical Decision Methods for Algebraic Systems of Equations in Perspective Telecommunication Tasks" of IMI-BAS, Bulgaria and JINR, Dubna, Russia.

## Bibliography

[Agboma 2008] Agboma F. , A. Liotta, (2008) "QoE-aware QoS management," in Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia, pp. 111-116

[Almohammadi, 2019] Almohammadi K., Quality Prediction Model Based on Artificial Neural Networks for Mobile 3D Video Streaming, International Journal of Computer Science and Information Security (IJCSIS), Vol. 17, No.11, November 2019

[Alreshoodi 2013] Alreshoodi M., J. Woods, Survey on QoE\QoS correlation models for multimedia services, International journal of distributed and parallel systems, Vol. 4, No. 3, May 2013

[Brooks 2010] Brooks, P., Hestnes B. (2010), "User measures of quality of experience: Why being objective and quantitative is important", IEEE network, 24(2), 8-13.

[Callet 2012] Le Callet P, MollerS and Perkis A. (eds), Qualinet White Paper on Definitions of Quality of Experience (2012). http://www.qualinet.eu/index.php?option=com_content&view=article&id=45&Itemid=52

[Chen 2009] Chen K., C. Tu, and W Xiao. Oneclick: A framework for measuring network quality of experience. In INFOCOM 2009, IEEE International Conference on Computer Communications., pages 702–710.

[Donges 2018] Donges N., “The Random Forest algorithm”, https://towardsdatascience.com/the-random-forest-algorithm-d457d4999ffcd, 2018

[Du 2009] Du Haiqing, Guo Ch., Liu Yixi, Liu Yong, Research on relationship between QoE and QoS on BP neural network, Proceedings of IC-NIDC 2009, pp. 312-315

[Elith 2008] Elith Jane, John R. Leathwick, and Trevor Hastie, “A working guide to boosted regression trees”, Journal of Animal Ecology, 77(4):802-813, 2008

[Fiedler 2010] Fiedler M., T. Hossfeld, and Phuoc Tran-Gia. A generic quantitative relationship between quality of experience and quality of service. Network, IEEE, 24(2):36 –41, march-april 2010.

[Frank 2006] Frank P., Incera J., A neural network base test bed for evaluating the quality of videostreemsin IP networks, 0-7695-2569-5/06, IEEE Proceedings of the Electronics, Robotics and Automotive Mechanics Conference (CERMA’06)

[Gong 2009] Gong Y., F. Yang, L. Huang, and S. Su. Model-based approach to measuring quality of experience. In Emerging Network Intelligence, 2009 First International Conference, pages 29–32.

[ITU-T Recommendation 2003] ITU-T Recommendation, (2003) “ITU-T Rec. P.800.1: Mean Opinion Score (MOS) Technology”

[Khan 2009] Khan A., Lingfen Sun, and Emmanuel Ifeachor, Content clustering based on video quality prediction model for MPEG4 video streaming over wireless networks, 2009 IEEE International Conference on Communications, pp. 1-5

[Kim 2008] Kim H. J., D. H. Lee, J. M. Lee, K. H. Lee, L. Won, and Seong G. C. The qoe evaluation method through the qos-qoe correlation model. In Networked Computing and Advanced Information Management, 2008.

(NCM'08). Fourth International Conference on, volume 2, pages 719 –725, Sept. 2008.

[Korhonen 2012] Korhonen J., Nino Burini, Junyong You, and Ehsan Nadernejad, (2012) "How to evaluate objective video quality metrics reliably", QoMEX IEEE.

[Machado 2011] Machado,V. C. Oliveira , A. Marcelino, S. Carlos, N. Vijaykumar, & C. Hirata, (2011) "A New Proposal to Provide Estimation of QoS and QoE over WiMAX Networks", 978-1-4673-0279- 1 © IEEE

[Mitra 2018] Mirta K. Ark. Zaslavsky, Chr. Ahlund, QoEModelling, Measurement and Prediction: A rewiew, 21 June 2018: ArXiv:1410.6952v1 [cs.NI] 25 Oct 2014

[Perkis 2006] Perkis A., S. Munkeby, and O.I. Hillestad. A model for measuring quality of experience. In Signal Processing Symposium, 2006. (NORSIG 2006). Proceedings of the 7th Nordic, pages 198–201

[Safavian 1991] Safavian S.R., D. Landgrebe, "A survey of decision tree classifier methodology", IEEE transactions on systems, man, and cybernetics", 21(3):660-674, 1991

[Scholkopf 1997] Scholkopf B, Kah-Kay Sung, Christopher JC Burges, Federico Girosi, Partha Niyogi, Tomaso Poggio, and Vladimir Vapnik, "Comparing Support Vector Machines with Gaussian kernels to radial basis function classifiers", IEEE transactions on Sognal Processing, 45(11): 2758-2765, 1997

[Seth 2005] Seth N., S.G. DeshmukhIndian, Prem VratIndian, Service quality models: a review, International Journal of Quality & Reliability Management 22(9), December 2005, 913-949 https://www.researchgate.net/publication/235286421_Service_quality_models_A_review

[Takahashi 2008] Takahashi A., D. Hands, &V. Barriac, (2008) "Standardization activities in the ITU for a QoE assessment of IPTV", Communications Magazine, IEEE, vol. 46, pp. 78-84.

[Tsaregorodtseva 2019] Tsaregorodtseva D., Using Machine Learning to Predict Quality of Experience of Video in LTE Networks, Dissertation for Master of Science in Trinity College Dublin, University of Dublin, April 2019

[Upadhyay 2019] Upadhyay Yash, "Introduction to FeedForward Neural Networks", https://towardsdatascience.com/feed-forward-neural-networks-c503faa46620 , 2018

[Xu 2019] Xu Lan, Zhang Y, Quality Prediction Model Based on Novel Elman Neural Network Ensemble, Hindawi Complexity Volume 2019, Article ID 9852134, 11 pages https://doi.org/10.1155/2019/9852134

## Authors' Information

***Zlatinka Kovacheva*** *– Institute of Mathematics and Informatics, Bulgarian Academy of Sciences; University of Mining and Geology, Sofia, Bulgaria; e-mail: zkovacheva@math.bas.bg*
*Major Fields of Scientific Research: Neural networks, Big data*

***Emiliya Saranova*** *– Institute of Mathematics and Informatics of the Bulgarian Academy of Sciences, Acad. G. Bonchev Str., Block 8, 1113 Sofia, Bulgaria]*
*University of Telecommunications and Post, Sofia, 1 Acad. St. Mladenov Str, Sofia 1700, Bulgaria, E-mail: e.saranova@utp.bg*
*Major Fields of Scientific Research: Information modeling, General theoretical information research, Multi-dimensional information systems*

***Stoyan Poryazov*** *– Institute of Mathematics and Informatics of the Bulgarian Academy of Sciences, Acad. G. Bonchev Str., Block 8, 1113 Sofia, Bulgaria, E-mail: stoyan@math.bas.bg*
*Major Fields of Scientific Research: Modeling and study of the traffic of telecommunication and computer systems, Development of methods and tools of information modeling and its application*

# Reliable Monte Carlo Methods for Multidimensional Sensitivity Analysis

**Venelin Todorov, Stoyan Poryazov**

***Abstract:*** *Sensitivity analysis is a promising technique for determining the stability, reliability, and efficiency of a mathematical model. Since the basic element in performing this procedure is the calculation of the corresponding numerical indicators, called total sensitivity indices, from a mathematical point of view this task is represented by multidimensional integrals. The total sensitivity index of an input parameter can be calculated with only one integral using the adaptive Monte Carlo method, analogous to the calculation of the first order indices. This makes the applied approach one of the most effective variance reduction based methods in terms of computational efficiency.*

***Keywords:*** *Multidimensional integration, Sensitivity Analysis, Monte Carlo methods, Air pollution modelling.*

***MSC****: 65C05, 65U05, 65F10, 65Y20*

***ITHEA Keywords****: NUMERICAL ANALYSIS: Applications, SIMULATION AND MODELLING: Applications.*

## Introduction

We discuss a systematic approach for sensitivity analysis studies in the area of air pollution modelling. The Unified Danish Eulerian Model (UNI-DEM) Zlatev [1995, 2006] is used in this particular study. Different parts of the large amount of output data, produced by the model, were used in various practical applications, where the reliability of this data should be properly estimated. Another reason to choose this model as a case study here is its sophisticated chemical scheme, where all relevant chemical processes in the atmosphere are accurately represented. We study the sensitivity of concentration variations of some of the most dangerous air pollutants with respect to the anthropogenic emissions levels and with respect to some chemical reactions rates. A special version of UNI-DEM (called SA-DEM) was developed for the purpose of this study. Description of UNI-DEM, SA-DEM and their parallel computer implementations will be given in the next section.

Different efficient stochastic algorithms for multidimensional integration have also been applied on a furter stage of these sensitivity studies. Between them are two adaptive Monte Carlo algorithms, described in more details in the paper. These will be compared with two QMC algorithms, namely Fibonacci lattice rule and Latin hypercube sampling. Fibonacci lattice rule is completely investigated in Wang and Hickernel [2002] and Latin hypercube sampling is described in detail in McKay et al. [1979].

## 1 Description and implementation of UNI-DEM

UNI-DEM is a powerful large-scale air pollution model for calculation the concentrations of a large number of pollutants and other chemical species in the air along certain time period. Its results can be used in various application areas (environmental protection, agriculture, health care, etc.). The large computational domain covers completely the European region and the Mediterranean.

UNI-DEM is mathematically represented by the following system of partial differential equations (PDE), in which the unknown concentrations $c_s$ of a large number of chemical species (pollutants and other chemically active components) take part. The main physical and chemical processes (advection, diffusion, chemical reactions, emissions and deposition) are represented in that system.

$$\begin{aligned}\frac{\partial c_s}{\partial t} &= -\frac{\partial(uc_s)}{\partial x} - \frac{\partial(vc_s)}{\partial y} - \frac{\partial(wc_s)}{\partial z} + \\ &+ \frac{\partial}{\partial x}\left(K_x\frac{\partial c_s}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y\frac{\partial c_s}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_z\frac{\partial c_s}{\partial z}\right) + \\ &+ E_s + Q_s(c_1, c_2, \ldots c_q) - (k_{1s} + k_{2s})c_s, \quad s = 1, 2, \ldots q\,.\end{aligned} \tag{1}$$

where $c_s$ are the concentrations of the chemical species; $u,\ v,\ w$ are the wind components along the coordinate axes; $K_x,\ K_y,\ K_z$ – the diffusion coefficients; $E_s$ – the emissions; $k_{1s},\ k_{2s}$ – dry / wet deposition coefficients; $Q_s(c_1, c_2, \ldots c_q)$ – non-linear functions describing the chemical reactions between species under consideration. The above PDE system is non-linear and stiff. Both non-linearity and stiffness are introduced mainly by the chemical scheme: the condensed CBM-IV (Carbon Bond Mechanism) Zlatev [1995, 2006]. It is quite detailed and accurate, but computationally expensive as well.

For the purpose of efficient numerical treatment, the system (1) is split according to the major physical and chemical processes and the following 3 submodels are formed: ***Advection-diffusion***, ***Chemistry & deposition*** and ***Vertical transport (vertical wind and convection)***.

The following methods are used in the numerical solution of the submodels:

- *Advection-diffusion part:* Finite elements, followed by predictor-corrector schemes with several different correctors.
- *Chemistry-deposition part:* An improved version of the QSSA (Quazi Steady-State Approximation) Zlatev [1995].
- *Vertical transport:* Finite elements, followed by theta-methods.

Spatial and time discretization makes each of the submodels a tuff computational task even for the most advanced supercomputer systems. Efficient parallelization has always been a crucial point in the computer implementation of UNI-DEM. The task became much more challenging with development of the sensitivity analysis version of the code, SA-DEM Zlatev [2006]. It consists of he following three parts:

– A modification of UNI-DEM with ability to modify certain parameters, subject to SA study. By now we have been interested in some chemical rate constants as well as in the input data for the anthropogenic emissions. A small number of input parameters is reserved for this purpose.

– A driver routine that automatically generates a set of tasks to produce the necessary results for a particular SA study. It allows to perform in parallel a large number of runs with common input data (reusing it), producing at once a whole set of values on a regular mesh (used later for calculating the sensitivity indices).

– An additional program for extracting the necessary mean monthly concentrations and computing the normalised ratios (to be analysed further on).

## Algorithm

Variance-based methods deliver results that are independent to the models behaviors: linearity, monotonicity and additivity of the relationship between input factor and model output sensitivity measures. The variance-based Sobol' method uses the sensitivity measures (indices) and takes into account interaction effects between inputs. An important advantage of this method is that it allows to compute not only the first-order indices, but also indices of a higher-order in a way similar to the computation of the main effects, the total sensitivity index can be calculated with just one MC integral per factor. The computational cost of estimating all first-order (m = 1) and total sensitivity indices via Sobol' approach is proportional to $dN$, where $N$ is the sample size and d is the number of input parameters.

The Sobol's method Dimov nad Georgieva [2010]is one of the most often used variance-based methods. It is based on a unique decomposition of the model function into orthogonal terms (summands) of increasing dimension and zero means. Its main advantage is computing in a uniform way not only the first order indices, but also the higher order indices (in quite a similar way as the computation of the main effects). The total sensitivity index can then be calculated with just one Monte Carlo integral per factor.

The Sobol method for global SA, applied here, is based on the so-called *High Dimensional Model Representation (HDMR)* (2) of the model function $f$ (integrable) in the $d$-dimensional factor spaceGeorgieva [2010]:

$$f(\mathbf{x}) = f_0 + \sum_{s=1}^{d} \sum_{l_1<\ldots<l_s} f_{l_1\ldots l_s}(x_{l_1}, x_{l_2}, \ldots, x_{l_s}) \,, \tag{2}$$

where $f_0$ is a constant. The representation (2) is not unique. Sobol has proven that under the conditions (3) for the right-hand-side functions

$$\int_0^1 f_{l_1\ldots l_s}(x_{l_1}, x_{l_2}, \ldots, x_{l_s}) \, \mathrm{x}_{l_k} = 0, \quad 1 \le k \le s, \quad s = 1, \ldots, d \tag{3}$$

the decomposition (2) is unique and is called *ANOVA-HDMR* of the model function $f(\mathbf{x})$ (the abbreviation ANOVA comming from *Analysis of Variances*). Moreover, the functions of the right-hand side can be defined in a unique way by multidimensional integrals as follows (see also **?**

).

$$f_0 = \int_{U^d} f(\mathbf{x}) \dot{\mathbf{x}} \, , \tag{4}$$

$$f_{l_1}(x_{l_1}) = \int_{U^{d-1}} f(\mathbf{x}) \prod_{k \neq l_1} \dot{\mathbf{x}}_k - f_0, \; l_1 \in 1, \ldots, d \, , \tag{5}$$

$$\int_{U^d} f_{i_1 \ldots i_\mu} f_{j_1 \ldots j_\nu} \dot{\mathbf{x}} = 0, \;\; (i_1, \ldots, i_\mu) \neq (j_1, \ldots, j_\nu), \;\; \mu, \nu \in \{1, \ldots, d\} \, . \tag{6}$$

**Definition:** Sobol global sensitivity indices

$$S_{l_1 \, \ldots \, l_\nu} = \frac{\mathbf{D}_{l_1 \, \ldots \, l_\nu}}{\mathbf{D}}, \quad \nu \in \{1, \ldots, d\} \tag{7}$$

are defined as ratios of the partial variances

$$\mathbf{D}_{l_1 \, \ldots \, l_\nu} = \int f^2_{l_1 \, \ldots \, l_\nu} \dot{\mathbf{x}}_{l_1} \ldots \dot{\mathbf{x}}_{l_\nu} \tag{8}$$

over the total variance

$$\mathbf{D} = \int_{U^d} f^2(\mathbf{x}) \dot{\mathbf{x}} - f_0^2, \qquad \mathbf{D} = \sum_{\nu=1}^{d} \sum_{l_1 < \ldots < l_\nu} \mathbf{D}_{l_1 \ldots l_\nu} \, . \tag{9}$$

From equalities (9) the following properties hold:

$$S_{l_1 \, \ldots \, l_s} \geq 0, \qquad \sum_{s=1}^{d} \sum_{l_1 < \ldots < l_s}^{d} S_{l_1 \, \ldots \, l_s} = 1 \, . \tag{10}$$

While the classical deterministic methods for numerical integration are effective for sub-integer functions having a relatively small dimension, for high dimensions they become even inapplicable because the number of sub-integral function values required to calculate is growing exponentially. On the other hand, the order of convergence of the adaptive Monte Carlo method for integration is independent of dimension. Therefore, the Monte Carlo approach is an effective apparatus for conducting sensitivity analysis of large-scale systems. Dispersion may increase with increasing dimension, but there are various Monte Carlo techniques to reduce variance - the sampling method and its modifications, the small discrepancy series, the importance partitioning method. The adaptability approach is also a widely used effective approach to improve the convergence order of deterministic and stochastic numerical integration methods. There are different approaches to designing adaptive Monte Carlo algorithms. The adaptive algorithm implemented here does not use any prior information on the smoothness of the subintegral function, but uses posterior variance information Dimov et al. [2003]. The basic idea is to concentrate random points in sub-areas in which the variance is large (in terms of preset accuracy), ie. the approach is based on the recursive division of the area, using a posteriori information about the current division error. The algorithm starts by dividing the intervals along all directions of the M sub-interval, with M being set as an input parameter. For each subdomain, the respective integral and variance are calculated. The resulting variance is

then compared to a predetermined value. The result of the comparison is used to further divide the area and increase the density of random points. Random numbers are used to determine the first and then the next direction to divide. In order to avoid unequal separation at different coordinates, the algorithm is designed so that a coordinate is re-split only after all other coordinates have been selected. The algorithm stops when the standard deviation in all subdivisions obtained after division has reached the predetermined accuracy. Thus, an approximation of the integral with the MC approach is obtained. This algorithm has been used to calculate the relevant sensitivity indices in the study of the effect of chemical reaction rate constants on the concentrations of some pollutants (eg ozone). The computational complexity is proportional to the sample size n and the number of input parameters.

Adaptive strategy Georgieva [2010] is well known method for evaluation of multidimensional integrals, especially when the integrand function has peculiarities and peaks. Adaptive Monte Carlo methods proposed by Lautrup use a "sequence of refinements" of the original area selected so that the computations to be concentrated in subdomains with computational difficulties. There are various adaptive strategies depending on the technique of adaptation. Our adaptive algorithm (simple adaptive Monte Carlo algorithm) does not use any a priori information about the smoothness of the integrand, but it uses a posteriori information for the variance obtained during calculations. The main idea is a concentration of random points in the subregions where the variance is large (in terms of a preliminary given accuracy), i.e. the approach is based on a recursive partitioning of the integration area using a posteriori error information for the current partition. Let $p_j$ and $I_{\Omega_j}$ are the following expressions: $p_j = \int_{\Omega_j} p(\mathbf{x})\,\mathrm{d}\mathbf{x}$ and $I_{\Omega_j} = \int_{\Omega_j} f(\mathbf{x})p(\mathbf{x})\,\mathrm{d}\mathbf{x}$. Consider now a random point $\xi^{(j)} \in \Omega_j$ with a density function $p(\mathbf{x})/p_j$. In this case $I_{\Omega_j} = \mathbf{E}\left[\frac{p_j}{N}\sum_{i=1}^{N} f(\xi_i^{(j)})\right] = \mathbf{E}\theta_N$. This adaptive algorithm gives an approximation with an error $\varepsilon \leq c\, N^{-1/2}$, where $c \leq 0.6745\sigma(\theta)$ ($\sigma(\theta)$ is the standard deviation). From the estimation of the error, it can be concluded that, in general, the simple adaptive Monte Carlo algorithm gives an error less than the error of the Plain Monte Carlo algorithm, but the order is the same. The adaptive MC algorithm applied here is described below.

