ISSN 2587-814X (print), ISSN 2587-8158 (online)
Russian version: ISSN 1998–0663 (print), ISSN 2587–8166 (online)

Vol. 14 No 2 – 2020

BUSINESS INFORMATICS

HSE SCIENTIFIC JOURNAL

CONTENTS

Modeling of social and economic systems

Data analysis and intelligence systems

Mathematical methods and algorithms of business informatics

Publisher:
National Research University
Higher School of Economics

Subscription index
in the Rospechat catalog –
80870

The journal is published quarterly

The journal is included
into the list of peer reviewed
scientific editions established
by the Supreme Certification
Commission of the Russian Federation

Address:
28/11, build. 4, Shablovka Street
Moscow 119049, Russia

Tel./fax: +7 (495) 772-9590 *26311
http://bijournal.hse.ru
E-mail: bijournal@hse.ru

Circulation:
English version – 200 copies,
Russian version – 200 copies,
online versions in English and Russian –
open access

Printed in HSE Printing House
3, Kochnovsky Proezd, Moscow,
Russia

ABOUT THE JOURNAL

Business Informatics is a peer reviewed interdisciplinary academic journal published since 2007 by National Research University Higher School of Economics (HSE), Moscow, Russian Federation. The journal is administered by School of Business Informatics. The journal is published quarterly.

The mission of the journal is to develop business informatics as a new field within both information technologies and management. It provides dissemination of latest technical and methodological developments, promotes new competences and provides a framework for discussion in the field of application of modern IT solutions in business, management and economics.

The journal publishes papers in the areas of, but not limited to:

✦ data analysis and intelligence systems

✦ information systems and technologies in business

✦ mathematical methods and algorithms of business informatics

✦ software engineering

✦ internet technologies

✦ business processes modeling and analysis

✦ standardization, certification, quality, innovations

✦ legal aspects of business informatics

✦ decision making and business intelligence

✦ modeling of social and economic systems

✦ information security.

The journal is included into the list of peer reviewed scientific editions established by the Supreme Certification Commission of the Russian Federation.

The journal is included into Web of Science Emerging Sources Citation Index (WoS ESCI) and Russian Science Citation Index on the Web of Science platform (RSCI).

International Standard Serial Number (ISSN): 2587-814X (in English), 1998-0663 (in Russian).

Editor-in-Chief: Dr. Alexey Golosov – President of FORS Development Center, Moscow, Russian Federation.

EDITORIAL BOARD

EDITOR-IN-CHIEF

DEPUTY EDITOR-IN-CHIEF

EDITORIAL BOARD

ABOUT THE HIGHER SCHOOL OF ECONOMICS

Consistently ranked as one of Russia's top universities, the Higher School of Economics (HSE) is a leader in Russian education and one of the preeminent economics and social sciences universities in Eastern Europe and Eurasia.
Having rapidly grown into a well-renowned research university over two decades, HSE sets itself apart with its international presence and cooperation.

Our faculty, researchers, and students represent over 50 countries, and are dedicated to maintaining the highest academic standards. Our newly adopted structural reforms support both HSE's drive to internationalize and the groundbreaking research of our faculty, researchers, and students.

Now a dynamic university with four campuses, HSE is a leader in combining Russian educational traditions with the best international teaching and research practices. HSE offers outstanding educational programs from secondary school to doctoral studies, with top departments and research centers in a number of international fields.

Since 2013, HSE has been a member of the 5-100 Russian Academic Excellence Project, a highly selective government program aimed at boosting the international competitiveness of Russian universities.

ABOUT THE SCHOOL OF BUSINESS INFORMATICS

The School of Business Informatics is one of the leading divisions of HSE's Faculty of Business and Management. The School offers students diverse courses taught by full-time HSE instructors and invited business practitioners. Students are also given the opportunity to carry out fundamental and applied projects at various academic centers and laboratories.