**Algorithm**

1. **Input data**: *total number of points $N1$, constant $M = 4$(the initial number of subregions taken), constant $\varepsilon$ (max value of the variance in each subregion), constant $\delta$ (maximal admissible number of subregions), d-dimensionality of the initial region/domain, f - the function of interest.*

   1.1. **Calculate** *the number of points to be taken in each subregion $N = N1/\delta$.*

2. **For** $j = 1,\ M^d$:

   2.1. **Calculate** *the approximation of $I_{\Omega_j}$ and the variance $\mathbf{D}_{\Omega_j}$ in subdomain $\Omega_j$ based on $N$ independent realizations of random variable $\theta_N$;*

   2.2. **If** $(\mathbf{D}_{\Omega_j} \geq \varepsilon)$ **then**

      2.2.1. **Choose** *the axis direction on which the partition will perform,*

      2.2.2. **Divide** *the current domain into two $(G_{j_1}, G_{j_2})$ along the chosen direction,*

      2.2.3. **If** *the length of obtained subinterval is less than $\delta$* **then go to** *step 2.2.1* **else** *$j = j_1$ $G_{j_1}$ is the current domain right and* **go to** *step 2.1;*

**2.3. Else if** $(\mathbf{D}_{\Omega_j} < \varepsilon)$ *but an approximation of* $I_{G_{j_2}}$ *has not been calculated yet*, **then** $j = j_2$ $G_{j_2}$ *is the current domain along the corresponding direction right and* **go to** *step 2.1;*

**2.4. Else if** $(\mathbf{D}_{\Omega_j} < \varepsilon)$ *but there are subdomains along the other axis directions,* **then go to** *step 2.1;*

**2.5. Else** *Accumulation in the approximation* $I_N$ *of* $I$*.*

**Computational complexity**

Let $N$ be the dimensionality of the problem under consideration. First let's describe briefly the Crude Monte Carlo algorithm. Let $\xi$ be a random point with probability density function $p(x)$. Introducing the random variable

$$\theta = f(\xi)$$

with mathematical expectation equal to the value of the integral $I_{G_j}$, then

$$E\theta = \int_{G_j} f(x)p(x)dx.$$

Let $\xi_1, \xi_2, \ldots, \xi_N$ be independent realizations of the random point $\xi$ with probability density function $p(x)$ and $\theta_1 = f(\xi_1), \ldots, \theta_N = f(\xi_N)$ . Then an approximate value of $I_{G_j}$ is

$$\hat{\theta}_N = \frac{1}{N}\sum_{i=1}^{N}\theta_i.$$

One can easily see that the computational complexity of the Crude Monte Carlo is linear, because in this simple case we have to choose $N$ random points in the domain and every such choice is at the cost of $\mathcal{O}(1)$ operations. One single evaluation of the function in any of these points is also at the cost of $\mathcal{O}(1)$ operations.

In the adaptive Monte Carlo algorithm we are doing the same number of operations as in the Crude Monte Carlo algorithm. For the simple case when we have the two dimensional case ($N = 2$) and on the first step in the optimized adaptive approach we have $M = 4$ subdomains in our optimized Adaptive approach and

$$\hat{\theta}_N = \frac{1}{N_1}\sum_{i=1}^{N_1}\theta_i + \frac{1}{N_2}\sum_{i=1}^{N_2}\theta_i + \frac{1}{N_3}\sum_{i=1}^{N_3}\theta_i + \frac{1}{N_4}\sum_{i=1}^{N_4}\theta_i,$$

where $N_1 + N_2 + N_3 + N_4 = N$, so we have the same number of operations as the Crude Monte Carlo, which computational complexity is linear, to evaluate an approximation of $I_{G_j}$.

In general case for the adaptive algorithm the computational complexity depends on the integrand. First, let's consider the worst case. We always have a domain(area) with variance greater than the parameter $\varepsilon$ and we need to divide this domain to $2^N$ subdomains. Additionally, for each of these $2^N$ newly obtained domains, we have to choose $N$ random points into it and every such choice is at the cost of $\mathcal{O}(N)$ operations. We will receive complexity $\mathcal{O}(N.2^N)$.

So we choose only $\mathcal{O}(1)$ subdomains where the variance is greater than the parameter $\varepsilon$ and this is independent of $N$. When we divide the domain on every step adaptiveness is not in all subdomains, but only in $\mathcal{O}(1)$ subdomains. At the beginning we have to choose $\frac{N}{k_0}$ random points. After that when dividing the domain into $2^N$ subdomains, we choose only $\mathcal{O}(1)$ subdomains, this choice is again independent of $N$. In these subdomains we choose $\frac{N}{k_1}$ points. On the $j^{th}$ step of the Adaptive approach we choose $\mathcal{O}(1)$ subdomains with $\frac{N}{k_j}$ points. We have that $\sum\limits_{j=0}^{i} \frac{1}{k_j} = 1$. Therefore for the computational complexity we obtain

$$\frac{N}{k_0} + \mathcal{O}(1)\frac{N}{k_1} + \cdots + \mathcal{O}(1)\frac{N}{k_i} =$$

$$= N\mathcal{O}(1)\left(\sum_{j=0}^{i}\frac{1}{k_j}\right) = N\mathcal{O}(1) = \mathcal{O}(N).$$

In this way we can conclude that the computational complexity of the optimized Adaptive algorithm is linear.

Two adaptive approaches ADAPT1 (M=1) and ADAPT2(M=2) will be compared with Fibonacci besed lattice rule (FIBO) and Latin hypercube sampling (LHS).

## 2 Sensitivity Studies with Respect to Emission Levels

In the huge output data stream of UNI-DEM are the mean monthly concentrations of more than 30 pollutants. We consider 2 of them: *ozone* ($O_3$) and *ammonia* ($NH_3$). In particular, we present some results of a sensitivity studiy of the mean monthly concentrations of ammonia.

Here we present some results of our research on the sensitivity of UNI-DEM output (in particular, the ammonia mean monthly concentrations) with respect to the anthropogenic emissions input variation. The anthropogenic emissions input consists of $4$ different components $\mathbf{E} = (\mathbf{E^A}, \mathbf{E^N}, \mathbf{E^S}, \mathbf{E^C})$ as follows:

$\mathbf{E^A}$ – ammonia $(NH_3)$;
$\mathbf{E^N}$ – nitrogen oxides $(NO\ +\ NO_2)$;
$\mathbf{E^S}$ – sulphur dioxide $(SO_2)$;
$\mathbf{E^C}$ – anthropogenic hydrocarbons.

The domain is the $4$-dimensional hypercube $[0.5, 1]^4$). Polynomials of 2-nd degree have been used as an approximation tool Georgieva [2010]. The input data have been generated by the improved version SA-DEM code, specialized for sensitivity studies (see the previous section).

The results for relative errors for evaluation of the quantities $f_0$, total variances and first-order and total sensitivity indices using various stochastic approaches for numerical integration are presented in Tables 1, 2, 3, respectively. The quantity $f_0$ is presented by $4$-dimensional integral whereas the rest of quantities under consideration are presented by 8-dimensional integrals following the ideas of *correlated sampling* technique to compute sensitivity measures in a reliable way Georgieva [2010].

The results in Table 1 show that the algorithms using generalized Fibonacci numbers and LHS simulate the behaviour of the Adaptive Monte Carlo algorithm, but for higher dimensions their

Table 1: Relative error for evaluation of $f_0 \approx 0.048$.

| # of samples $n$ | ADAPT1 Relative error | ADAPT2 Relative error | FIBO Relative error | LHS Relative error |
|---|---|---|---|---|
| $2^{10}$ | 1.88e-03 | 1.03e-03 | 2.09e-04 | 5.37e-04 |
| $2^{12}$ | 2.05e-04 | 5.05e-04 | 4.32e-05 | 2.27e-04 |
| $2^{14}$ | 1.83e-04 | 1.38e-05 | 2.25e-05 | 6.28e-05 |
| $2^{16}$ | 9.89e-05 | 4.05e-04 | 8.70e-06 | 7.74e-05 |
| $2^{18}$ | 3.95e-05 | 3.83e-06 | 1.79e-06 | 3.80e-06 |
| $2^{20}$ | 4.99e-05 | 2.93e-05 | 4.21e-07 | 7.16e-06 |

Table 2: Relative error for evaluation of the total variance $\mathbf{D} \approx 0.0002$.

| # of samples $n$ | ADAPT1 Relative error | ADAPT2 Relative error | FIBO Relative error | LHS Relative error |
|---|---|---|---|---|
| $2^{10}$ | 1.56e-03 | 4.76e-03 | 1.63e-01 | 1.74e-02 |
| $2^{12}$ | 2.58e-03 | 4.28e-04 | 2.39e-02 | 1.04e-02 |
| $2^{14}$ | 6.03e-04 | 2.79e-04 | 2.90e-03 | 1.04e-02 |
| $2^{16}$ | 1.83e-04 | 5.12e-04 | 2.65e-04 | 3.65e-04 |
| $2^{18}$ | 5.77e-05 | 1.21e-04 | 3.01e-04 | 1.21e-05 |
| $2^{20}$ | 3.42e-05 | 3.28e-05 | 1.19e-04 | 5.96e-05 |

efficiency decreases. The particular case study confirms the conclusion that these algorithms are suitable and more efficient for smooth functions with relatively low dimensions. From Tables 1 and 2 we can conclude that all stochastic approaches under consideration give reliable relative errors for sufficiently large number of samples. This is not the case for some sensitivity indices, which are very small by absolute value (see Table 3), but fortunately, these are of low importance too. The most efficient in terms of computational complexity is the FIBO algorithm, followed very closely by the LHS algorithm. Adaptive algorithm gives results of the same order as LHS and FIBO, and sometimes even outperforms them – see for example the relative errors for $S_1^{tot}$ in Table 3.

Most influential emissions about ammonia output concentrations are ammonia emissions themselves (about $89\%$ for Milan). The second most influential emissions about ammonia output are sulphur dioxide emissions (about $11\%$) - see Fig. 1. Pie charts representation of first- and second-order sensitivity indices of the ammonia in Milan on Fig. 1 has been obtained applying the correlated sampling of SobolâĂŹ variance-based approach for multidimensional sensitivity analysis for computing all possible sensitivity measures to study influence of four chosen groups of air pollutant emissions over the concentration of the three important air pollutants.

Table 3: Relative error for estimation of sensitivity indices of the input anthropogenic emissions by using various Monte Carlo and quasi-Monte Carlo approaches ($n = 2^{16} = 65536$).

| Est. quantity | Ref. value | **ADAPT1** | **ADAPT2** | **FIBO** | **LHS** |
|---|---|---|---|---|---|
| $S_1$ | 9**e**-01 | 7.67e-04 | 1.22e-03 | 3.62e-04 | 9.79e-03 |
| $S_2$ | 2**e**-04 | 1.47e-03 | 4.96e-02 | 1.74e-01 | 6.60e-01 |
| $S_3$ | 1**e**-01 | 4.11e-03 | 1.59e-03 | 3.22e-03 | 8.65e-03 |
| $S_4$ | 4**e**-05 | 1.04e-01 | 1.69e-01 | 4.87e-01 | 6.70e-01 |
| $S_1^{tot}$ | 9**e**-01 | 4.99e-05 | 5.36e-05 | 4.61e-04 | 4.31e-04 |
| $S_2^{tot}$ | 2**e**-04 | 5.23e-01 | 5.00e+00 | 3.45e-01 | 2.94e+01 |
| $S_3^{tot}$ | 1**e**-01 | 1.15e-02 | 1.28e-02 | 1.96e-03 | 1.10e-02 |
| $S_4^{tot}$ | 5**e**-05 | 1.88e+01 | 3.43e+01 | 5.06e+01 | 2.41e+02 |

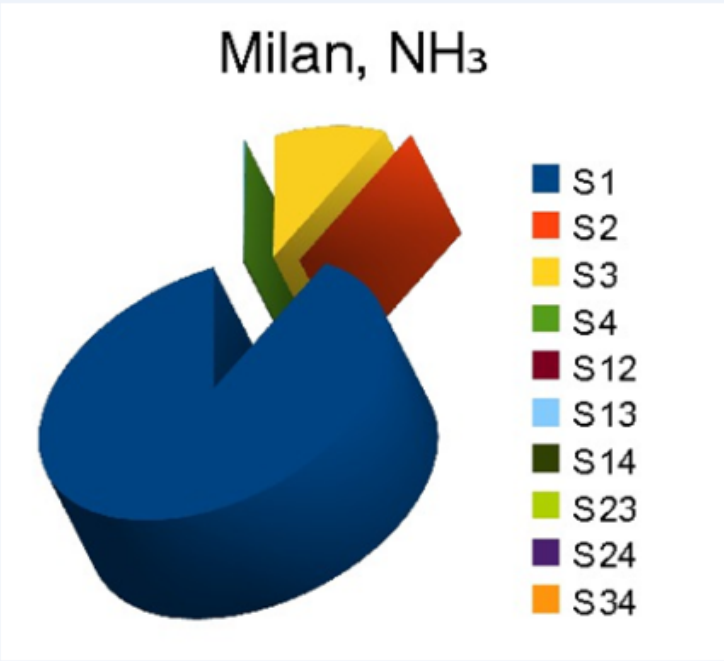


Figure 1: Pie charts representation of first- and second-order sensitivity indices of the ammonia in Milan

## Sensitivity Studies with Respect to Chemical Reactions Rates

We will also study the sensitivity of the ozone concentration values in the air over Genova with respect to the rate variation of some chemical reactions of the condensed CBM-IV scheme (Zlatev [1995]), namely: ## $1, 3, 7, 22$ (time-dependent) and $27, 28$ (time independent). The simplified chemical equations of those reactions are as follows:

$[\#\mathbf{1}]\ NO_2 + h\nu \Longrightarrow NO + O;$
$[\#\mathbf{3}]\ O_3 + NO \Longrightarrow NO_2;$
$[\#\mathbf{7}]\ NO_2 + O_3 \Longrightarrow NO_3;$
$[\#\mathbf{22}]\ HO_2 + NO \Longrightarrow OH + NO_2;$
$[\#\mathbf{27}]\ HO_2 + HO_2 \Longrightarrow H_2O_2;$
$[\#\mathbf{28}]\ OH + CO \Longrightarrow HO_2.$

The domain under consideration is the 6-dimensional hypercube $[0.6, 1.4]^6$). Polynomials of second degree have been used for approximation again (see Georgieva [2010]).

Homma and Saltelli discuss in Homma and Saltelli [1996] which is the better estimation of $f_0^2 = \left(\int_{U^d} f(\mathbf{x})\mathrm{d}\mathbf{x}\right)^2$ in the expression for total variance and Sobol global sensitivity measures. In case of estimating sensitivity indices of a fixed order, the formula

$$f_0^2 \approx \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_{i,1}, \ldots, \mathbf{x}_{i,d})\ f(\mathbf{x}'_{i,1}, \ldots, \mathbf{x}'_{i,d}),$$

where $\mathbf{x}$ and $\mathbf{x}'$ are two independent sample vectors, is better (as recommended in Georgieva [2010]).

Table 4: Relative error for evaluation of $f_0 \approx 0.27$.

| # of samples $n$ | **ADAPT1** Relative error | **ADAPT2** Relative error | **FIBO** Relative error | **LHS** Relative error |
|---|---|---|---|---|
| $2^{10}$ | 2.74e-04 | 3.21e-04 | 2.08e-03 | 3.73e-04 |
| $2^{12}$ | 9.55e-05 | 4.43e-05 | 1.40e-04 | 2.41e-04 |
| $2^{14}$ | 1.20e-04 | 5.64e-05 | 3.98e-04 | 7.53e-05 |
| $2^{16}$ | 3.49e-05 | 3.72e-05 | 2.61e-04 | 2.02e-04 |

The relative errors for evaluation of the quantities $f_0$, total variances, first and second order sensitivity indices by using various stochastic approaches for numerical integration are presented in Tables 4, 5, 6 respectively. Here the quantity $f_0$ is presented by a $6$-dimensional integral, whereas the total variance and the sensitivity indices are presented by 12-dimensional integrals, following the ideas of *correlated sampling*.

Table 5: Relative error for evaluation of the total variance $\mathbf{D} \approx 0.0025$.

| # of samples $n$ | **ADAPT1** Relative error | **ADAPT2** Relative error | **FIBO** Relative error | **LHS** Relative error |
|---|---|---|---|---|
| $2^{10}$ | 9.67e-04 | 1.18e-03 | 6.73e+00 | 1.91e-02 |
| $2^{12}$ | 9.10e-04 | 2.24e-03 | 5.27e-01 | 9.99e-02 |
| $2^{14}$ | 1.40e-04 | 1.86e-04 | 1.02e-01 | 1.62e-02 |
| $2^{16}$ | 3.01e-05 | 1.48e-04 | 1.97e-03 | 3.56e-05 |

Table 6: Relative error for estimation of sensitivity indices of several chemical reaction rate parameters by using various Monte Carlo and quasi-Monte Carlo approaches ($n = 2^{16} = 65536$).

| Est. quantity | Ref. value | **ADAPT1** | **ADAPT2** | **FIBO** | **LHS** |
|---|---|---|---|---|---|
| $S_1$ | 4**e**-01 | 1.55e-04 | 3.48e-04 | 3.82e-02 | 3.04e-02 |
| $S_2$ | 3**e**-01 | 4.34e-04 | 1.58e-04 | 1.03e-02 | 7.35e-04 |
| $S_3$ | 5**e**-02 | 3.42e-04 | 8.09e-05 | 5.48e-01 | 2.33e-02 |
| $S_4$ | 3**e**-01 | 4.75e-04 | 9.04e-04 | 1.07e-02 | 2.47e-02 |
| $S_5$ | 4**e**-07 | 1.31e+01 | 1.07e+01 | 3.40e+03 | 9.25e+02 |
| $S_6$ | 2**e**-02 | 1.08e-03 | 4.54e-04 | 1.32e+00 | 3.81e-02 |
| $S_{12}$ | 6**e**-03 | 1.30e-02 | 7.92e-03 | 3.21e+00 | 8.99e-02 |
| $S_{14}$ | 5**e**-03 | 5.30e-03 | 1.81e-03 | 8.64e+00 | 2.74e-01 |
| $S_{15}$ | 8**e**-06 | 9.34e+02 | 9.34e+02 | 9.19e+02 | 9.21e+02 |
| $S_{24}$ | 3**e**-03 | 1.26e-03 | 7.24e-03 | 1.37e+01 | 7.10e-01 |
| $S_{45}$ | 1**e**-05 | 9.93e-02 | 8.55e-02 | 4.25e+01 | 1.05e+01 |

Quasi-MC lattice rule based on generalized Fibonacci numbers and Latin hypercube sampling produce better results for 6-dimensional integrals in comparison with $12$-dimensional integrals. It is clear that with the increasing the dimensionality of the integral Adaptive method produce more accurate results than both FIBO and LHS. Adaptive Monte Carlo algorithm gives better results in case of higher dimensional integrals and lower number of samples. For most of the sensitivity indices Adaptive Monte Carlo algorithm gives more accurate results than FIBO and LHS by at least 2 orders of degree.

The representation of the reference values of the first-order and second-order sensitivity indices of input parameters (for ozone concentrations) are given on Fig.2. One can observe that the second-order sensitivity indices take rather small portions. This fact shows that the UNI-DEM is additive according to the rates of the six chemical reactions studied in this work.

## Conclusion

This study focuses on environmental safety. Sensitivity analysis, and in particular the results obtained, play an extremely important two-sided role: to test and improve mathematical models, and, on the other hand, to reliably interpret the numerical results by relevant specialists. Dispersion-based analysis is a very effective tool for in-depth study of the relationship between individual parameters, outputs and internal mechanisms governing the system in question. By identifying the major chemical reactions that affect the behavior of the system, specialists in various fields of application (physics, chemistry) will be able to obtain valuable information about improving the model, which in turn will increase the reliability and sustainability of forecasts. Thus, through a sensitivity analysis, the mathematical model will help to make more accurate estimates of the effects of harmful emissions on human health and agricultural losses. The results obtained show that the stochastic adaptive approach developed is one of the most efficient methods based on reducing the variance in terms of computational efficiency and accuracy.

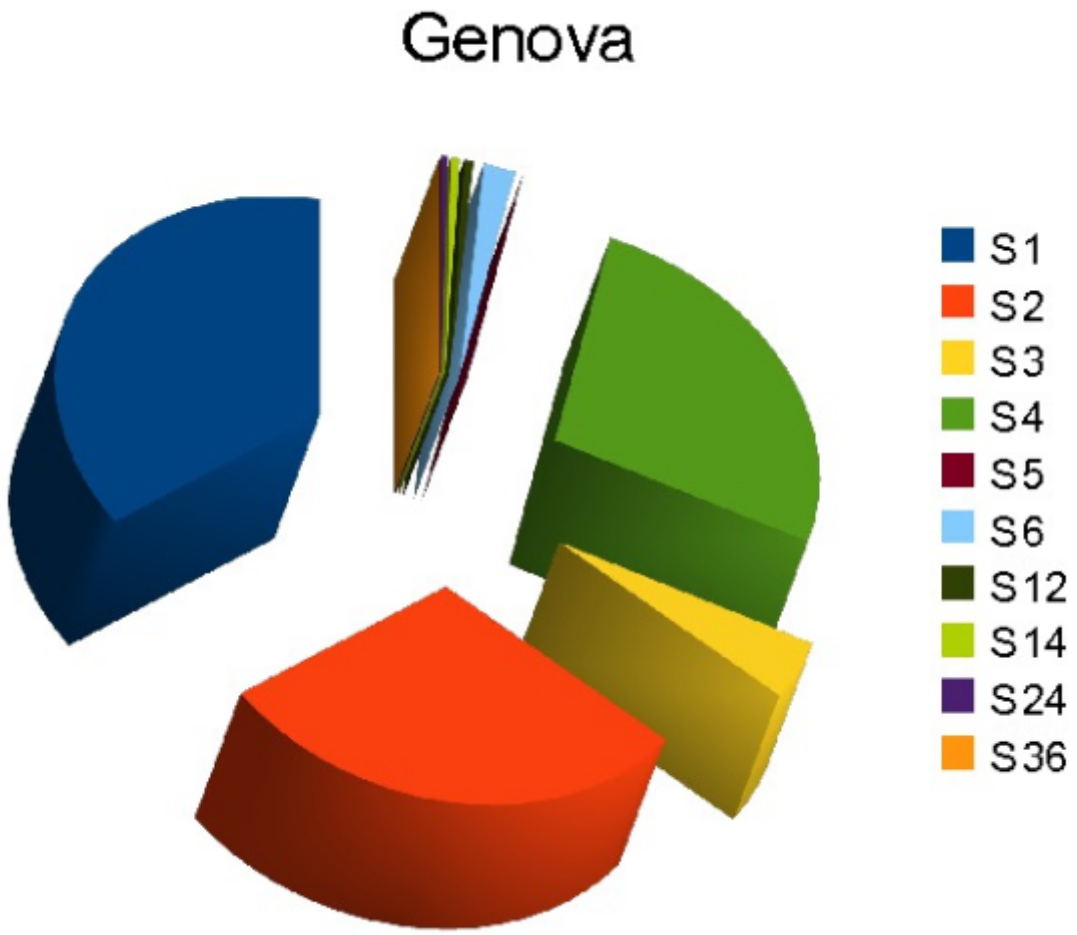


Figure 2: Pie charts representation of first- and second-order sensitivity indices of the ozone in Genova

## Acknowledgements

Venelin Todorov is supported by the National Scientific Program "Young scientists and Postdoctoral candidates" 2020-2021 and by the National Scientific Program "Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICT in SES)", contract No DO1-205/23.11.2018, financed by the Ministry of Education and Science in Bulgaria.

## Bibliography

I. Dimov, R. Georgieva (2010). Monte Carlo algorithms for evaluating Sobol sensitivity indices. Mathe- matics and Computers in Simulation, 81 (3), 506-514.

Dimov I., Karaivanova A., Georgieva R., Ivanovska S. Parallel Importance Separation and Adaptive Monte Carlo Algorithms for Multiple Integrals, Springer Lecture Notes in Computer Science, 2542, Springer-Verlag, Berlin, Heidelberg, 2003, New York: 99–107.

R. Georgieva, PhD thesis, Sofia, 2010.

McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21(2), 239âĂŞ45 (1979)

T. Homma, A. Saltelli, Importance Measures in Global Sensitivity Analysis of Nonlinear Models, Reliability Engineering and System Safety 52, 1996, 1–17.

S. Poryazov. A suitable unit of sensitivity in telecommunications. TELECOM 2011, 13-14.10.2011, Sofia, ISSN 1314-2690, p. 165–172.

Z. Zlatev, Computer treatment of large air pollution models, 1995.

Y. Wang and F. J. Hickernell, *An historical overview of lattice point sets*, 2002

Z. Zlatev, I. T. Dimov, Computational and Numerical Challenges in Environmental Modelling, Elsevier, Amsterdam, 2006.

## Authors' Information

***Venelin Todorov*** *- Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Information Modeling Department, Acad. Georgi Bonchev Str., Block 8, Sofia 1113, Bulgaria;*
*Institute of Information and Communication Technologies, Bulgarian Academy of Sciences,*
*Department of Parallel Algorithms, Acad. Georgi Bonchev Str., Block 25A, Sofia 1113, Bulgaria;*
*e-mail: vtodorov@math.bas.bg,venelin@parallel.bas.bg*
*Major Fields of Scientific Research: Monte Carlo methods, Sensitivity Analysis Multi-dimensional integrals, Digital nets, Lattice sequences*

# УСТРОЙСТВА УМНОЖЕНИЯ ОДИНАРНОЙ ТОЧНОСТИ НА БАЗЕ FPGA

**Владимир Опанасенко, Сергей Крывый, Станислав Завьялов**

***Аннотация**: Определено число частичных произведений для обеспечения результата одинарной точности. В работе предложены последовательная и последовательно-параллельная структуры устройств умножения матриц. Приведены временные и аппаратурные оценки их реализации на базе FPGA. Синтезированы функциональные блоки с плавающей точкой, совместимые со стандартом IEEE–754, которые могут быть использованы в качестве библиотечного элемента при разработке сложных вычислительных устройств.*

***Ключевые слова**: одинарная точность, последовательная структура, последовательно-параллельная структура, умножение, FPGA.*

***ITHEA Keywords**: B. Hardware: B.2 ARITHMETIC AND LOGIC STRUCTURES: B.2.4 High-Speed Arithmetic*

## Введение

Принцип работы произвольного устройства умножения [Майоров, 1970] можно представить совокупностью двух операций – формирования и суммирования частичных произведений, которые могут использоваться в различной последовательности. Изменение этой последовательности влияет на значение основных характеристик устройства – быстродействие и стоимость [Opanasenko, 2017]. Однако, в любом случае эти характеристики зависят от числа частичных произведение, которые необходимо сформировать для обеспечения заданной точности.

Для разрядности $n$ сомножителей произведение в общем случае представляется $2n$ –разрядным. Удвоенная разрядность результата существенна для случая организации умножения многократной точности. В других случаях произведение ограничивают $n$ разрядами, используя симметричное округление, так как не имеет смысла определять результат с погрешностью, намного меньшей наследственной погрешности, определяемой неточностью сомножителей (при максимуме модуля погрешности сомножителей $\varepsilon = 2^{-n-1}$ модуль погрешности произведения находится в диапазоне $0 \div 2^{-n}$).

## 1. Постановка задачи определения числа частичных произведений с одинарной точностью результата.

Пусть сомножители $A$ и $B$ представлены в виде

$$A = \sum_{i=1}^{l} a_i 2^{-ih}, \; B = \sum_{j=1}^{l} b_j 2^{-jh}, \tag{1}$$

где $h$ - число двоичных разрядов сомножителей модуля-умножителя (МУ); $a_i \le p-1, b_j \le p-1$ - целые числа двоичного представления $p$ –ричной цифры, $p = 2^h$; $l$ –разрядность $p$ –ричных сомножителей ( $n = lh$ ). Тогда произведение определяется выражением

$$C = AB = \sum_{i=1}^{l} \sum_{j=1}^{l} a_i b_j 2^{-(i+j)h}.$$

Заметим, что частичные произведения $c_{ij} = a_i b_j$ являются $2h$ –разрядными числами, причем каждое из них представляется суммой

$$c_{ij} = d_{ij} 2^h + g_{ij}, \tag{2}$$

где $d_{ij} \le p-1$ и $g_{ij} \le p-1$ – двоичное представление старшей и младшей $p$ –ричной цифры частичного произведения.

Частичные произведения $c_{ij}, i, j = \overline{1,l}$ упорядочим в виде ленточной матрицы шириной $l$ таким образом, чтобы столбцы матрицы содержали компоненты, имеющие одинаковые весовые множители:

$$\left\| \begin{array}{cccccccccc} c_{1,1} & c_{1,2} & c_{1,3} & \ldots & c_{1,l} & 0 & 0 & \ldots & 0 & 0 \\ 0 & c_{2,1} & c_{2,2} & \ldots & c_{2,l-1} & c_{2,l} & 0 & \ldots & 0 & 0 \\ 0 & 0 & c_{3,1} & \ldots & c_{3,l-2} & c_{3,l-1} & c_{3,l} & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & c_{l-1,2} & c_{l-1,3} & c_{l-1,4} & \ldots & c_{l-1,l} & 0 \\ 0 & 0 & 0 & \ldots & c_{l,1} & c_{l,2} & c_{l,3} & \ldots & c_{l,l-1} & c_{l,l} \end{array} \right\| \quad (3)$$

Каждая строка содержит элементы $c_{ij}$ с постоянным номером $i$, нумерация строк (сверху–вниз) соответствует возрастанию значения $i$, число столбцов – $(2l-1)$, сумма индексов $i+j$ для каждого столбца постоянна и возрастает при перечислении столбцов (слева-направо) от 2 до $2l$. В результате

$$C = \sum_{k=2}^{l+1} 2^{-kh} \left( \sum_{i=1}^{k-1} c_{i,\, j=k-i} + 2^{-lh} \sum_{i=k}^{l} c_{i,\, j=l+k-i} \right), \quad (4)$$

где при $i = l+1$

$$2^{-lh} \sum_{i=l+1}^{l} c_{i,\, j=l+k-i} = 0,$$

так как нижний предел суммирования больше верхнего. Чтобы разделить результат на старшие и младшие разряды (по $l$ $p$ –ричных разрядов), преобразуем (4) к виду

$$C = \sum_{k=2}^{l} 2^{-kh} \sum_{i=1}^{k-1} c_{i,\, j=k-i} + 2^{-(l+1)h} \sum_{i=1}^{l} c_{i,\, j=l+1-i} + 2^{-(l+2)h} \sum_{i=2}^{l} c_{i,\, j=l+2-i} + \sum_{k=3}^{l} 2^{-(l+k)h} \sum_{i=k}^{l} c_{i,\, j=l+k-i}, \quad (5)$$

Тогда с учетом (2) получим

$$C=\sum_{k=2}^{l}2^{-kh}\sum_{i=1}^{k-1}c_{i,j=k-i}+2^{-lh}\sum_{i=1}^{l}d_{i,j=l+1-i}+2^{-(l+1)h}\sum_{i=1}^{l}g_{i,j=l+1-i}+2^{-(l+1)h}\sum_{i=2}^{l}d_{i,j=l+2-i}+$$
$$+2^{-(l+2)h}\sum_{i=2}^{l}g_{i,j=l+2-i}+2^{-(l+3)h}\sum_{i=3}^{l}d_{i,j=l+3-i}+2^{-(l+3)h}\sum_{i=3}^{l}g_{i,j=l+3-i}+\sum_{k=4}^{l}2^{-lh}\sum_{i=k}^{l}c_{i,j=l+k-i}, \quad (6)$$

Первые два слагаемых в (6) составляют основу $l$ старших $p$ –ричных разрядов произведения, остальные члены – основу младших разрядов. Если в (6) принять $g_{i,j=l+1-i}\approx p=2^h$, $i=\overline{1,l}$ а $d_{i,j=l+2-i}\approx p=2^h$, $i=\overline{2,l}$, то

$$2^{-(l+1)h}\sum_{i=1}^{l}g_{i,j=l+1-i}+2^{-(l+1)h}\sum_{i=2}^{l}d_{i,j=l+2-i}<(2l-1)2^{-lh}, \quad (7)$$

$$2^{-(l+2)h}\sum_{i=2}^{l}g_{i,j=l+1-i}<(l-1)2^{-(l+1)h},\ \ 2^{-(l+2)h}\sum_{i=3}^{l}d_{i,j=l+3-i}<(l-3)2^{-(l+1)h}. \quad (8)$$

Из оценок (7) и (8) следует, что для получения произведения однократной длины $n$ (совпадает с разрядностью сомножителей) с точностью до единицы или половины младшего двоичного разряда $2^{-lh}$ достаточно ограничиться первыми тремя слагаемыми (5), так как старшие разряды исключенного слагаемого $\sum_{k=3}^{l}2^{-(l+k)h}\sum_{i=k}^{k-1}c_{i,j=l+k-i}$ согласно (8) вносят погрешность порядка $(l-3)\times 2^{-(l+1)h}=2^{-(l+1)h+\log_2(l-3)}$.

При достаточно большом $h$ и реальных значениях $l, h>\log_2(l-3)$, например, при $h=8$ и $l=8$ (соответствует 64-разрядным двоичным числам) погрешность за счет отбрасывания младших частичных произведений не превышает $2^{-69}$, что существенно меньше $2^{-65}$ (погрешность произведения однократной длины при симметричном округлении).

Такой подход соответствует вычислению частичных произведений первых $(l+1)$ столбцов матрицы (3) с весовыми множителями $2^{-2h}$ до $2^{-(l+2)h}$ и при сохранении заданной точности обеспечивает существенную экономию вычислительных ресурсов. При

использовании МУ для получения произведения однократной длины $l$ –разрядных $p$ –ричных сомножителей вместо вычисления в общем случае $l^2$ частичных произведений достаточно выполнить $1/2(l^2+3l-2)$ перемножений, что приводит к экономии времени и числа МУ, пропорционально разности

$$l^2-\frac{l^2+3l-2}{2}=\frac{l^2+3l+2}{2}.$$

## 2. Распределенная реконфигурируемая обработка на примере перемножения матриц

Одной из основных особенностей программируемой логики является возможность использования принципа параллельной обработки данных при решении широкого круга задач. Увеличение ресурсов современной программируемой логики и уменьшение их стоимости, в частности ПЛИС типа FPGA фирмы Xilinx, позволяют существенно повысить быстродействие разрабатываемых устройств и реализовать аппаратно алгоритмы, работающие в реальном режиме времени. Распараллеливание вычислений или логических операций может осуществляться как на уровне разрядов представления информации, так и на уровне блоков, выполняющих соответствующие алгоритмы математической модели.

Рассмотрим реализацию алгоритма перемножения матрицы $A=\|a_{ij}\|$ ( $\forall i=1\div n, j=1\div m$ ) на матрицу $B=\|b_{jk}\|$ ( $\forall k=1\div r$ ). Результирующая матрица $C=\|c_{ik}\|$ размером $n\times r$ формируется следующим образом:

$$C=AB=\|a_{ij}\|\times\|b_{jk}\|=\|c_{ik}\|,$$

где

$$c_{ik}=\sum_{i=1}^{n}a_{ij}b_{jk}. \qquad (9)$$

Таким образом, согласно (9), каждый $j$ –й элемент $i$ –ой строки матрицы $A$ последовательно умножается на соответствующий $j$ –й элемент столбца матрицы $B$ и

полученные частичные произведения суммируются, формируя элемент $c_{ik}$ матрицы $C = \| c_{ik} \|$.

Для определения каждого элемента результирующей матрицы используются операции умножения и суммирования частичных произведений. Суммирование может осуществляться двумя способами: накоплением (аккумулированием) частичных произведений при последовательном их поступлении на вход аккумулятора с выхода МУ и параллельном суммировании частичных произведений. Первый способ предполагает наличие блока, выполняющего умножение и суммирование (накопление) полученных частичных произведений. Второй способ использует набор умножителей и многовходовый сумматор для получения элемента результирующей матрицы.

Эти способы реализованы следующими вариантами:

– последовательный (ПС), когда обрабатывающее поле состоит из одного блока, последовательно вычисляющего сумму парных произведений в (9), структурная организация представлена на рис. 1;

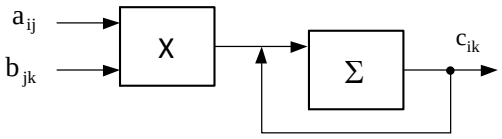


**Рис. 1**. Структура ПС перемножения матриц

– параллельно–последовательный (ПП1), когда обрабатывающее поле содержит множество блоков (рис. 2), число которых соответствуют количеству ($i = n$) строк матрицы $A$, с помощью которых одновременно вычисляется $k$–й ($k = 1 \div r$) элемент $c_{ik}$ столбца и далее последовательно формируются все столбцы матрицы $C = \| c_{ik} \|$ в (9);

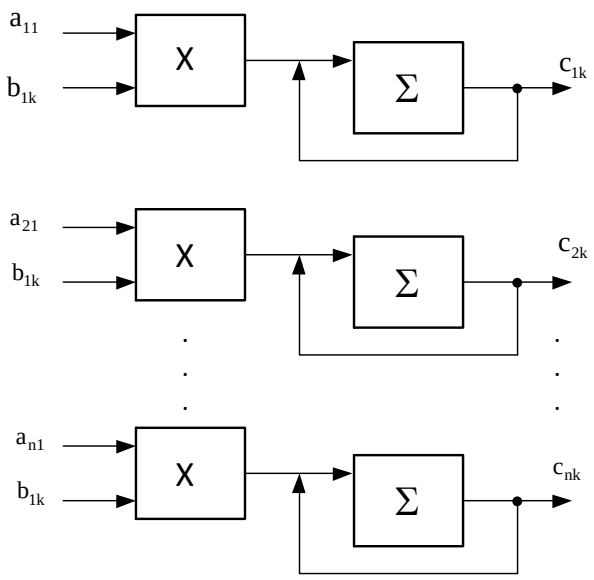


**Рис. 2**. Структура ПП1 перемножения матриц

– параллельно–последовательный (ПП2), когда обрабатывающее поле содержит такое количество блоков (рис. 3), в котором число МУ соответствуют количеству ($i = n$) строк матрицы $A$, параллельно реализуя, таким образом, вычисление одного элемента $c_{ik}$ , а далее последовательно вычисляются остальные элементы $c_{ik}$ матрицы $C = \| c_{ik} \|$ .