Within the undergraduate program, students participate each year in different case-competitions (PWC, E&Y, Deloitte, Cisco, Google, CIMA, Microsoft Imagine CUP, IBM Smarter Planet, GMC etc.) and some of them are usually as being best students by IBM, Microsoft, SAP, etc. Students also have an opportunity to participate in exchange programs with the University of Passau, the University of Munster, the University of Business and Economics in Vienna, the Seoul National University of Science and Technology, the Radbound University Nijmegen and various summer schools (Hong Kong, Israel etc.). Graduates successfully continue their studies in Russia and abroad, start their own businesses and are employed in high-skilled positions in IT companies.

There are four graduate programs provided by the School:

- Business Informatics
- E-Business;
- Information Security Management;
- Big Data Systems.

The School's activities are aimed at achieving greater integration into the global education and research community. A member of the European Research Center for Information Systems (ERCIS), the School cooperates with leading universities and research institutions around the world through academic exchange programs and participation in international educational and research projects.

DOI: 10.17323/2587-814X.2020.2.7.20

Transfer learning and domain adaptation based on modeling of socio-economic systems

Oleg D. Kazakov
E-mail: kod8383@mail.ru

Olga V. Mikheenko
E-mail: miheenkoov@mail.ru

Bryansk State Technological University of Engineering
Address: 3, Stanke Dimitrov Avenue, Bryansk 241037, Russia

Abstract

This article deals with the application of transfer learning methods and domain adaptation in a recurrent neural network based on the long short-term memory architecture (LSTM) to improve the efficiency of management decisions and state economic policy. Review of existing approaches in this area allows us to draw a conclusion about the need to solve a number of practical issues of improving the quality of predictive analytics for preparing forecasts of the development of socio-economic systems. In particular, in the context of applying machine learning algorithms, one of the problems is the limited number of marked data. The authors have implemented training of the original recurrent neural network on synthetic data obtained as a result of simulation, followed by transfer training and domain adaptation. To achieve this goal, a simulation model was developed by combining notations of system dynamics with agent-based modeling in the AnyLogic system, which allows us to investigate the influence of a combination of factors on the key parameters of the efficiency of the socio-economic system. The original LSTM training was realized with the help of TensorFlow, an open source software library for machine learning. The suggested approach makes it possible to expand the possibilities of complex application of simulation methods for building a neural network in order to justify the parameters of the development of the socio-economic system and allows us to get information about its future state.

Key words: transfer learning; domain adaptation, simulation modeling; decision support systems; socio-economic development of regions.

Citation: Kazakov O.D., Mikheenko O.V. (2020) Transfer learning and domain adaptation based on modeling of socio-economic systems. *Business Informatics*, vol. 14, no 2, pp. 7–20.
DOI: 10.17323/2587-814X.2020.2.7.20

Introduction

Management of the development of social and economic systems is mainly based on documents containing the planned values of indicators on a particular topic (strategy, concept, forecast, etc.). To date, the management of the region is carried out through monitoring by adjusting the planned values in accordance with those actually achieved [1]. This means that the basis for future development for the most part lies in the indicators of past periods obtained with a significant delay, if we take into account the real situation with the publication of official statistics. In this regard, the development of tools to justify the values of forecast economic parameters, which allows us to achieve planned control figures with a high degree of reliability, is an important scientific task. This task ultimately acts as an objective condition for the implementation of an effective economic policy.

The basis of the whole set of methods of socio-economic forecasting is traditionally made up of statistical methods used to build appropriate models of time series [2, 3]. Among the most common methods for analyzing time series are the following [4]: regression forecasting models (multiple and non-linear regression), exponential smoothing (ES) models, maximum similarity sampling model (MMSP), the Markov chains model, the Markov chains model on classification and regression trees (CART), a model based on the genetic algorithm (GA), a model on support vectors (SVM). The widest and most applicable of the classes of models are autoregressive forecasting models (ARIMAX, GARCH, ARDLM).

Recently, deep machine learning methods, whose quality metrics are much better than classical methods, have proven their effectiveness. However, the use of such models requires a huge amount of tagged data, which in real conditions it is not always possible to obtain. At the same time, when forecasting most of the indicators characterizing socio-economic systems and processes, statistical data are used for one decade and, in the best case, by month. In other words, there are only a hundred marked entries at the input. The problem could be solved in one way or another if teaching without a teacher was applied, which, unfortunately, at this stage of development of serial computer systems cannot be implemented in practice.