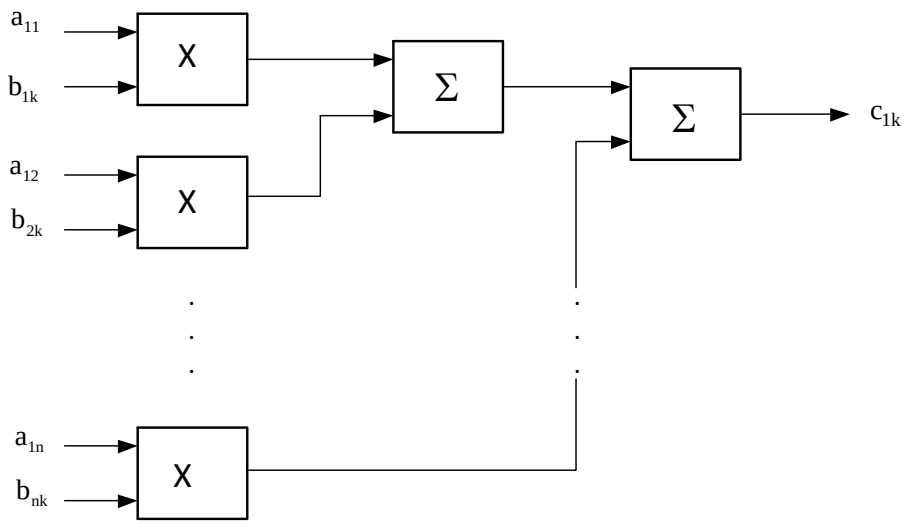


**Рис. 3**. Структура ПП2 перемножения матриц

Распределенная арифметика реализует арифметические операции и основана на использовании логических функциональных генераторов (LUT). Поскольку базовая логическая ячейка ПЛИС типа FPGA фирмы Xilinx содержит LUT, то архитектура FPGA позволяет на их основе реализовать как логические, так и арифметические функции, в том числе умножение, широко используемое в цифровой обработке сигналов.

Рассмотрим реализацию устройства, выполняющего умножение квадратных матриц порядка $m = 10$ для целых 16–разрядных чисел, реализованного в кристалле серии Virtex со степенью быстродействия – 8. Аппаратные затраты на реализацию определяются логической емкостью кристалла, т.е. соответствующим количеством слайсов. Количество затраченных слайсов включает в себя входные, выходные и промежуточные регистры для реализации конвейерного способа вычислений. Время выполнения операции умножения двух 16–разрядных чисел с накоплением 32–разрядной суммы (суммированием результата умножения с числом, находящимся в аккумуляторе) для указанного типа кристалла составляет 6,424 нс. В табл. 1 приведены аппаратные и временные оценки для рассмотренных вариантов реализации.

Таблица 1.

| Вариант реализации алгоритма умножения матриц | Количество умножителей сумматоров | Быстродействие (полное время перемножения матриц), нс | Аппаратные затраты (количество слайсов) |
|---|---|---|---|
| ПС | 1/1 | 6424 | 181 |
| ПП1 | 10/10 | 642,4 | 1810 |
| ПП2 | 10/1 | 890 | 1665 |

## 3. Представление чисел в формате с плавающей точкой

Формат представления чисел с плавающей точкой использует своего рода "подвижное окно" точности, соответствующее масштабу числа. В стандарте IEEE 754 [Hollash, 2018] формат одинарной точности представлен 32 битами – 1 бит для знака, 8 бит для порядка и 23 бита для дроби мантиссы. Однако данный формат предполагает наличие «скрытого» старшего бита ( $f_0$ ), так что мантисса на самом деле представлена 24 битами ( $p = 24$ ).

Представим число с плавающей точкой в следующем виде: $A=(-1)^{sign}\times s^{e}\times F$, где: sign – знак числа; $s$ – основание системы счисления (в данном случае $s=2$); $e$ – порядок (значение которого для формата одинарной точности представлено со смещением на $b=+127$ ); $F=f_0+f$ – мантисса; $f=\left(f_1s^{-1}+f_2s^{-2}+...f_is^{-i}+...+f_{p-1}s^{-(p-1)}\right)$, $(0\le f_i<s)$ – дробь мантиссы; $p$ – разрядность мантиссы ( $p=24$ ).

Если старшая значащая цифра отлична от нуля ( $f_0\neq 0$ ), то число считается нормализованным – $A=(-1)^{sign}\times 2^{(e-b)}\times 1.f$ . Нормализованные числа представляются диапазоном от $\pm 2^{-126}$ до $(2-2^{-23})\times 2^{127}$.

Самые большие и самые малые допустимые величины порядков принимают значения $e_{\max}=+127$ и $e_{\min}=-126$ . Так как имеется $s^{p}$ возможных значений мантиссы, и ( $e_{\max}-e_{\min}+1$) возможных значений показателей, то число с плавающей запятой может быть закодировано следующим количеством битов -$\left[\log_2(e_{\max}-e_{\min}+1)\right]+[\log_2(s^{p})]+1$, где последнее слагаемое (+1) предназначено для знакового разряда.

Значение знакового разряда «0» соответствует положительному числу, а «1» – отрицательному.

Поле порядка должно представлять положительные и отрицательные значения порядка. К фактическому порядку добавляется смещение – для формата одинарной точности это значение равно 127. Таким образом, порядок нулевого значения предполагает, что в поле порядка сохранено значение 127. Например, сохраненное значение порядка 200 указывает порядок (200 – 127) или фактическое значение 73. Значения порядка –127 (все «0») и + 128 (все «1») зарезервированы для специальных чисел.

## 4. Core–блоки функциональных устройств с плавающей точкой

Рассмотрим разработку устройств, выполняющих операции с плавающей точкой (алгебраическое сложение, умножение, деление, сравнение и преобразование форматов) над 32–разрядными операндами в соответствии со стандартом IEEE–754 [Hollash, 2018]. Обобщенная структура функциональных блоков с плавающей точкой представлена на рис. 4 и состоит из трех составных модулей: модуль контроля входных аргументов (МКА); функциональный модуль (ФМ) и модуль формирования результата (МФР). Описание

модулей выполнено на VHDL–языке, при разработке использован синтезатор FPGA Compiler II фирмы Synopsys, для формирования Core–блоков применена система CORE Generator System фирмы Xilinx. Разработанные модули верифицированы методом моделирования с определением временных и аппаратных параметров. Модули представляют собой законченные типовые технические решения и могут быть использованы в других проектах в качестве soft cores.

МКА преобразует входные данные, анализирует их на соответствие стандарту IEEE–754 с формированием соответствующих признаков. Соответствующие числа и информацию относительно классов входных данных выдает как результаты функциональному модулю с плавающей точкой.

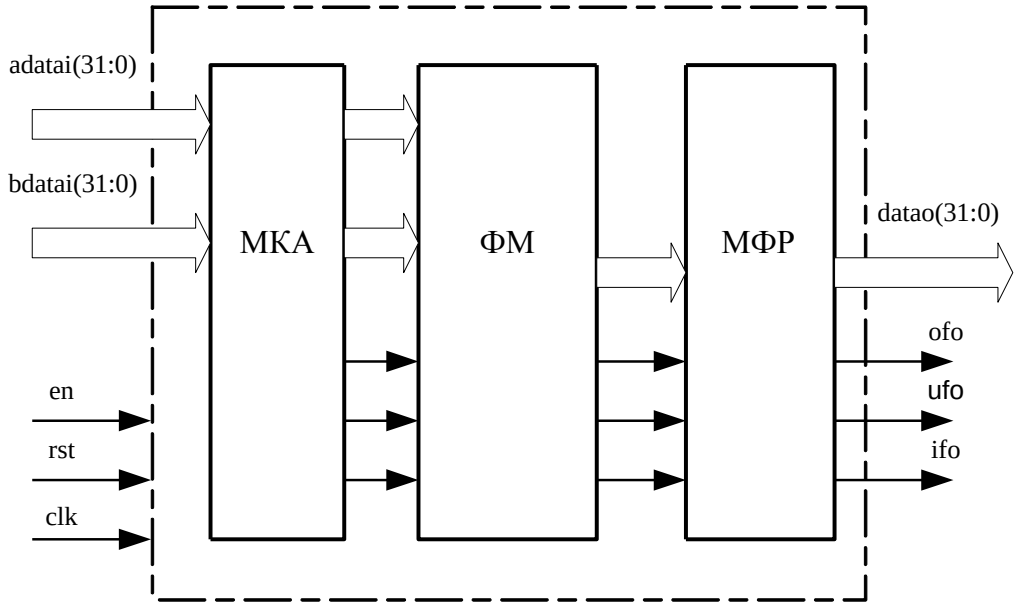


**Рис. 4**. Структура функционального блока с плавающей точкой

ФМ выполняет заданную операцию с плавающей запятой с формированием соответствующих признаков. МФР выполняет согласование формата результирующих данных со стандартом IEEE–754 и окончательную установку флагов. Входы и выходы блока с плавающей точкой не привязаны к фиксированным контактам ввода-вывода конкретного кристалла ПЛИС, поэтому допускается использование произвольного кристалла. Назначение входов и выходов блока показано на рис. 4: clk – глобальный сигнал clock; rst – глобальный сигнал reset; en – сигнал Enable; adatai (31:0) – входная шина

данных A; bdatai (31:0) – входная шина данных B; datao (31:0) – выходная шина данных; ofo – флаг “Overflow”; ufo – флаг “Underflow”; ifo – флаг «Недопустимая операция».

Для согласования полученного результата преобразования со стандартом IEEE–754 необходимо представить числа в нормализованном виде. Поэтому требуется определить месторасположение старшей значащей «1» и выполнить сдвиг в сторону старших значащих бит на требуемое количество разрядов с одновременным вычитанием этого значения из результирующего порядка. Наличие мощных логических ресурсов в кристаллах серии Virtex позволяет эту процедуру ускорить путем быстрого определения количества сдвигов. Тогда, в отличие от реализации последовательного сдвига с одновременным анализом старшего значащего бита, выполняется параллельный сдвиг на требуемое количество разрядов для нормализации мантиссы.

Суммирование с плавающей точкой включает пять строго последовательных операций: сравнение порядков, сдвиг вправо мантиссы меньшего числа, суммирование мантисс, поиск левой единицы мантиссы результата, нормализация мантиссы результата.

Для реализации операции поиска левой единицы предлагается использование приоритетного шифратора. Пусть имеется ( $n = 24$ ) значащих бит мантиссы. Требуется определить номер старшего «ненулевого» разряда и выполнить нормализацию мантиссы $F = \{f_{23}, f_{22}, ..., f_i, ..., f_0\}$.

Приоритетный шифратор представляет собой комбинационную схему, имеющую $n$ входов и ( $Ent\{\log_2 n\}$ ) выходов, которая состоит из двух последовательно соединенных схем - первая выделяет старшую значащую единицу, а вторая ее номер (количество требуемых сдвигов) в операнде.

Первая схема имеет $n$ входов и $n$ выходов, реализуя следующую систему логических уравнений:

$$a_i = f_i \left( \bigcap_{i=i+1}^{(n-1)} (\overline{f_i}) \right), \ \forall i = 0 \div (n-1). \qquad (5)$$

Вторая схема имеет $n$ входов и ( $Ent\{\log_2 n\}$ ) выходов, реализуя следующую систему логических уравнений:

$$y_j = \bigcup_{k=1}^{N=Ent\{(n-1)/(2^{(j+1)})\}} \left[ \bigcup_{i=2^j+(k-1)\times 2^{(j+1)}}^{2^j+(2^j-1)+(k-1)\times 2^{(j+1)}} a_i \right], \ (\forall j = 0 \div (Ent\{\log_2 n\} - 1)). \qquad (6)$$

Таким образом, выражения (5) и (6) позволяют синтезировать приоритетный шифратор произвольной разрядности, представляющий собой параметрический модуль, который может быть использован при разработке новых проектов другими пользователями.

Умножение с плавающей точкой включает два основных действия – вычисление произведения мантисс и нормализацию результирующей мантиссы. Результирующий порядок (как сумма порядков сомножителей) вычисляется параллельно.

При разработке типовых модулей, так же как и при разработке обычных проектов, целесообразно использование уже хорошо отработанных доступных IP–Core.

Рассмотрим пример построения 32–разрядного блока умножения с плавающей точкой. Блок состоит из трех элементов, первые два из которых, в соответствии с рис. 4, входят в функциональный модуль ФМ, а третий – в функциональный модуль МФР [Palagin, 2017].

Первый элемент формирует 24-разрядные операнды для блока умножения (“1” в старшем – 23–м разряде и 23 разряда дроби мантиссы), суммирует порядки перемножаемых чисел (8 разрядов) и определяет знак результата. Второй элемент выполняет операцию умножения и формируется средствами Core–генератора фирмы Xilinx.

Третий элемент выполняет проверку условий и формирование результата. Проверяются следующие условия: если сумма порядков чисел равна или более 255, формируется сигнал переполнения (overflow); если 24 разряд произведения равен “1”, то производится сдвиг произведения на один разряд в сторону младших разрядов и увеличение порядка на единицу; если, после увеличения порядка на единицу, значение порядок становится равным 255, то формируется сигнал переполнения.

При использовании элемента умножения комбинационного типа результат умножения формируется на такте, следующем за тактом регистрации операндов. В тех случаях, когда приходится перемножать массивы чисел, поступающих синхронно с какой-либо тактовой последовательностью CLK, предпочтительно использование элемента умножения конвейерного типа. В разработанном модуле использованы элементы умножения, как комбинационного типа, так и с 6–уровневым (при реализации на LUT) или 2–уровневым (при использовании встроенных блоков умножения разрядности 18x18 в кристаллах Virtex) конвейером, что позволяет за счет увеличения тактовой частоты существенно уменьшить

время перемножения массивов чисел. Для временного согласования в первый элемент модуля в этом случае введены четыре или два последовательно включаемых регистра для конвейерной передачи на выход порядка и знака произведения. Задержка (Latency) между регистрацией первых операндов и регистрацией первого произведения модуля умножения равна 5 или 3 периодам CLK соответственно при использовании 4–уровневого или 2–уровневого конвейерного элемента умножения.

На рис. 5 изображена диаграмма работы модуля контроля МКА, выполненная средствами редактора State Editor.

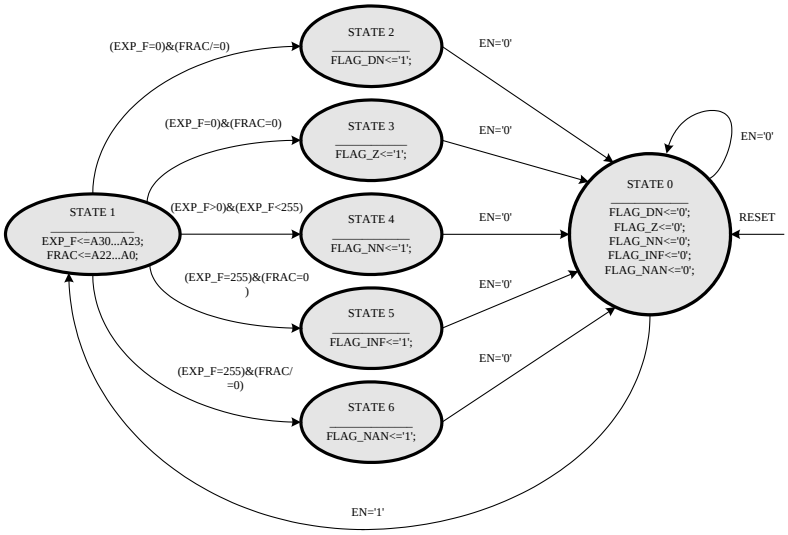


**Рис. 5**. Диаграмма работы модуля МКА, выполненного средствами редактора State Editor

На первом такте при наличии сигнала EN=1 блок переходит в состояние STATE1, на котором из входного операнда A (разряды 0–31) формируются сигналы EXP_F (порядок числа – разряды 23–30) и FRAC (дробь мантиссы – разряды 0–22).

Далее производится проверка условий, при выполнении одного из которых блок переходит на втором такте в одно из состояний (STATE2 – STATE6) с формированием соответствующего флага:

– если значение порядка равно нулю, а дробь мантиссы ненулевая, то входной операнд является ненормализованным числом;

– если значения порядка и дроби мантиссы равны нулю, то входной операнд является нулем;

– если значение порядка больше нуля и меньше 255, то входной операнд является нормализованным числом;

– если значение порядка равно 255 и дробь мантиссы нулевая, то входной операнд является бесконечностью ( $\pm\infty$ );

– если значения порядка равно 255 и дробь мантиссы ненулевая, то входной операнд не является вещественным числом (NAN).

Переход блока в исходное состояние производится по сигналу сброса (RESET) или установке сигнала EN в нулевое состояние.

В кристалле серии Spartan–II (XC2S50–5) блок занимает 46 слайсов (Slices) и работает с тактовой частотой 103 МГц.

Преимущество предложенных реализаций по сравнению с известными достигается за счет оптимального распределения описаний составляющих модулей в разных режимах, а также оригинального приоритетного шифратора, который позволяет определить номер старшей значащей «единицы» для последующего выполнения операции нормализации мантиссы за один временной такт.

В табл. 2 представлены сравнительные оценки аппаратных ресурсов и производительности разработанных модулей умножения, реализованных с использованием Core фирмы Xilinx, с аналогичными модулями фирмы Digital Core Design.

Таблица 2. Сравнительные оценки по аппаратным ресурсам и производительности

| Серия и тип кристалла | Модуль фирмы Digital Core Design | | Модуль с использованием Core фирмы Xilinx | | | |
|---|---|---|---|---|---|---|
| | Ресурсы (кол-во Slices) | Частота, МГц | Без конвейера | | С конвейером | |
| | | | Ресурсы (кол-во Slices) | Частота, МГц | Ресурсы (кол-во Slices) | Частота, МГц |
| Spartan-II 2S200-6 | 970 | 47 | 382 | 42 | 416 | 75 |
| Virtex V300-6 | 963 | 45 | 385 | 38 | 425 | 90 |
| Virtex-II (Multiplier 18x18) 2V250-5 | 677+4 Multiplier 18x18 | 74 | 110+4 Multiplier 18x18 | 52 | 170+4 Multiplier 18x18 | 83 |
| Virtex-II (Multiplier LUT) 2V250-5 | - | - | 386 | 49 | 427 | 128 |

Ресурсы оцениваются количеством слайсов (Slices), каждый из которых содержит 2 логические ячейки (Logic Cell), состоящих из функционального генератора с четырьмя входами, логики ускоренного переноса и запоминающего элемента (триггера). Производительность оценивается частотой синхропоследовательности CLK.

Аппаратные ресурсы оценены относительно известных реализаций: $\Delta Q_i = Q_0 / Q_\mu$ , где: $Q_0$ – аппаратные затраты модуля фирмы Digital Core Design; $Q_\mu$ – аппаратные затраты предложенного модуля; $Q_1$ – аппаратные затраты модуля без конвейера; $Q_2$ – аппаратные затраты модуля с конвейерной организацией.

Для кристалла ПЛИС типа 2S200–6: $\Delta T_1 = T_0 / T_1 = 2,54$; $\Delta T_2 = T_0 / T_2 = 2,33$.

Для кристалла ПЛИС типа V300–6: $\Delta T_1 = T_0 / T_1 = 2,5$; $\Delta T_2 = T_0 / T_2 = 2,27$.

Для кристалла ПЛИС типа 2V250–5: $\Delta T_1 = T_0 / T_1 = 6,15$; $\Delta T_2 = T_0 / T_2 = 9,67$.

Вариант реализации модуля на кристалле 2V250-5 с использованием LUT отсутствует в предложениях фирмы Digital Core Design, однако по аппаратным затратам он сравним с предложенной реализаций на кристалле V300–6, но позволяет работать (примерно на треть) с большей тактовой частотой.

## Заключение

Полученное аналитическим путем число частичных произведений обеспечивает результат одинарной точности. В работе предложены последовательная и последовательно-параллельная структуры устройств перемножения матриц. Приведен пример для таких устройств перемножения матриц с временными и аппаратурными оценками их реализации.

Синтезированные функциональные блоки с плавающей точкой, совместимые со стандартом IEEE–754, могут быть использованы в качестве библиотечного элемента при разработке сложных вычислительных устройств.

## Библиография

[Майоров, 1970] С.А. Майоров, Г.И. Новиков. Структура цифровых вычислительных машин. Ленинград: Машиностроение, 1970. 480 с.

[Палагин, 2006] Палагин А.В., Опанасенко В.Н. Реконфигурируемые вычислительные системы. Киев: Просвіта, 2006. 295 с.