To solve this problem, we proposed to use a recurrent neural network built on the architecture of long short-term memory (LSTM) and trained on synthetic data obtained as a result of simulation with subsequent transfer learning (transfer learning) and domain adaptation (domain adaptation). By having real statistics for several decades, this will allow us, with a high degree of accuracy, to predict the values of economic parameters taking into account modern development vectors. Decision support systems based on these algorithms make it possible to most accurately justify economic plans and forecasts for the development of territories and ensure the achievement of strategic development guidelines.

1. Methods

1.1. Transfer learning and domain adaptation at LSTM

The main idea of transfer training is to solve the problem on the basis of “ready data” obtained as a result of solving similar problems. This means that you can first train a neural network on a large amount of data, and subsequently retrain it on a specific target set. In this regard, there are two main advantages of using transfer training [5]:

✦ a significant reduction in time and costs in the context of using the appropriate infrastructure for training, by training only a certain part of the final model;

- increasing the efficiency of the final model through the use of models trained on available data.

The results of this study are closely related to the second of these advantages, since in the predictive analysis of socio-economic systems this is a determining factor.

As the available data, synthetic data obtained as a result of simulation were used. Simulation is an experimental way to study reality using a computer model [6]. In simulation models, real economic processes are described as if they were actually happening [7]. Thus, simulation models can be used to study real socio-economic systems under the condition that economic objects and processes are replaced by a set of mathematical dependencies that determine what state the system will go from initially set [8].

The weights from the model trained on synthetic data obtained as a result of simulation are transferred to a new model. For this, the authors used the TensorFlow open machine learning software library.

The methods of transfer training and domain adaptation, as a rule, depend on machine learning algorithms used to solve the tasks [9]. One of the most effective tools for predictive analytics of socio-economic systems is recurrent neural networks with long short-term memory (LSTM networks). In particular, models based on the LSTM architecture are very effective for forecasting the time series — one of the most common tasks in managing socio-economic systems [10]. It should be noted that this efficiency does not decrease when predicting several steps.

The basic architecture of the recurrence network, developed back in the 1980s, is built from nodes, each of which is connected to all other nodes. For training with a teacher with discrete time, data is supplied to the input nodes at each next step. In this case, other nodes (output and hidden) complete their activation and the output signals are prepared for transmission to neurons of the next level [11]. Thus, a recurrent network with long-term memory allows use of information received in the past to solve current problems. In particular, it makes it possible to predict the values of the time series, since it does not use the activation function inside its recurrent components, and the stored value does not blur in time (*Figure 1*) [12, 13].

The LSTM module has five main components that allow it to simulate both long-term and short-term data [13]:

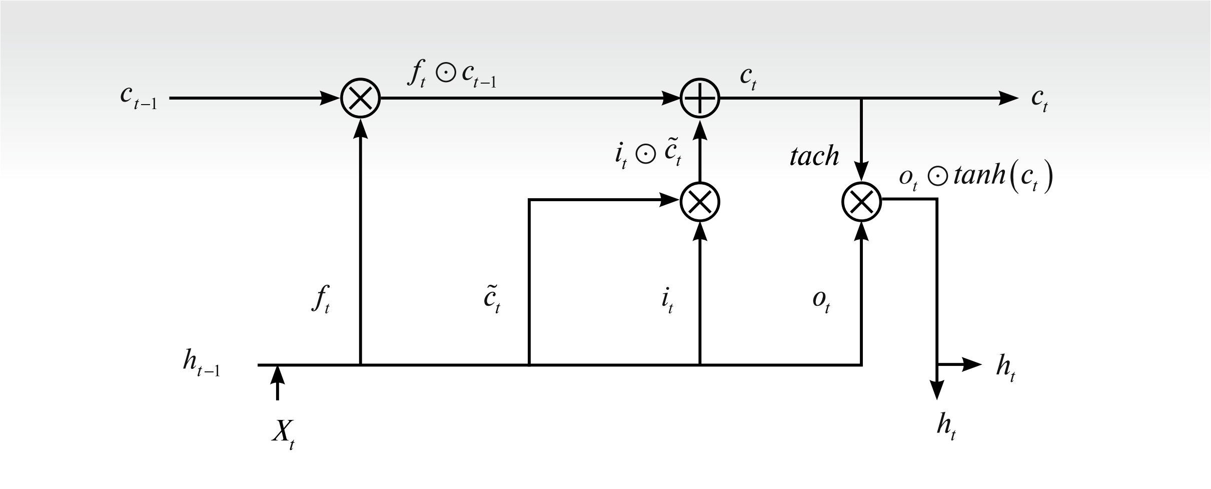


Fig. 1. Architecture of LSTM [12, 13]