[Opanasenko, 2017] Opanasenko V.N., Kryvyi S.L. Synthesis of neural-like networks on the basis of conversion of cyclic Hamming codes. *Cybernetics and Systems Analysis*. 2017. Vol. 53, N.4. P. 627–635. DOI: DOI 10.1007/s10559-017-9965-z.

[Hollash, 2018] Hollash S. IEEE Standard 754 Floating Point Numbers. Available at https://steve.hollasch.net/cgindex/coding/ieeefloat.html.

[Palagin, 2017] Palagin A., Opanasenko V. The implementation of extended arithmetic's on FPGA-based structures. Proceedings of the 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS'2017. (21–23 September 2017, Bucharest, Romania). 2017. Vol. 2. P. 1014–1019. DOI: DOI 10.1109/IDAACS.2017.8095239.

## Сведения об авторах

***Опанасенко Владимир Николаевич*** *– профессор, доктор технических наук, ведущий научный сотрудник Института кибернетики им. В.М. Глушкова НАН Украины, Украина, Киев, 03187, просп. Глушкова, 40;* ***e-mail****: opanasenkovm@nas.gov.ua*

***Крывый Сергей Лукьянович*** *– профессор, доктор физико-математических наук, профессор Киевского национального университета им. Тараса Шевченко, Украина, Киев, 03187, просп. Глушкова, 4д, Факультет кибернетики;* ***e-mail:*** *krivoi@i.com.ua*

***Завьялов Станислав Борисович*** *– кандидат технических наук, директор ООО «Радионикс», Украина, Киев;* ***e-mail:*** *radionix13@gmail.com.*

# FPGA-based single accuracy multiplication devices

**Volodymyr Opanasenko, Sergii Kryvyi, Stanislaw Zavyalov**

***Abstract**: The number of partial products is determined to ensure a single precision result. The paper proposes a serial and serial-parallel structure of matrix multiplication devices. Time and hardware assessments of their implementation on the basis of FPGA are given. Functional floating point blocks are synthesized that are compatible with the IEEE-754 standard, which can be used as a library element in the development of complex computing devices.*

***Keywords**: single precision, serial structure, serial-parallel structure, multiplication, FPGA.*

# К ВОПРОСУ ПОСТРОЕНИЯ ОПТИМАЛЬНОГО ДЕРЕВА РЕШЕНИЙ

**Виталий Величко**

***Аннотация:*** *В работе приведена формальная постановка задачи построения оптимального дерева решений в терминах задачи бинарной идентификации. Оптимальное дерево решений определено как дерево минимального размера и способное без ошибок классифицировать все объекты из обучающей выборки. Рассматривается случай, когда все атрибуты объектов являются номинальными. Для отбора наилучших правил используется мера информационного выигрыша на основе вычисления условной энтропии. В работе показано, что для задачи построения оптимального дерева решений, сформулированной в терминах задачи бинарной идентификации, существует полиномиальный алгоритм ее решения при условии определения стоимости теста (логического правила) как функции свойств теста. Вычислительная сложность приведенного алгоритма ограничена полиномом третьей степени мощности множества объектов обучающей выборки. Для упрощения рассуждений принято, что для каждого значения целевого атрибута существует не менее одного теста, условная энтропия которого на множестве объектов из обучающей выборки равна 0. Задача построения оптимального дерева решений не является NP-полной задачей при условии задания ограничений на определение функции стоимости логического правила (теста).*

***Ключевые слова:*** *задача бинарной идентификации, оптимальное дерево решений, вычислительная сложность алгоритма.*

***ITHEA Keywords****: F.2.2 [Analysis of algorithms and problem complexity]: Nonnumerical Algorithms and Problems, I.2.6 [Artificial intelligence]: Learning, I.2.4 [Artificial intelligence]: Knowledge Representation Formalisms and Methods.*

## Введение

Деревья решения являются популярным подходом к решению задач Data Mining [Субботин, 2019]. Они позволяют получить иерархическую структуру классифицирующих правил, которая имеет вид дерева [Quinlan, 1986]. Деревья решений могут оценивать значения категориальных (номинальных) атрибутов, имеющих конечное число дискретных значений, а также количественных атрибутов. Древовидные модели, в которых целевая переменная может принимать дискретный набор значений, называются деревьями классификации; в этих древовидных структурах листья представляют метки значений целевого атрибута, а ветви представляют конъюнкцию значений нецелевых атрибутов, которые ведут к этим меткам значений целевого атрибута. Деревья решений, где целевая переменная может принимать непрерывные значения (обычно действительные числа), называются деревьями регрессии [Xindong at all, 2008].

Формально дерево решений можно определить как способ построения классификационной или регрессионной модели в виде древовидной структуры. Узлы дерева подразделяются на решающие узлы (в которых представлены правила) и листья - узлы, дающие решения. Под правилом понимается логическая конструкция, представленная в виде "ЕСЛИ... ТО..." ("IF-THEN").

В процессе обхода дерева в каждом узле в зависимости от проверяемого условия принимается определенное решение – перемещение по той или иной ветке дерева от корня к «листьевым» (конечным) вершинам. В «листьевой» вершине дерева содержится искомое значение интересующего атрибута. В узлах бинарных деревьев решений

ветвление идет только в двух направлениях, т.е. существует только 2 ответа на поставленный вопрос: «да» и «нет». Обучение деревьев решений выполняется индуктивно на основе прецедентов – наблюдений за состоянием моделируемого объекта или процесса. Рассмотрим пример дерева решений, полученного на основе анализа таблицы «Объект-свойство» (Таблица 1). Задача, состоит в том, чтобы на основе анализа примеров ситуаций из таблицы (Таблица 1) сформировать в явном виде правила определяющие значения целевого атрибута (играть в гольф или не играть) в зависимости от значений нецелевых атрибутов (предикторов). Названия и значения атрибутов в какой-то мере условны и служат главным образом для иллюстрации построения и использования деревьев решений.

***Таблица 1. Пример таблицы «Объект-свойство» играть в гольф***

| | Атрибуты (Предикторы) | | | | Целевой атрибут (Target) |
|---|---|---|---|---|---|
| Наименования атрибутов | Наблюдение | Температура | Влажность | Ветер | Играть в гольф |
| Примеры объектов с известными значениями атрибутов | Дождь | Жарко | Высокая | Нет | Нет |
| | Дождь | Жарко | Высокая | Да | Нет |
| | Пасмурно | Жарко | Высокая | Нет | Да |
| | Солнечно | Умеренная | Высокая | Нет | Да |

| | | | | | |
|---|---|---|---|---|---|
| | Солнечно | Прохладная | Нормальная | Нет | Да |
| | Солнечно | Прохладная | Нормальная | Да | Нет |
| | Пасмурно | Прохладная | Нормальная | Да | Да |
| | Дождь | Умеренная | Высокая | Нет | Нет |
| | Дождь | Прохладная | Нормальная | Нет | Да |
| | Солнечно | Умеренная | Нормальная | Нет | Да |

Извлеченные из таблицы правила представлены в виде дерева на Рисунке 1. Сами правила в дизъюнктивной нормальной форме (ДНФ) приведены в Таблице 2. Правила из таблицы получены с помощью аналитической платформы Deductor 4.3 [Deductor, 2020]. Для каждого правила вычислена поддержка (количество примеров из таблицы «объект-свойство» рис.1 для которых правило выполняется) и достоверность (количество примеров для которых выполнение правила дает правильный результат).

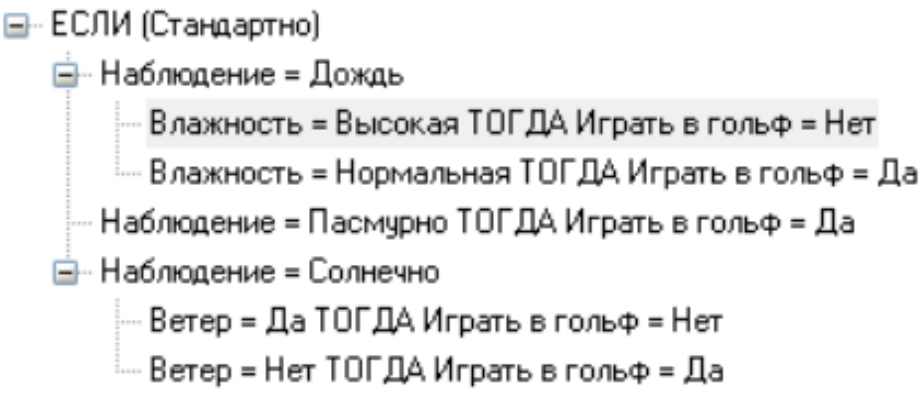


Рисунок 1. Дерево решений играть в гольф

***Таблица 2. Классифицирующие правила играть в гольф***

| N | Условие | Следствие (Целевой атрибут: **Играть в гольф**) | Поддержка | | Достоверность | |
|---|---|---|---|---|---|---|
| | | | % | Кол-во | % | Кол-во |
| 1 | Наблюдение = Дождь **И** Влажность = Высокая | Нет | 30,00 | 3 | 100,00 | 3 |
| 2 | Наблюдение = Дождь **И** Влажность = Нормальная | Да | 10,00 | 1 | 100,00 | 1 |
| 3 | Наблюдение = Пасмурно | Да | 20,00 | 2 | 100,00 | 2 |
| 4 | Наблюдение = Солнечно **И** Ветер = Да | Нет | 10,00 | 1 | 100,00 | 1 |
| 5 | Наблюдение = Солнечно **И** Ветер = Нет | Да | 30,00 | 3 | 100,00 | 3 |

## Принцип построения дерева решений

Принцип построения дерева [Quinlan, 1986] следующий. Дерево строится «сверху вниз» от корня. Начинается процесс с определения, какой атрибут (предиктор) следует выбрать для проверки в корне дерева. Для этого каждый атрибут исследуется на предмет, насколько хорошо он классифицирует набор данных (разделяет множество примеров на группы по одинаковым значениям целевого атрибута). Когда разделяющий атрибут выбран, для каждого его значения создается ветка дерева, набор данных разделяется в соответствии со значением к каждой ветке, процесс повторяется рекурсивно для каждой ветки. Можно сформулировать правило для выбора атрибута следующим образом: выбранный атрибут должен разбивать множество объектов так, чтобы получаемые в итоге подмножества состояли из объектов, имеющих одно значение целевого атрибута, или были максимально приближены к этому, т.е. количество объектов, которые имеют другие значения целевого атрибута в каждом из этих множеств было как можно меньше. Также проверяется заданный критерий остановки ветвления дерева, который позволяет ограничить выделение детализированных и малозначащих правил. Целью применения критериев остановки ветвления является выделение наиболее полных и точных правил. Под термином наиболее полные правила будем понимать правила, которые истинны для наибольшего количества объектов, имеющих одинаковое значение целевого атрибута. Полнота правила характеризуется его относительной поддержкой. Под термином наиболее точные правила будем понимать правила, которые истинны только для одного значения целевого атрибута. Точность правила характеризуется его относительной достоверностью.

Алгоритмы построения дерева решения, использующие приведенный принцип, такие как ID3, C4.5 являются «жадными». На каждой итерации алгоритма, которая состоит в выборе разделяющего атрибута, максимизируется определенный «локальный» критерий оптимальности в предположении, что получившееся дерево в целом будет «оптимальным». Существуют различные критерии выбора атрибута для расщепления

дерева. Наиболее известные – «мера информационного выигрыша» (англ. information gain measure, gain ratio) или мера энтропии и индекс Gini. При помощи индекса Gini атрибут выбирается на основании расстояний между распределениями значения целевого атрибута. Часто алгоритмы построения деревьев решений дают слишком детализированные деревья, которые имеют много узлов и ветвей. Это связано с явлением переобучения, для избежания которого используется алгоритм отсечения ветвей (pruning). Также для устранения этого недостатка используют метод комитетов из решающих деревьев [Ho,1998]. Популярность использования деревьев решений связана с наглядностью и и возможностью получения правил в явном виде. Алгоритмы построения деревьев решений реализуют наивный принцип последовательного просмотра атрибута. Но дерево решений принципиально не способно находить наиболее полные и точные правила в данных. Приведем простой пример, иллюстрирующий это утверждение. Дана таблица «объект-свойство» (Таблица 3), содержащая 8 примеров, 2 значения целевого атрибута, каждое из которых представлено 4 примерами.

***Таблица 3. «Объект-свойство» пример 2***

| Атрибуты / Объекты | a0 | a1 | a2 | Target |
|---|---|---|---|---|
| 1 | FALSE | FALSE | A | t0 |
| 2 | FALSE | FALSE | A | t0 |
| 3 | TRUE | TRUE | D | t0 |

| Атрибуты / Объекты | a0 | a1 | a2 | Target |
|---|---|---|---|---|
| 4 | TRUE | TRUE | E | t0 |
| 5 | FALSE | TRUE | B | t1 |
| 6 | FALSE | TRUE | C | t1 |
| 7 | TRUE | FALSE | F | t1 |
| 8 | TRUE | FALSE | F | t1 |

Очевидно, что наилучшее дерево решений для приведенного примера содержит 4 правила и имеет вид, приведенный в Таблице 4. Необходимо уточнить, что для значения целевого атрибута t0 вместо правила №1 может быть также сформировано правило a2 = A, а для значения целевого атрибута t1 вместо правила №4 может быть также сформировано правило a2 = F. Данные правила по критерию меры информационного выигрыша эквивалентны правилам №1 и №4 и общее количество правил (четыре) в наилучшем дереве решений все равно не изменяется.

***Таблица 4. Оптимальные классифицирующие правила пример 2***

| N | Условие | Следствие (Целевой атрибут, Target) | Поддержка | | Достоверность | |
|---|---|---|---|---|---|---|
| | | | % | Кол-во | % | Кол-во |
| 1 | a0 = FALSE **И** a1 = FALSE | t0 | 25,00 | 2 | 100,00 | 2 |
| 2 | a0 = TRUE **И** a1 = TRUE | t0 | 25,00 | 2 | 100,00 | 2 |
| 3 | a0 = FALSE **И** a1 = TRUE | t1 | 25,00 | 2 | 100,00 | 2 |
| 4 | a0 = TRUE **И** a1 = FALSE | t1 | 25,00 | 2 | 100,00 | 2 |

“Жадные” алгоритмы построения дерева решения не выбирают атрибуты a0 и a1 для разделения множества примеров по значениям целевого атрибута (Таблица 3), потому что мера информационного выигрыша для любого из атрибутов a0 или a1 равна 0 (каждое из 2 значений атрибутов a0 и a1 равновероятно для обеих значений целевого атрибута t0 и t1, которые также равновероятны). Для построения дерева решений всегда будет выбран атрибут a2. Правила в ДНФ, полученные с помощью "жадных" алгоритмов построения дерева, приведены в Таблице 5.

***Таблица 5. Классифицирующие правила при использовании "жадных" алгоритмов пример 2***

| N | Условие | Следствие (Целевой атрибут, Target) | Поддержка | | Достоверность | |
|---|---|---|---|---|---|---|
| | | | % | Кол-во | % | Кол-во |
| 1 | a2 = A | t0 | 25,00 | 2 | 100,00 | 2 |
| 2 | a2 = B | t1 | 12,50 | 1 | 100,00 | 1 |
| 3 | a2 = C | t1 | 12,50 | 1 | 100,00 | 1 |
| 4 | a2 = D | t0 | 12,50 | 1 | 100,00 | 1 |
| 5 | a2 = E | t0 | 12,50 | 1 | 100,00 | 1 |
| 6 | a2 = F | t1 | 25,00 | 2 | 100,00 | 2 |

Всего выделено 6 правил, из которых 4 имеют поддержку 1, т.е. справедливы только для одного объекта и не обладают обобщающей способностью.

## Вычислительная сложность построения оптимальной процедуры бинарной идентификации

В общем случае, при использовании "жадных" алгоритмов, задача построения оптимального дерева решений NP-полная [Hyafil and Rivest, 1976]. Это очевидно, потому что для таких алгоритмов локальные оптимальные решения не могут гарантировать получение глобального оптимального дерева решений. NP-полнота задачи построения оптимального дерева решений рассматривается в работах [Garey, 1972], [Hyafil and Rivest, 1976]. В [Hyafil and Rivest, 1976] приведена обобщенная формулировка задачи бинарной идентификации, без задания каких-либо ограничений на множество бинарных тестов и свойств элементов множества бинарных тестов. Под оптимальным бинарным деревом понимается дерево, которое минимизирует ожидаемое количество тестов, необходимых для идентификации неизвестного объекта. Авторами доказывается, что в общем случае построение оптимальных бинарных деревьев решений является NP-полной задачей. Также утверждается, что эта модель идентична модели, изученной в [Garey, 1972]. Рассмотрим более подробно данную модель, и приведем формальную постановку задачи бинарной идентификации, которая была предложена в [Garey, 1972]. Дано:

a) конечное множество из $n$ объектов, $O_1,O_2,...,O_n$ , которые являются возможными для некоторого неизвестного объекта;

b) соответствующее множество $n$ известных вероятностей, $p_1,p_2,...,p_n$ , удовлетворяющих $0 < p_i \le 1$ , где $p_i$ - вероятность того, что неизвестный объект является $Q_i$ , и поскольку мы рассматриваем только один неизвестный объект – $\sum_{i=1}^{n} p_i = 1$;

c) конечное множество из $m$ бинарных тестов или вопросов, $Q_1,Q_2,...,Q_m$ , каждый из которых, является функцией $Q_j : \{O_1,O_2,...,O_n\} \to \{TRUE,FALSE\}$ ,

где $Q_j(O_i)$ определяет результат применения теста $Q_j$ для неизвестного объекта $O_i$;

d) соответствующее множество из $m$ стоимостей $C_1,C_2,...,C_m$, где $C_j$ затраты на выполнение теста $Q_j$, каждое $C_j > 0$.

Табличная форма описания задачи приведена в таблице 6. В столбцах приведены объекты, в строках тесты, а ячейках – результат применения теста $Q_j$ к объекту $O_i$ – $Q_j(O_i)$.

***Таблица 6. Табличная форма описания задачи бинарной идентификации***

| | | $p_1$ | $p_2$ | … | $p_n$ |
|---|---|---|---|---|---|
| | | $O_1$ | $O_2$ | … | $O_n$ |
| $C_1$ | $Q_1$ | *TRUE* | *FALSE* | … | *TRUE* |
| $C_2$ | $Q_2$ | *FALSE* | *FALSE* | … | *TRUE* |
| … | … | … | … | … | … |
| $C_m$ | $Q_m$ | *FALSE* | *TRUE* | … | *FALSE* |

Стоимость процедуры бинарной идентификации будет определяться как ее средняя стоимость, рассчитанная по всем тестам как сумма произведения стоимости каждого теста на вероятность того, что тест будет использован при применении процедуры бинарной идентификации. Эквивалентно, среднюю стоимость можно вычислить как сумму по всем объектам произведения вероятности объекта, на сумму затрат на вопросы, которые задаются, для идентификации объекта, когда он является неизвестным объектом. Тогда оптимальная процедура бинарной идентификации - это такая, которая обеспечивает минимальную стоимость всех процедур для одной и той же проблемы. В [Garey, 1972] доказывается, что вычислительная сложность построения оптимальной процедуры бинарной идентификации с помощью алгоритма обратной индукции пропорциональна to $m \cdot n \cdot 2^n$ , т.е. оптимальная процедура бинарной идентификации является NP-полной задачей.

## Формальная постановка задачи построения оптимального дерева решений решений

Рассмотрим формальную постановку задачи построения оптимального дерева решений на основе анализа таблицы «объект-свойство» в терминах задачи бинарной идентификации. Рассмотрим случай, когда все атрибуты имеют конечное число дискретных значений.

Обозначим:

$V$ – конечное множество значений целевого атрибута (target) $V = \left\{ v_j \middle| j = \overline{1, n_1} \right\}$, $n_1 = |V|$ – количество значений целевого атрибута;

$W$ – конечное множество значений нецелевых атрибутов (предикторов) $W = \bigcup_{i=1}^{n_2} W^i$ , $n_2$ – количество нецелевых атрибутов, $W^i$ – множество

значений *i*-ого нецелевого атрибута, $W^i = \left\{ r_j^i \middle| j = \overline{1,n_3} \right\}$, где $r_j^i$ – *j*-тое значение *i-г*о атрибута, $n_3$ – количество значений *i-г*о нецелевого атрибута;

*A* – конечное множество известных объектов $A = \left\{ a_k \middle| k = \overline{1,n_4} \right\}$, где $n_4$ – количество известных объектов. Для каждого объекта $a_k \in A$ определено только одно значение $v_{a_k} \in V$ и множество $R_{a_k} = \left\{ r_j^i \middle| r_j^i \in W^i, j \in \{1...n_3\}, i = \overline{1,n_5} \right\}$, где $r_j^i$ – *j*-тое значение *i-г*о нецелевого атрибутов, $n_5$ – количество нецелевых атрибутов, которые определены для объекта $a_k$. Иначе, каждый объект $a_k$ можно представить в синтаксисе логики высказываний следующей формулой $a_k = \bigvee_{i=1}^{n5} r_j^i$, которая определяет имликацию: $\bigvee_{i=1}^{n5} r_j^i \to v_{a_k}$. Для каждого значения целевого атрибута $v_j$ известно множество $V_j = \left\{ a_k \middle| k = \overline{1,n_6} \right\}$, где $n_6$ – количество объектов из множества *A*, имеющих значение целевого атрибута $v_j$. Для каждого $v_j$ определено $p_j = \frac{|V_j|}{|V|}$ – вероятность наблюдения значения целевого атрибута $v_j$, $0 < p_j \le 1$, $\sum_{j=1}^{n_1} p_j = 1.$ Обобщенная форма таблицы «объект-свойство», содержащей исходные данные для построения дерева решений приведена в таблице 7;

***Таблица 7. Таблица «объект-свойство» с исходными данными для построения дерева решений***

| | | Предикторы ($W$) | | | | Целевой атрибут ($V$ ,Target) | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $W^1$ | $W^2$ | | $W^{n2}$ | $v_1$ | $v_2$ | … | $v_{n1}$ |
| $A$ | $a_1$ | $r_1^1$ | $r_1^2$ | … | $r_1^{n2}$ | *FALSE* | *FALSE* | … | *TRUE* |
| | $a_2$ | $r_2^1$ | $r_2^2$ | … | $r_2^{n2}$ | *FALSE* | *TRUE* | … | *FALSE* |
| | … | … | … | … | … | … | … | … | … |
| | $a_{n4}$ | $r_{n3}^1$ | $r_{n3}^2$ | … | $r_{n3}^{n2}$ | *TRUE* | *FALSE* | … | *FALSE* |

$Q$ – конечное множество из $m$ бинарных тестов или вопросов, $Q_1,Q_2,...,Q_m$, каждый из которых, является функцией $Q_j(a_k)$: $\{a_1,a_2,...,a_{n4}\} \to \{TRUE,FALSE | \forall v_j \in V\}$ , где $Q_j(a_k)$ определяет результат применения теста $Q_j$ для объекта $a_k \in A$. Каждый тест (правило дерева решений) соответствует импликации в синтаксисе логики высказываний: $Q_j : x_1 \wedge x_2 \wedge ... \wedge x_{n_7} \to y$ , где $x_1, x_2, ..., x_{n_7}$ определены на элементах множества $W$ - $\{r_j^i | r_j^i \in W^i\}$ , $n_7$ – количество элементов в логической формуле, а $y$ определено на множестве $\{v_j | v_j \in V\}$;

$C$ – соответствующее множество из $m$ стоимостей $C_1,C_2,...,C_m$ , где $C_j$ затраты на выполнение теста $Q_j$, каждое $C_j > 0$.

Необходимо найти множество тестов суммарной минимальной стоимости, необходимых для однозначной правильной идентификации каждого объекта $a_k \in A$, т.е. таких тестов, которые для объекта $a_k$ принимают значение ИСТИНА только для значения $v_{a_k} \in V$ и значение ЛОЖЬ для всех остальных значений $v_j \in V$.

Табличная форма описания задачи приведена в таблице 8. В столбцах Target приведены значения целевого атрибута, в строках тесты (логические правила), а в ячейках – результат применения теста $Q_j(a_k)$ к объекту $a_k \in A$ для различных значений целевого атрибута.

***Таблица 8. Табличная форма описания задачи построения оптимального дерева решений***

| | | Target | | | |
|---|---|---|---|---|---|
| | | $p_1$ | $p_2$ | … | $p_{n1}$ |
| | $V$ | $v_1$ | $v_2$ | … | $v_{n1}$ |
| $C_1$ | $Q_1(a_k)$ | *FALSE* | *FALSE* | … | *FALSE* |
| $C_2$ | $Q_2(a_k)$ | *FALSE* | *TRUE* | … | *TRUE* |
| … | … | … | … | … | … |
| $C_m$ | $Q_m(a_k)$ | *TRUE* | *FALSE* | … | *FALSE* |

Если сравнить Таблицы 6 и 8, то очевидно, что постановки задач построения оптимального дерева решений и оптимальной процедуры бинарной идентификации схожи. Однако в задаче построения процедуры бинарной идентификации стоимость теста $C_m$ никак не связана ни со свойствами теста, ни с вероятностями наблюдения объектов. Рассмотрим алгоритм построения оптимального дерева решений на основе анализа таблицы «объект-свойство» (табл. 7) при условии задания стоимости теста как функции свойств теста и вероятности наблюдения значения целевого атрибута.

## Процедура построения оптимального дерева решений и ее вычислительная сложность

Уточним определение понятия оптимального дерева решений, построенного на основе анализа таблицы «объект-свойство». Оптимальным будем считать дерево, которое содержит минимальное количество правил (имеющих максимальную поддержку), которые однозначно и безошибочно (со 100% достоверностью) определяют значения целевого атрибута для каждого объекта, входящего в таблицу «объект-свойство». Кроме этого, в соответствии с выводами статистической теории обучения распознаванию [Вапник и Червоненкис, 1974], должны быть отобраны более простые правила, т.е. включающие минимальное количество элементов (предикторов) при прочих равных характеристиках (поддержка и достоверность).

Таким образом, процедура построения оптимального дерева решений включает 2 этапа:

1) Нахождение всех возможных правил на основе анализа таблицы «объект-свойство;
2) Отбор наилучших правил из найденных на первом этапе.

Рассмотрим первый этап.

Найдем какое максимальное количество тестов в синтаксисе логики высказываний в ДНФ может быть задано таблицей «объект-свойство». Тестом является каждая ячейка таблицы, соответствующая определенному значению атрибута объекта. Таким образом, множество всех возможных тестов включает множество *W* (таблица 7), а максимальное количество таких тестов равно $|W|$. Каждый объект или строка таблицы также является тестом (соответствует конъюнкции всех атрибутов объекта), т.е. в множество всех возможных тестов входит множество *A* . Количество таких тестов равно $|A|$. Тесты, которые представлены множеством *A*, имеют 100% достоверность, потому что по условию задачи для каждого объекта $a_k \in A$ определено только одно значение целевого атрибута $v_{a_k} \in V$ . Правда такие тесты не решают задачи построения обобщенной логической формулы класса объектов, потому что обладают относительной поддержкой равной $1/|A|$. Какие еще тесты, пригодные для идентификации значений целевого атрибута (target), может задавать таблица объект-свойство? Такими могут быть фрагменты описаний объектов, которые повторяются в описаниях не менее двух различных объектов. Фрагменты описаний объектов, которые не повторяются, не могут правилами (тестами). Выделять такой фрагмент из полного описания объекта в качестве теста, а не использовать в качестве теста полное описание объекта, нет каких-либо оснований. Оценим теоретически возможное количество фрагментов описаний объектов, которые повторяются в описаниях различных объектов из таблицы «объект-свойство» более одного раза. Для нахождения таких фрагментов необходимо выполнить операцию сравнения описаний объектов друг с другом. Формально, для $\forall a_j \in A$ и $\forall a_i \in A, i, j = \overline{1,n_4}, i \neq j, n_4 = |A|$ , необходимо найти $R_{ji} = R_{a_j} \cap R_{a_i}$ и если $R_{ji} \neq \varnothing$ сформировать тест (правило) $Q_{ij} = \bigvee_{l=1}^{n7} r_l \left| r_l \in R_{ji}, n_7 = |R_{ji}| \right.$. Таким образом максимально возможное количество тестов будет равно сумме членов арифметической прогрессии,

от 1 до $|A|-1$ с шагом 1, если пересечения описаний всех объектов различны и ни одно из них не равно пустому множеству. По формуле суммы членов арифметической прогрессии количество таких тестов равно $|A|\cdot(|A|-1)/2$. Тогда максимальное количество тестов $m$, которое задает исходная таблица «объект-свойство», можно найти по формуле: $m \le |A|\cdot(|A|-1)/2 + |W| + |A| = \frac{|A|^2+|A|}{2} + |W|$.

Количество операций для нахождения всех возможных тестов можно оценить как $O\left(\frac{|A|^2-|A|}{2}\right)$.

Рассмотрим второй этап. В качестве критерия, характеризующего качество правила, можно использовать меру информационного выигрыша, выраженную через количество информации. Рассмотрим ее более подробно. Базовым понятием всей теории информации является понятие энтропии. Информационная энтропия – мера неопределённости некоторой некоторой ситуации или системы, в частности мера непредсказуемости появления какого-либо символа первичного алфавита. При отсутствии информационных потерь энтропия численно равна количеству информации на символ передаваемого сообщения. Информационная двоичная энтропия для атрибута (случайной величины) рассчитывается по формуле Шеннона: $H(Y) = -\sum_{i=1}^{n} \frac{N_i}{N} \bullet \log_2\left(\frac{N_i}{N}\right)$, где $n$ — количество значений атрибута, $N_i$ – количество примеров, которые имеют i-ое значения атрибута, $N$ – общее число примеров в множестве. Фактически информационная двоичная энтропия определяет минимальное число бит, которые необходимы для кодирования выбранного атрибута для надежной передачи информации в виде двоичных чисел. Энтропия, в отличии от дисперсии, не зависит от типа распределения вероятностей случайных величин. Если две случайные величины $X$ и $Y$, каким-то образом связаны

друг с другом, то знание одной из них, уменьшает неопределенность значений другой [Коротаев, 2003]. Оставшаяся неопределенность оценивается условной энтропией. Условная энтропия $X$ при условии знания $Y$ определяется как: $H\left(X\middle|Y\right)=\sum_{k=1}^{K}P\left(Y_k\right)\sum_{m=1}^{M}P\left(X_m\middle|Y_k\right)\bullet\log_2\left(P\left(X_m\middle|Y_k\right)\right)$, где – условные вероятности (вероятность $m$-го значения $X$ при условии $Y=Y_k$), количество значений случайных величин $X$ и $Y$ ($M$ и $K$) не обязательно совпадают. Чтобы рассчитать $H\left(X\middle|Y\right)$, рассчитывают $K$ энтропий $X$, соответствующих фиксированному $Y_k$ далее суммируют результаты с весами $P\left(Y_k\right)$. Условная энтропия всегда меньше безусловной, точнее:

$$0\le H\left(X\middle|Y\right)\le H\left(X\right).$$

Нулевое значение условной энтропии соответствует однозначной зависимости $X$ от $Y$, максимальное значение – полной независимости $X$ и $Y$.

Условная энтропия – это предельно общая характеристика степени зависимости некоторых переменных [Коротаев, 2003]. Ее можно сравнить с корреляцией, но если корреляция характеризует линейную связь переменных, то условная энтропия характеризует любую связь. Информационный выигрыш от использования одной случайной величины для прогнозирования другой случайной величины определяется разностью между безусловной и условной энтропиями этих случайных величин.

В задаче построения дерева решений (табличная форма описания задачи приведена в таблице 8) меру информационного выигрыша для каждого правила можно вычислить по формуле:

$Gain\left(Q_j\right)_{v_i}=H\left(target=v_i\right)-H\left(target=v_i\middle|Q_j\right)$, где $Gain\left(Q_j\right)_{v_i}$ – мера информационного выигрыша от использовании правила $Q_j$ для определения значения целевого атрибута $v_i$ на множестве объектов в

исходной таблице «объект-свойство»; $H(target = v_i)$ – информационная энтропия $v_i$ значения целевого атрибута (target), $H(target = v_i | Q_j)$ – условная энтропия $v_i$ значения целевого атрибута при условии использовании правила $Q_j$.

Информационная энтропия и условная энтропия вычисляются на множестве объектов из исходной таблице «объект-свойство». Для упрощения дальнейших рассуждений предположим, что для каждого значения целевого атрибута существует не менее одного теста, условная энтропия которого на множестве объектов из таблицы «объект-свойство» равна 0.

Второй этап алгоритма построения оптимального дерева решений включает следующие шаги.

1) Для каждого теста $Q_i$, $i = \overline{1,m}$ и всех объектов $a_k \in A, k = \overline{1,n_4}, n_4 = |A|$ определить $v_i \in V$ для которых $Q_i(a_k) = True$ и для каждого $v_j \in V, j = \overline{1,n_1}, n_1 = |V|$ сформировать множества объектов $V_{v_j}^{Qi} = \{a_k | Q_i(a_k) = True\}$. На данном шаге выполняется добавление объектов в множества по значениям целевого атрибута, для которых текущий тест принимает принимает значение истина. Определим систему множеств $X^{Qi} = \left\{ V_{v_j}^{Q_i} \middle| V_{v_j}^{Q_i} \neq \varnothing \right\}$. Если $X^{Qi}$ содержит более одного множества то необходимо удалить данный тест из множества $Q$. Таким образом в множестве $Q$ остаются тесты, которые принимают значения истина только для одного значения целевого атрибута на множестве объектов из исходной таблицы «объект-свойство». Количество операций на данном шаге можно оценить как $O\left( m \cdot |A| = \left( \frac{|A|^2 + |A|}{2} + |W| \right) \cdot |A| = \frac{|A|^3 + |A|^2}{2} + |W| \cdot |A| \right)$.

2) Для каждого теста $Q_i$ из множества $Q$ для всех объектов $a_k \in V_{v_j}^{Qi}$ проверить выполнение условия $v_{a_k} = v_j$. Если условие не выполняется, то удалить данный тест из множества $Q$. Количество операций на данном шаге не превышает $O\left(m \cdot |A|\right)$. Шаги 1 и 2 алгоритма решают задачу отбора правил со 100% достоверностью.

3) Определить значение стоимости $C_i$ теста $Q_i$ в соответствии со свойствами теста и требованиями, которые предъявляются к наилучшему тесту:

$$C_j = F\left(Gain\left(Q_j\right)_{v_i}, p_j, n7\right)$$

Мера информационного выигрыша $Gain\left(Q_j\right)_{v_i}$ никак не учитывает количество элементов в логической формуле теста $Q_i$. Тесты, состоящие из минимального числа элементов, являются лучшими при прочих равных характеристиках [Вапник и Червоненкис, 1974]. Поэтому в функцию стоимости теста необходимо в качестве аргумента добавить $n_7$ – количество элементов в логической формуле. Для минимизации количества вычисляемых тестов, вначале должны вычисляться наиболее вероятные тесты, т.е. такие, которые проверяют наличие у объекта наиболее вероятного значения целевого атрибута – $p_j$. Лучшие тесты должны иметь минимальную стоимость, поэтому стоимость теста должна быть прямо пропорционально количеству элементов в логической формуле, обратно пропорционально вероятности значения целевого атрибута и мере информационного выигрыша. В качестве функции $F$, удовлетворяющей приведенным требованиям может быть выбрана, например, следующая:

$$C_j = \left(1 \Big/ Gain\left(Q_j\right)_{v_i}\right).\left(1 / p_j\right) \cdot n7$$

Количество операций на данном шаге не превышает $m$.

4) Отсортировать множество тестов $Q$ по возрастанию $C_i$. Количество операций на данном шаге не превышает $O\left(|m|\cdot\log\left(|m|\right)\right)$.

5) Для каждого объекта $a_k \in A$ последовательно проверить истинность тестов из отсортированного множества $Q$ в порядке возрастания стоимости теста – $C$. При получении результата истина, отметить данный тест и перейти к следующему объекту $a_k \in A$. Количество операций на данном шаге не превышает $m\cdot|A|$. Использование меры информационного выигрыша на 3 шаге и шаги 4 и 5 предназначено для отбора правил с максимальной поддержкой. Шаг 5 предназначен для выполнения условия определения значения целевого атрибута для каждого объекта, входящего в таблицу «объект-свойство», с помощью выбранных тестов.

6) Отмеченные тесты из $Q$ представляют собой множество тестов минимальной стоимости, необходимых для однозначной идентификации каждого объекта из множества $A$ и соответствуют оптимальному дереву решений в форме множества логических функций.

Каждый шаг приведенного алгоритма имеет полиномиальную сложность (сложность ограничена полиномом третьей степени мощности множества объектов обучающей выборки). Таким образом сложность приведенного алгоритма полиноминальна.

## Выводы и дискуссия

В работе приведена формальная постановка задачи построения оптимального дерева решений на основе анализа таблицы «объект-свойство» в терминах задачи бинарной идентификации. Рассматривается случай, когда все атрибуты имеют конечное число дискретных значений. В работе показано, что существует полиномиальный алгоритм построения оптимального дерева решений при условии определения стоимости теста как функции свойств теста (логического правила). Задача построения оптимального дерева решений не является NP-полной задачей при

условии задания ограничений на определение функции стоимости логического правила. Можно показать, что принятое допущение о существовании для каждого значения целевого атрибута не менее одного теста с условной энтропией равной 0 на множестве объектов из таблицы «объект-свойство» не влияет на полученный вывод о полиномиальной сложности задачи построения оптимального дерева решений. Для построения полиномиального алгоритма решения задачи в этом случае необходимо предварительно построить специальную структуру, в которой логические правила (тесты) упорядочиваются по вхождению более короткого правила в более длинное и добавить операцию отрицания в логические правила. Примером подобной структуры может служить растущая пирамидальная сеть, в которой определен алгоритм нахождения логических правил [Гладун, 1994]. Детальное описание алгоритма построения оптимального дерева решений на основе растущей пирамидальной сети и оценка его сложности будет приведено в дальнейших работах.

**Литература**

[Субботин, 2019] Субботин С. А. Построение деревьев решений для случая малоинформативных признаков // Радіоелектроніка, інформатика, управління. e-ISSN 1607-3274. – 2019. – № 1. – с.122-131. DOI 10.15588/1607-3274-2019-1-12

[Quinlan, 1986] Quinlan J. R. Induction of decision trees // Machine learning. – 1986. – V. 1, № 1. – pp. 81–106.

[Xindong at all, 2008] Wu Xindong, Kumar Vipin; J. Ross Quinlan, Ghosh Joydeep, Yang Qiang, Motoda, Hiroshi; McLachlan, Geoffrey J.; Ng, Angus; Liu, Bing; Yu, Philip S.; Zhou, Zhi-Hua (2008-01-01). Top 10 algorithms in data mining. Knowledge and Information Systems. 14 (1): 1–37. doi:10.1007/s10115-007-0114-2. ISSN 0219-3116

[Deductor, 2020] ООО «Лаборатория баз данных» Электронный ресурс, режим доступа: http://www.basegroup.ru.

[Ho,1998] Ho T.K. (1998). The Random Subspace Method for Constructing Decision Forests // IEEE Transactions on Pattern Analysis and Machine Intelligence. 20 (8): 832–844. doi:10.1109/34.709601.

[Hyafil and Rivest, 1976] Laurent Hyafil, Ronald L. Rivest. Constructing optimal binary decision trees is np-complete // Information Processing Letters. – 1976. –V. 5, № 1. – pp. 15–17. DOI:10.1016/0020-0190(76)90095-8

[Garey, 1972] M. R. Garey. Optimal Binary Identification Procedure // SIAM Journal on Applied Mathematics, Vol. 23, No. 2 (Sep., 1972), pp. 173-186.