$$\begin{cases} f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_t\right) \\ i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right) \\ o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right) \\ \tilde{c}_t = tanh\left(W_c\left[h_{t-1}, x_t\right] + b_c\right) \\ c_t = f_t \odot c_{t-1} + i_t \odot c_t \\ h_t = o_t \odot tanh\left(c_t\right) \end{cases}, \qquad (1)$$

where c_t – “state of the cell”, representing its internal memory, which stores both short-term and long-term information;

h_t – “hidden state”: such information about the output state, which is calculated by the current input, the previous hidden state and the current input of the cell, which will be used to predict one or another time series. The latent state may decide to extract short-term or long-term or both types of information from stored in (c_t);

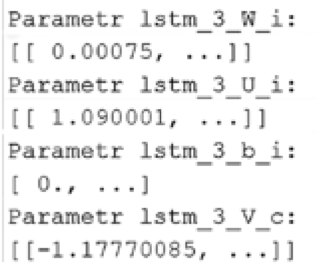
i_t – “entrance gate”: determine the amount of information coming from the current input in (c_t);

f_t – “transitional gate”: determine the amount of information flowing from the current and previous (c_{t-1}) inputs to the current (c_t);

o_t – “exit gate”: determine the amount of information falling from the current (c_t) into a hidden state.

Suppose there is a well-functioning model for predictive time series analytics “Kazakov_LSTM.h5” (the process of its training is presented in the next section of the article). Then, to view the parameters of this model, you can use the following instructions (*Figure 2*).

Thus, we obtain the following conclusion:



```
Parametr lstm_3_W_i:
[[ 0.00075, ...]]
Parametr lstm_3_U_i:
[[ 1.090001, ...]]
Parametr lstm_3_b_i:
[ 0., ...]
Parametr lstm_3_V_c:
[[-1.17770085, ...]]
```

(2)

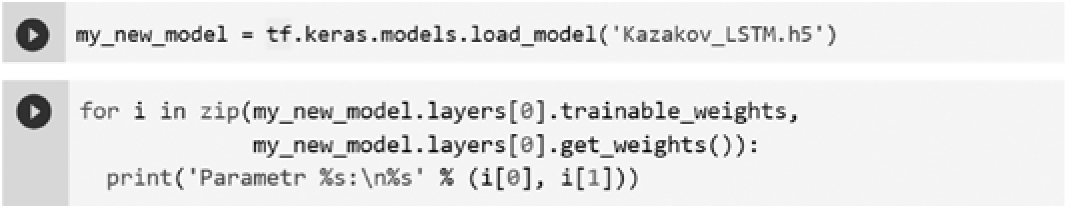
where *W*-matrices – matrices that transform the input data;

U-matrices – matrices that transform the previous hidden state into another internal value;

b-vectors – offset for each block;

V – a vector that determines what values to derive from the new internal state.

The concept of domain adaptation is closely related to transfer training. The essence of this adaptation is to train the model on data from the source domain so that it shows comparable quality on the target domain [14]. The source domain can be synthetic data that can simply be generated by running the corresponding simulation model, and the target domain is a time series that reflects the dynamics of certain key indicators of the socio-economic system. Then the task of domain adaptation is to train the model on synthetic data, which will work well with real objects.



```
my_new_model = tf.keras.models.load_model('Kazakov_LSTM.h5')
```

```
for i in zip(my_new_model.layers[0].trainable_weights,
             my_new_model.layers[0].get_weights()):
  print('Parametr %s:\n%s' % (i[0], i[1]))
```

Fig. 2. Listing “LSTM parameter output”

The stage of domain adaptation is reduced to freezing weights in the "Kazakov_LSTM.h5" model in their previously prepared state. Domain adaptation weights are trained on the target data set. For this purpose, in the model after LSTM we add fully connected (dense) layers.