[Вапник и Червоненкис, 1974] Вапник В.Н., Червоненкис А.Я. Теория распознавания образов, статистические проблемы обучения. – М.: Наука, 1974. – 416 с.

[Коротаев, 2003] С. М. Коротаев. Энтропия и информация – универсальные естественнонаучные понятия // Электронный ресурс, режим доступа: http: //www.chronos.msu.ru/old/rreports/korotaev_entropia/korotaev_entropia.htm

[Гладун, 1994] Гладун В.П. Процессы формирования новых знаний. – Sofia: SD “Педагог-6”, 1994. – 192 с.

**Об авторе**

***Виталий Величко*** *– Институт кибернетики им. В. М. Глушкова НАН Украины; Кандидат технических наук. Старший научный сотрудник. Проспект Глушкова 40, Киев, Украина, 03187; e-mail: aduisukr@gmail.com*

*Основные направления научных исследований: индуктивный логический вывод, компьютерные онтологии, информационные системы с обработкой объектов естественного языка*

# To the Problem of Constructing the Optimal Decision Tree

**Vitalii Velychko**

***Abstract****: The paper presents a formal statement of the problem of constructing an optimal decision tree in terms of a binary identification problem. An optimal decision tree is defined as a minimum size tree and capable of classifying all objects from the training set without errors. The case is considered when all attributes of objects are nominal. To select the best rules, a measure of information gain is used based on the calculation of conditional entropy. The paper shows that there is a polynomial algorithm for solving the problem of constructing an optimal decision tree, formulated in terms of a binary identification problem. The condition for the existence of the algorithm is the determination of the cost of the test (logical rule) as a function of the test properties. The computational complexity of the above algorithm is limited by the third degree polynomial of the power of the set of objects of the training sample. For the sake of simplicity, it is assumed that for each value of the target property there is at least one test, the conditional entropy of which on the set of objects from the training set is equal to 0. The problem of constructing an optimal decision tree is not an NP-complete problem if constraints are set on determining the cost function of a logical rule (test).*

***Key words:*** *binary identification problem, optimal decision tree, computational complexity class P.*

# СРЕДСТВА ТРАНСДИСЦИПЛИНАРНОГО ПРЕДСТАВЛЕНИЯ ИНФОРМАЦИОННЫХ РЕСУРСОВ РАЗНЫХ СТИЛЕЙ

**Светлана Гайко, Виталий Приходнюк**

***Аннотация**: Статья касается проблемы эффективного использования разнородных информационных ресурсов (ИР). Огромное количество естественноязычных документов, накопленных в сети, остаются пассивными источниками знаний. Для эффективной работы с такими документами, прежде всего, требуется их структуризация, которая не может производиться вручную, учитывая необходимые для этого трудозатраты. Также необходимо определить способ представления результатов структуризации ИР, т.к. от этого зависят дальнейшие возможности при работе с ними. Важнейшей из таких возможностей является интеграция ИР разных стилей, описывающих различные предметные области (ПдО) в едином информационном пространстве. В статье предлагается модель интеграции разнородных документов, реализующая принцип трансдисциплинарности. Описана технология автоматизированной обработки естественноязычных текстов (ЕЯТ) различных стилей и формирования на их основе интерактивных документов. Рассмотрен механизм отображения их структуры, которым является нарративный дискурс. Представлена математическая модель данной технологии, модель поведения системы (в виде UML-диаграмм), а также архитектура программного средства, созданного на основе описанных моделей.*

***Ключевые слова**: трансдисциплинарность, онтология, растущая пирамидальная сеть (РПС), нарративный дискурс, функциональные стили.*

***ITHEA классификация**: H. Information Systems – H.1 MODELS AND PRINCIPLES – H.1.2 User/Machine Systems.*

## Введение

В последние десятилетия стала актуальной необходимость решения большого количества сложных многофакторных проблем в различных сферах человеческой деятельности. В этих условиях повысилась потребность в компьютерных средствах, позволяющих использовать ИР из разных областей знаний. Наряду с этим, ЕЯТ до сих пор остаются самой распространенной формой представления знаний. Такие знания являются пассивными, поскольку большинство документов созданы на основе использования различных информационных технологий и стандартов. В основном, они относятся к классу слабоструктурированных, а по совокупности и характеру изложения, к классу Больших Данных (Big Data) [Mayer-Schönberger, 2013]. Преобразование таких пассивных систем знаний, отображающихся в виде документов, которые содержат описания определенных процессов и их свойств, в интерактивную форму является весьма актуальной проблемой. Однако для этого необходимо реализовать когнитивные процедуры их преобразования, что, как минимум, определяет условия реализации взаимодействия с этими системами, уже активных знаний [Стрижак, Трансдисциплінарна інтеграція, 2014], [Стрижак, 2020].

Таким образом можно выделить основные задачи компьютерной обработки естественноязычных документов. К ним относятся:

1) автоматизированная структуризация разнородных ИР;

2) представление результатов обработки в виде интерактивных документов;

3) интеграция ИР релевантных задачам конкретного пользователя (эксперта).

Решение поставленных задач связано с рядом технологических и методологических проблем. Это обусловлено как фактом накопления сверхбольших объемов информации, так и высокой интенсивностью процессов сетевого взаимодействия.

Учитывая это, наиболее сложной является именно третья задача интеграции ИР, которая сводится не к простому объединению контекстов нарративов, отвечающих заданной тематике, а обеспечивает активное использование свойств связей между контекстами [Стрижак, Трансдисциплінарна інтеграція, 2014].

В статье описана модель трансдисциплинарного представления разностилевых ИР, лежащая в основе построения информационно-аналитической системы, призванной решать задачи как обработки и структуризации ЕЯТ, так и их конструктивной интеграции.

## Обзор литературы

Категория трансдисциплинарности имеет широкую трактовку и применение в современной науке. В сфере информационных технологий данная категория трактуется как гиперсвойство множественной частичной упорядоченности элементов информационной среды [Стрижак, Трансдисциплінарна інтеграція, 2014]. То есть, категория трансдисциплинарности позволяет рассматривать интеграцию информационных ресурсов как некий процесс использования любых релевантных контекстов, и взаимодействия сетевых информационных систем. Данная концепция представлена в работах [Палагин, 2014], [Широков, 2017], [Величко, 2017].

Одним из возможных подходов к реализации трансдисциплинарной интеграции ИР является онтологический подход. Трансдисциплинарные онтологии обеспечивают корректное агрегирование различных тематических процессов путем формирования структурированной

совокупности информационных объектов-концептов ПдО, которые определяются как единый тип данных. Технология их использования в сетевой среде, в которой активируются процессы взаимодействия сложных информационных систем, позволяет установить над ИР, которые активно используются, отношение частичного порядка. Онтологические аспекты трансдисциплинарной интеграции ИР изложены в работе [Стрижак, Онтологические аспекты, 2014]. В работах [Басюк, 2017], [Палагин, 2012] посвященных вопросам онтологического инжиниринга в целом, представляется также широкая библиография по данному вопросу.

Механизмом отображения структуры разностилевых ИР является нарративный дискурс [David, 2012]. Данный механизм подробно описан в статье [Стрижак, 2020]. Суть его заключается в рассмотрении неструктурированных сетевых документов как нарративов, которые представляют собой множество объектов, являющихся концептами, и множество объектов, которые являются классами. Такие объекты, в свою очередь, способны к взаимодействию (свойство дискурса), и это дает возможность отображать связность двух и более нарративов. Построив таксономическую структуру как каждого документа, так и метатаксономию содержания всех используемых документов ПдО, достигается отображение связности документов и процессов взаимодействия с ними.

## Входные данные и методы

Модель трансдисциплинарного представления ИР различных стилей является описанием технологии автоматизированной обработки разностилевых документов глобальной среды и представление их в форме, пригодной для дальнейшего превращения в активный формат интерактивных баз знаний. Практическая реализация такой технологии осуществляется на основе WEB-ориентированного сервиса Когнитивная информационная технология (КИТ) «Полиэдр».

Представленная в статье технология предусматривает применение метода построения РПС [Гладун, 2004], а также метода рекурсивной редукции [Приходнюк, 2018].

Для построения онтологического классификатора стилей экспертным образом был обработан массив научных, учебных, законодательных, ведомственных, публицистических и, частично, художественных документов. В результате была построена таксономическая структура функциональных стилей языка, а также определены атрибуты объектов, относящихся к основным стилям.

## Основная часть

Модель трансдисциплинарного представления ИР различных стилей состоит из математической модели и модели поведения системы (в виде UML-диаграмм), а также архитектуры программного средства, созданного на ее основе.

### Математическая модель

Поскольку механизмом отображения структуры разностилевых ИР является нарративный дискурс, прежде всего, необходимо формализовать данное понятие.

Как отмечалось выше, естественноязычные документы являются пассивными системами знаний. Они, фактически, представляют собой определенные тексты, которые отображаются в виде последовательного изложения определенных концептов, устойчиво заданных условий их существования, описаний их свойств и функциональностей и тому подобное. Их можно представить в виде:

$$O_{nr} = \langle X(K), p \rangle, \quad (1)$$

где $O_{nr}$ – документ с последовательно определенными описаниями;

$X$ – концепты;

$K$ – описания концептов (контексты);

$p$ – отношение строгого порядка, определяет условия существования концептов $X$ в тексте.

То есть, нарративом является текст, который пассивно излагает определенную систему знаний, и имеет признаки онтологической системы, в данном случае, имеет выколотое значение онтологии [Стрижак, Онтологические аспекты, 2014], [Басюк, 2017], [Палагин, 2012], [Палагин, 2016], [Приходнюк, 2018].

Операциональнисть произвольного текста вида (1) определяется следующими гиперсвойствами [Стрижак, Трансдисциплінарна інтеграція, 2014], [Стрижак, 2020]: рефлексия – $\Re_f$, рекурсия – $\Re_k$, и редукция – $\Re_d$. Когнитивный характер указанных гиперсвойств реализуется в их интерпретации гиперфункциями типа: анализ больших данных, их структурирование, синтез, выбор и принятие решений, которые согласно [Amit, 2005] являются когнитивными.

Данные гиперсвойства образуют определенное замкнутое множество $\Re 3$, которое обеспечивает связывание и динамическое изменение упорядочения контекстных описаний нарратива $O_{nr}$ [Malyshevsky, 1998].

Это значит, что выколотую онтологию вида (1) можно расширить множеством $\Re 3$ и гиперфункцией ее интерпретации $F_{\Re 3}$:

$$O_{nr} = \langle X(K), p \rangle \to O_{nd} = \langle X(K), \Re 3, F_{\Re 3} \rangle \qquad (2)$$

Через преобразование (2), подстановкой множества гиперсвойств $\Re 3$ и функциональным расширением $F_{\Re 3}$, вместо отношения строгого упорядочения $p$ можно получить онтологическую систему $O_{nd}$, которая

обеспечивает различные виды связей между концептами $X$ и контекстами $K$.

Автоматизированная обработка разностилевых документов состоит из трех этапов:

– классификация ЕЯТ – определение его стиля;

– идентификация информации из текста согласно определенного стиля;

– трансдисциплинарная интеграция информации, содержащейся в различных обработанных текстах.

Для корректной обработки текста согласно его стиля необходимо иметь описания собственно существующих стилей. Естественным способом представления такой классификации является онтологический реестр стилей в виде онтологии $O_S$ (рис. 1).

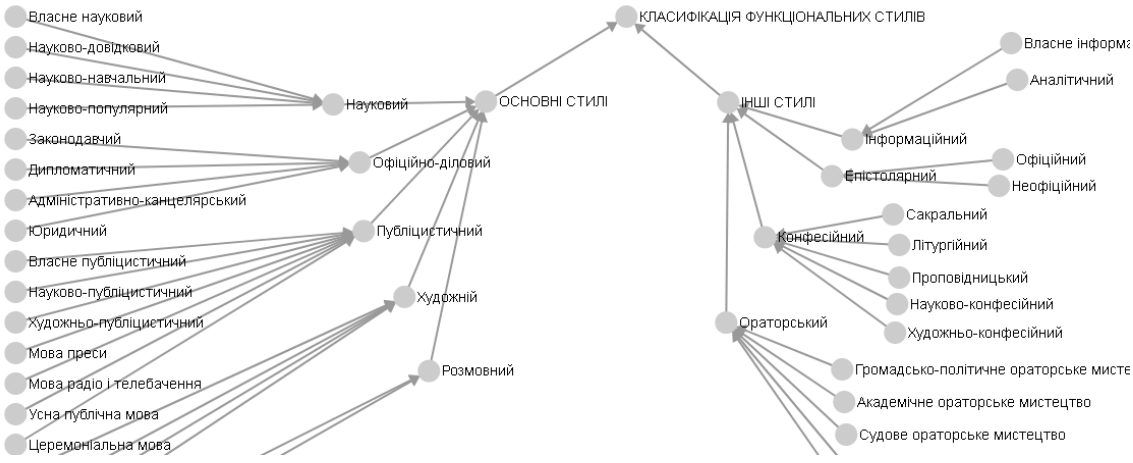


*Рисунок 1 Онтологический реестр стилей*

Поскольку данный реестр является онтологией, он может быть представлен как:

$$O = \langle X, R, A' \rangle,$$

где $X$ – множество концептов, представленных собственно стилями и группами стилей;

$R$ – множество связей между концептами;

$A'$ – атрибуты концептов, которые задают классификационные характеристики соответствующих стилей.

Задача классификации текстов по стилю решается путем построения РПС. Онтология стилей $O_S$ может быть преобразована в РПС. Для этого нужен промежуточный шаг – выделение таксономии из онтологии:

$$O_S \xrightarrow{G_T} T_S$$

Таксономия $T_S$ имеет структуру:

$$T_S = \langle X_T, R_T \rangle$$

где $X_T$ – множество объектов, принадлежащих таксономии;

$R_T$ – множество связей, содержащихся в таксономии.

Множество $X_T$ состоит из объектов $X_S$ начальной онтологии $O_S$, а также множества объектов, представляющих классификационные свойства соответствующих стилей (построенных на основе атрибутов $A'_S$ объектов $X_S$):

$$X_T = X_S \cup X(A'_S)$$

Множество $R_T$ содержит взаимосвязи между объектами $X_T$, представляющие стили, и объектами $X(A'_S)$, которые представляют классификационные характеристики объектов:

$$R_T = R(X_S, X(A'_S))$$

Для таксономии $T_S$ можно построить такую РПС, которая будет унивалентной ей:

$$T_S \xrightarrow{G_\psi} \psi_S$$

$$T_S \cong \psi_S$$

Для решения задачи идентификации информации в тексте необходимо представить классификационные атрибуты, содержащиеся в онтологии, в форме, пригодной для работы с методом рекурсивной редукции. Одной из таких форм может быть активная онтология, которая имеет вид:

$$O_t = \langle X, R, F, A, D, R_S \rangle$$

где $X$ – множество концептов;

$R$ – конечное множество семантически значимых отношений между концептами;

$F$ – конечное множество функций интерпретации, заданных на концептах и / или отношениях (отметка, какую информацию считывать);

$A$ – конечное множество аксиом, которые используются для записи всегда истинных высказываний (определений и ограничений) в терминах тематики ПдО;

$D$ – множество дополнительных определений понятий в терминах тематики ПдО (текстовые описания);

$R_S$ – множество ограничений, определяющих область действия понятийных структур определенной тематики ПдО.

Для представления онтологического реестра в такой форме следует интерпретировать множество атрибутов следующим образом:

$$O_{St} = \langle X, R, F(A'), A(A'), D(A'), R_S(A') \rangle$$

Для этого необходимо построить $G_t$ – преобразование интерпретации атрибутов:

$$O_S \xrightarrow{G_{Ot}} O_{St}$$

При этом:

- Конечные множества функций интерпретации $F$ и дополнительных ограничений $R_S$ создаются с помощью специализированной процедуры;
- Множество дополнительных определений понятий $D$ может быть представлено с помощью определенных атрибутов без дополнительной интерпретации;
- Конечное множество аксиом $A$ не используется в задаче классификации, поэтому можно принять $A = \varnothing$.

Активная онтология может быть преобразована в набор правил для рекурсивного редуктора. Каждое правило имеет следующую структуру:

$$g = \left\langle f_{ap}^{g}, f_{tr}^{g} \right\rangle,$$

где $f_{ap}^{g}$ – функция применимости, определяющая, может ли правило быть применено к определенному набору входной информации;

$f_{tr}^{g}$ – функция трансформации, задающая преобразование набора входной информации.

Правило такого вида может быть сформировано с помощью соответствующего преобразования:

$$R_S\left(A'\right) \xrightarrow{G_{rd}} f_{ap}^{g}$$

$$F\left(A'\right) \xrightarrow{G_{rd}} f_{tr}^{g}$$

где $G_{rd}$ – преобразование формирования редуктивных правил.

Созданная таким образом база правил может быть использована для формирования специализированного оператора редукции, предназначенного для создания выколотых онтологий:

$$O_{nr} = \langle X, R, K(X) \rangle$$

где $K(X)$ – контексты объектов $X$, являющие собой подмножество атрибутов $A'$.

Выколотая онтология также может быть преобразована в РПС:

$$O_{nr} \xrightarrow{G_\psi} \psi_{nr}$$

Также дополнительно для решения задачи трансдисциплинарной интеграции информации РПС необходимо расширить контекстами:

$$W^T \to X_t \left( W^T \right)$$

где $W^T$ – контекст;

$X_t$ – концепт, который содержится в контексте.

Преобразование нарратива на принципах трансдисциплинарной интеграции формирует из "выколотых онтологий" "таксономию нарративного дискурса". Преобразование осуществляется с помощью функции поиска:

$$Q_S(H,T) = \left\{ H \left\langle \left\{ V(I) \times V(T) \right\} \right\rangle \right\}$$

где $H$ – индекс, полученный в результате индексации массива нарративов сетевых документов с помощью специализированной функции индексации $Q_H$;

$V(I), V(T)$ – идентификаторы лексемы $I$ и таксономии $T$ нарратива документа $O_{nd}$ соответственно.

Функция поиска позволяет формировать связи между контекстами всех лексических единиц множества сетевых нарративных документов. Данная операция является основной для построения функции контекстной связности:

$$Q_C(T) = \bigcup_{x \in T} \left( Q_S \left( Q_H(C), L_x \right) \right)$$

где $C$ – множество нарративов сетевых документов $O_{nd}$, которые определяют содержательность информационной среды, в рамках которого осуществляется связывание;

$T$ – таксономия, с лексическими единицами которой осуществляется связывание;

$L_x$ – текстовое представление контекста лексической единицы $x$, принадлежащей таксономии $T$.

Множество индексов $\{H\}$, которое формируется на основе применения к множеству $C$ функции индексации $Q_H$, формирует индексную зону $\tilde{H}$ всех нарративов сетевых документов $O_{nd}$:

$$\tilde{H} = \left\{ \{H\} \times T \right\}$$

Таким образом, функцию контекстной связности можно представить в виде:

$$Q_C(I) = \tilde{H} \times C \times \tilde{T} \times \Re 3$$

Функция контекстной связности создает условия для формирования нарративного дискурса на основе семантико-лексической и концептографической обработки всех нарративов сетевых документов $O_{nd}$.

## Модель поведения системы

Модель поведения системы приводится для представления взаимодействий, отношений и зависимостей компонентов, из которых состоит система структурирования разностилевых ИР.

Модель поведения системы может быть представлена структурой:

$$M_D = \langle d_{use}, d_{act}, d_{seq} \rangle$$

где $d_{use}$ – UML-диаграмма вариантов использования системы;

$d_{act}$ – UML-диаграмма активности системы;

$d_{seq}$ – UML-диаграмма взаимодействия системы.

UML-диаграмму вариантов использования программной системы структуризации разностилевых документов представлено на рис. 2.

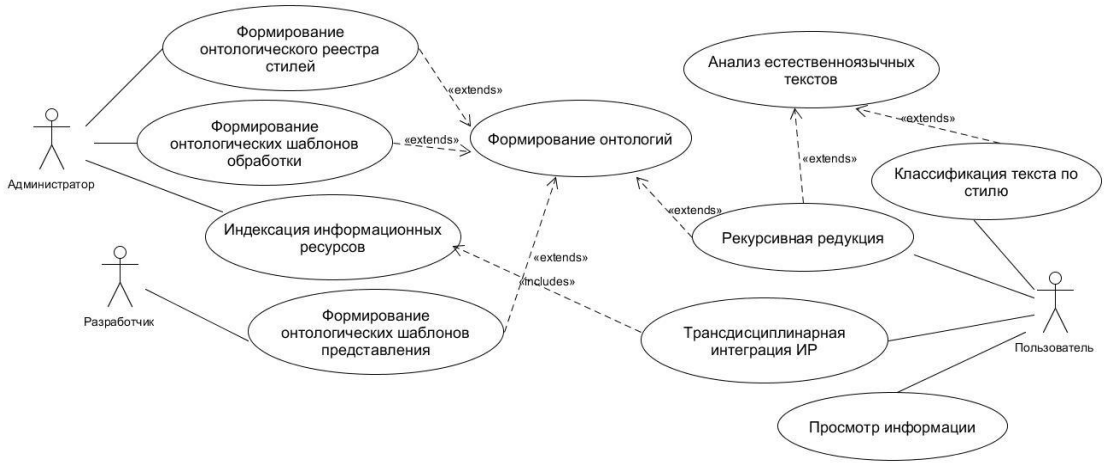


*Рисунок 2 Варианты использования системы*

Данная диаграмма отражает участников процесса структурирования разностилевых документов и их основные возможности. Так, в рамках системы могут действовать «Пользователь», «Администратор» и

«Разработчик». Пользователь осуществляет классификацию текста по стилю, применяет рекурсивную редукцию для получения онтологий, просматривает информацию, а также формирует трансдисциплинарную интеграцию ИР. Администратор системы формирует и наполняет онтологический реестр стилей, формирует онтологические шаблоны обработки и осуществляет индексацию ИР. Разработчик, имеющий доступ ко всем функциям системы, создает онтологические шаблоны представления.

Алгоритм работы с системой структурирования документов различных стилей показывает диаграмма активности (рис. 3).

Пользователь может применять редукцию текстов, осуществлять их классификацию и пользоваться режимом трансдисциплинарного представления. В режиме администрирования создаются шаблоны представления и обработки, а также осуществляется анализ существующих и необходимых стилей.

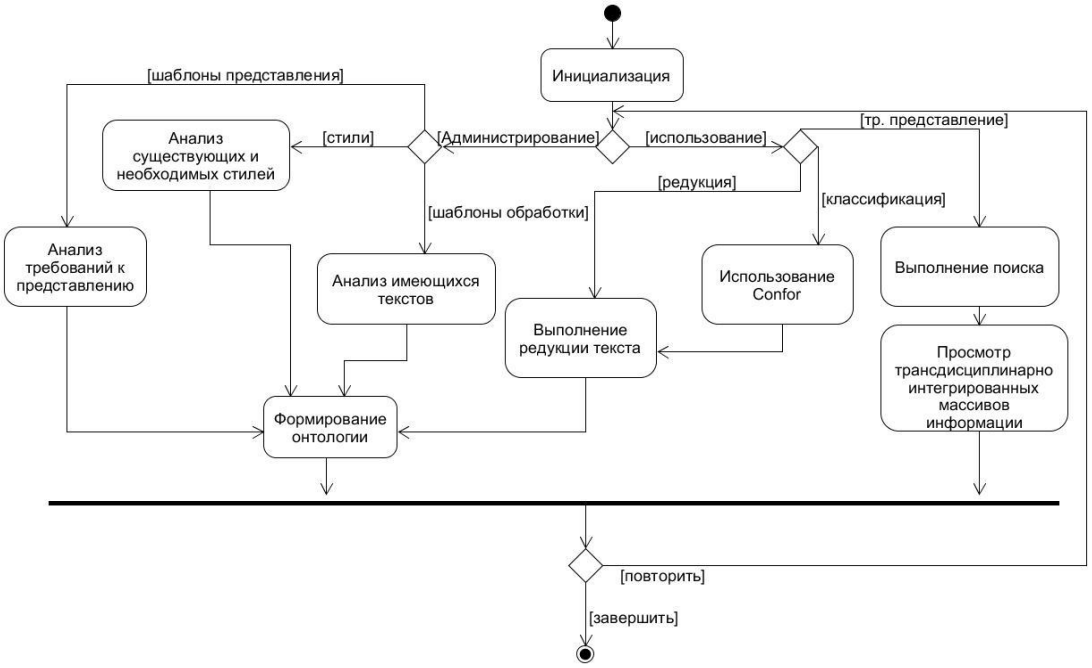


*Рисунок 3 Диаграмма активности системы*

На рис. 4 представлена диаграмма взаимодействия.

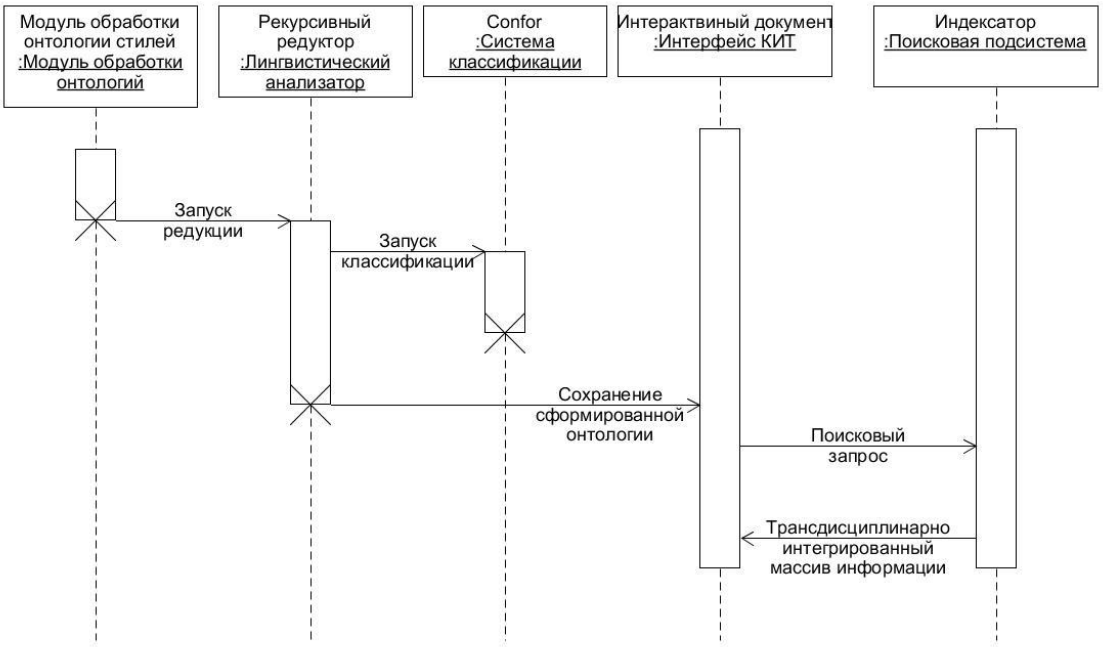


*Рисунок 4 Диаграмма взаимодействия системы*

Взаимодействие подсистем происходит в следующей последовательности:

1) В модуле обработки онтологий происходит обработка онтологии стилей;

2) Запускается редукция ЕЯТ к которой подключается классификация текста обрабатываемого по стилю;

3) Сформированная онтология отображается интерфейсом КИТ «Полиэдр» в виде интерактивного документа;

4) Элементы полученного интерактивного документа могут выступать в качестве поискового запроса, выполнив который с помощью индексации, система предоставляет пользователю трансдисциплинарно интегрированный массив информации.

## Архитектура системы

Для обеспечения надежной и быстрой работы системы трансдисциплинарной интеграции ЕЯТ различных стилей используется трехуровневая архитектура, включая уровень представления, уровень логики и уровень данных (рис. 5).

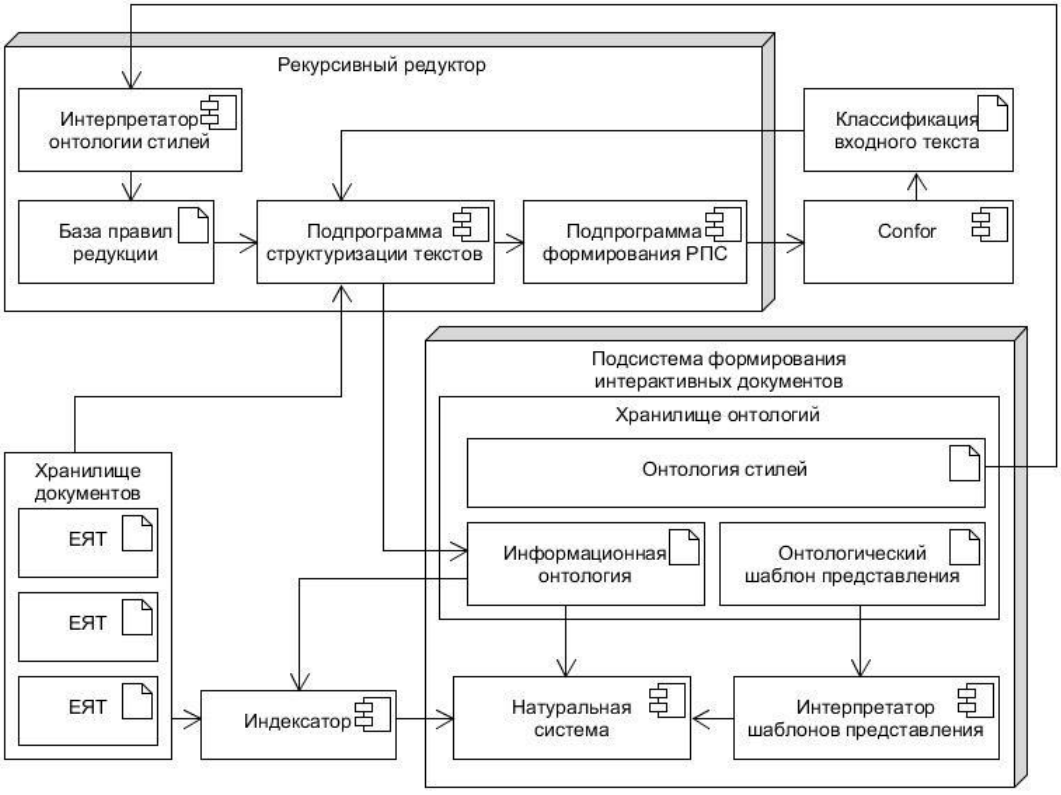


*Рисунок 5 Базовая архитектура системы структурирования документов различных стилей*

Система состоит из двух подсистем: подсистемы, осуществляющей структурирование текстов («Рекурсивный редуктор»), и подсистемы, обеспечивающей пользовательский интерфейс («Подсистема формирования интерактивных документов»).

Система предназначена для работы с распределенным массивом ЕЯТ, которые могут быть получены из различных источников. Центральным элементом системы является онтология стилей, содержащаяся в хранилище онтологий в рамках подсистемы формирования интерактивных документов. Данная подсистема в том числе обеспечивает интерфейс администратора для изменения онтологии стилей.

При запуске подсистемы структуризации онтология стилей извлекается из хранилища, и с помощью специализированной подпрограммы ( «интерпретатор онтологии стилей») превращается в подмножество базы правил редукции.

С учетом этих правил и осуществляется структуризация на первом этапе. Результаты структуризации превращаются в РПС с помощью соответствующей подпрограммы, и на их основе проводится классификация исходного текста с помощью классификационной системы «Confor». Полученная классификация подается на вход подпрограммы структуризации, что позволяет повторить редукцию с большей точностью и получить конечный результат - информационную онтологию, которая в дальнейшем хранится в соответствующей базе.

Подсистема формирования интерактивных документов содержит две основные составляющие: хранилище онтологий и натуральную систему. Дополнительно используется интерпретатор онтологических шаблонов представления, что позволяет модифицировать функциональность натуральной системы. Онтологические шаблоны представления, используемые в работе интерпретатора, хранятся в хранилище рядом с информационными онтологиями.

Дополнительно натуральная система позволяет выполнять трансдисциплинарную интеграцию информации. Для этого используется специализированный модуль индексации («Индексатор»). На вход индексатора подаются как структурированные (информационные онтологии), так и неструктурированные (ЕЯТ) документы, а результаты его работы отображаются с помощью натуральной системы.

При этом каждая из подсистем имеет свои уровни логики, данных и представления. Для подсистемы формирования интерактивных документов уровень данных обеспечивается хранилищем онтологий, а уровни логики и представления – натуральной системой. Подсистема структуризации не предусмотрена для прямого взаимодействия с пользователем и не имеет уровня представления, уровень логики обеспечивается подпрограммой структуризации, а уровень данных – специализированными подпрограммами считывания и записи файлов (в частности, интерпретатором онтологии стилей и подпрограммой формирования РПС).

Такая архитектура позволяет обеспечить максимально эффективное взаимодействие различных частей системы между собой и системы в целом – с пользователем.

## Выводы

Структуризация ЕЯТ позволяет значительно повысить эффективность работы с имеющейся в них информацией. Разработка системы, обеспечивающей автоматизированную структуризацию, дает возможность эффективной работы с очень большими массивами данных, которые невозможно обработать вручную.

Применение онтологического классификатора стилей и двухуровневой схемы обработки (с дополнительной классификацией) позволяет значительно повысить точность выходного результата и уменьшить количество ошибок обработки.

Осуществление трансдисциплинарной интеграции позволяет в дальнейшем повысить эффективность обработки больших массивов разнородных пространственно распределенных документов.

## Литература

[Amit, 2005] Amit Konar. Cognitive Engineering: A Distributed Approach to Machine Intelligence. Series: Advanced Information and Knowledge Processing, Springer, Cham, 2005, 354 p.

[David, 2013] David K. E. Modeling Narrative Discourse: Ph.D. thesis / David K. E. Columbia University, New York City, 2012, 383 p.

[Malyshevsky, 1998] Malyshevsky A. Qualitative models in the theory of complex systems. In: Science. Fizmatlit, Moscow, 1998, 528 p.

[Mayer-Schönberger, 2013] Mayer-Schönberger V, Cukier K. Big Data: A Revolution That Will Transform How We Live, Work, and Think. MA: Houghton Mifflin Harcourt, Boston, 2013, 252 p.

[Басюк, 2017] Басюк Т. М., Досин Д. Г. Онтологічний інжиніринг. Вид-во Львівської політехніки, Львів, 2017, 224 с.

[Величко, 2017] Величко В. Ю., Попова М. А., Приходнюк В. В., Стрижак О. Є. ТОДОС – IT-платформа формування трансдисциплінарних інформаційних середовищ. Системи озброєння і військова техніка, № 1, 2017, С. 10–19. http://nbuv.gov.ua/UJRN/soivt_2017_1_4.

[Гладун, 2004] Гладун В. П., Ващенко Н. Д., Величко В. Ю., Ткаченко Ю. Г. Структуризация и анализ данных в растущих пирамидальных сетях. Syst. Res. Inf. Technol., no. 1, 2004, pp. 82–92. doi: 10.1017/CBO9781107415324.004.

[Палагин, 2012] Палагин А. В., Крывый С. Л., Петренко М. Г. Онтологические методы и средства обработки предметных знаний. Изд-во ВНУ им. В. Даля, Луганск, 2012, 324 с.

[Палагин, 2014] Палагин А. В. Трансдисциплинарность, информатика и развитие современной цивилизации. Вістник НАН України, no. 7, 2014, pp. 25–33.

[Палагин, 2016] Палагин А. В. Онтологическая концепция информатизации научных исследований. Кибернетика и системный анализ. Т. 52, № 1, 2016, с. 3-9.

[Приходнюк, 2018] Приходнюк В. В. Технологічні засоби трансдисциплінарного представлення геопросторової інформації [Текст]: дис. канд. наук.: (05.13.06 – Інформаційні технології). Інст. телеком. і глоб. інформ. Простору, Київ, 2018, 267 с.

[Стрижак, Онтологические аспекты, 2014] Стрижак А. Е. Онтологические аспекты трансдисциплинарной интеграции информационных ресурсов. Открытые информационные и компьютерные интегрированные технологии, №. 65, 2014. С. 211–223.

[Стрижак, Трансдисциплінарна інтеграція, 2014] Стрижак О. Є. Трансдисциплінарна інтеграція інформаційних ресурсів [Текст] : автореф. дис. д-ра техн. наук : 05.13.06. Нац. акад. наук України, Ін-т телекомунікацій і глобал. інформ. простору, Київ, 2014, 547 с

[Стрижак, 2020] Стрижак О. Є. Таксономічні засади наративного дискурсу. Медична інформатика та інженерія. (In print)

[Широков, 2017] Широков В. А. Язык. Информация. Система : Трансдисциплинарность в лингвистике. Palmarium Academic Publishing, Киев, 2017, 270 с.

## Информация об авторах

***Гайко Светлана Ивановна*** *– аспирант, Институт телекоммуникаций и глобального информационного пространства НАН Украины, Киев-186, 03186, Чоколовский бульвар, 13; e-mail: svitgai@i.ua*

*Основные области научных исследований: обработка ЕЯТ, онтологический инжиниринг.*

***Приходнюк Виталий Валерьевич*** *– кандидат технических наук, Национальный центр «Малая академия наук Украины», Киев, 04119, ул. Дегтяревская 38/44, 13; e-mail: Prikhodnyuk_Vitaly@nas.gov.ua*

*Основные области научных исследований: обработка ЕЯТ, онтологический инжиниринг.*

## MEANS OF TRANSDISCIPLINARY REPRESENTATION OF DIFFERENT STYLES INFORMATION RESOURCES

**Svitlana Gaiko, Vitalii Prykhodniuk**

***Abstract****: The article deals with the problem of effective use the heterogeneous information resources (IR). A huge number of natural language documents accumulated on the web remain passive sources of knowledge. To work effectively with such documents, first of all, their structuring is required, which cannot be done manually, taking into account the necessary labor costs. It is also necessary to determine the way of presenting the results of IR structuring, since further possibilities when working with them depend on this. The most important of these capabilities is the integration of different styles IR that*

*describe different subject areas in a single information space. The article proposes a model for the integration of heterogeneous documents that implements the principle of transdisciplinarity. The technology of automated processing of natural language texts (NLT) of various styles and the formation of interactive documents on their basis is described. The mechanism of displaying their structure, which is the narrative discourse, is considered. A mathematical model of this technology, a model of system behavior (in the form of UML diagrams), as well as the architecture of a software tool created on the basis of the described models are presented.*

***Keywords**: transdisciplinarity, ontology, growing pyramidal network (GPN), narrative discourse, functional styles.*

# TABLE OF CONTENTS

*Quality of Experience Modeling of Multimedia On-Line Services*

Zlatinka Kovacheva, Stoyan Poryazov, Emiliya Saranova ..... 3

*Reliable Monte Carlo Methods for Multidimensional Sensitivity Analysis*

Venelin Todorov, Stoyan Poryazov ..... 22

*Устройства умножения одинарной точности на базе FPGA*

Владимир Опанасенко, Сергей Крывый, Станислав Завьялов ..... 35

*К вопросу построения оптимального дерева решений*

Виталий Величко ..... 52

*Средства трансдисциплинарного представления информационных ресурсов разных стилей*

Светлана Гайко, Виталий Приходнюк ..... 78

Table of Contents ..... 100