1.2. System-dynamic modeling of indicators of innovative development of socio-economic systems

In order to form a data set within the source domain, we will build a system-dynamic model that allows us to determine the parameters of the socio-economic system, in particular, to evaluate the values of the indicators of innovative development of the regions. The main document setting strategic guidelines for state policy in the field of innovative development in order to counter modern global challenges and threats is the Innovation Development Strategy of the Russian Federation for the period until 2020 [15]. The strategy therein determines the long-term development priorities of all subjects of innovation, and also sets a number of target indicators, which, in accordance with the installation of the Government of the country, should be taken into account when developing concepts and programs for the socio-economic development of Russia and its regions.

The strategy defines the values of target indicators for 2020. At the same time, 2010 is fixed as the base year, and 2013 and 2016 are intermediate control points. An analysis of the actual values of most of the target indicators for 2016 revealed a general tendency to lag behind the planned level.

Since statistical services prepare analytical data for the reporting period with a significant time lag, a serious problem is the fact that the state authorities responsible for the implementation of the Strategy are trying to develop managerial decisions, focusing on irrelevant performance results. It is extremely difficult to call such a process effective management.

In our opinion, the reverse movement will be the most effective approach when the value of a specific target indicator for a certain date is differentiated among the subjects of the federation and communicated to the regional authorities in advance, in the form of recommended forecast values. In this case, local government services will become direct participants in the process of implementing national strategic initiatives, including taking into account certain responsibilities for failure to achieve targets. In addition, it will be possible to manage on the basis of an up-to-date map reflecting the innovative development of the long-term objectives of the Strategy by regions.

The main components of the system model, which allows us to determine the innovative development of the region in accordance with the state strategy, are strategic tasks in key areas. The relationship between the final indicator of innovative development and these components (subindexes) can be described as follows:

$$I = \sum_{j=1}^{m} w_j \cdot I_j, \tag{3}$$

where I_j – j-th subindex value;

m – number of subindexes;

w_j – weight factor of the j-th subindex.

Subindexes are summary indicators reflecting the formation of primary indicators as part of the solution of a specific strategic task in priority areas of innovation. The number of primary indicators in directions varies from 2 to 12, and each of them can be considered as an independent complex system-dynamic model. Consider the model for the formation of a first-order private indicator "Inventive Activity Coefficient," which is determined in the framework of the priority strategic task "Innovative Business." Models of other indicators can be formed in a similar way and are not presented in the framework of this article due to the significant scale of the study.

The system-dynamic model of the level of inventive activity is presented in *Figure 3.* The model is based on determining the ratio of the number of patent applications filed by domestic inventors to the total population. The number of patent applications developed and filed depends on the number of organizations engaged in research and development, the number of personnel involved in research and development, as well as the amount of internal expenses of organizations on research and development.

The corresponding mathematical model can be represented as follows:

$$\frac{d\left(\text{Inventions\_developed}\right)}{dt} = \text{- patent}$$

$$\frac{d\left(\text{Useful\_Models \_Developed}\right)}{dt} = \\ = \text{- Useful\_Models\_Filed}$$

$$\frac{d\left(\text{patent}\right)}{dt} = \text{Inventions\_submitted} + \\ + \text{Utility\_models\_submitted}$$

$$\frac{d\left(\text{population}\right)}{dt} = \text{Birth - Death} + \\ + \text{Migration - Emigration} \qquad (4)$$

$$\text{Inventions}\left(\text{Utility\_models}\right)\text{\_developed} = \\ = \text{corp\_research} \times \text{person\_research} \times \\ \times \text{Internal\_R \& D costs}$$

$$\text{Death} = \text{Death\_trudospos\_zhazhra} + \\ + \text{Death\_ infantile} + \text{other}$$

$$\text{Migration} = \text{Refugees} + \text{Temporary shelter} + \\ + \text{Forced migrants} + \\ + \text{Arriving coeff\_inv\_activity} = \\ = \frac{\text{patent}}{\text{population}} \times 10\,000.$$

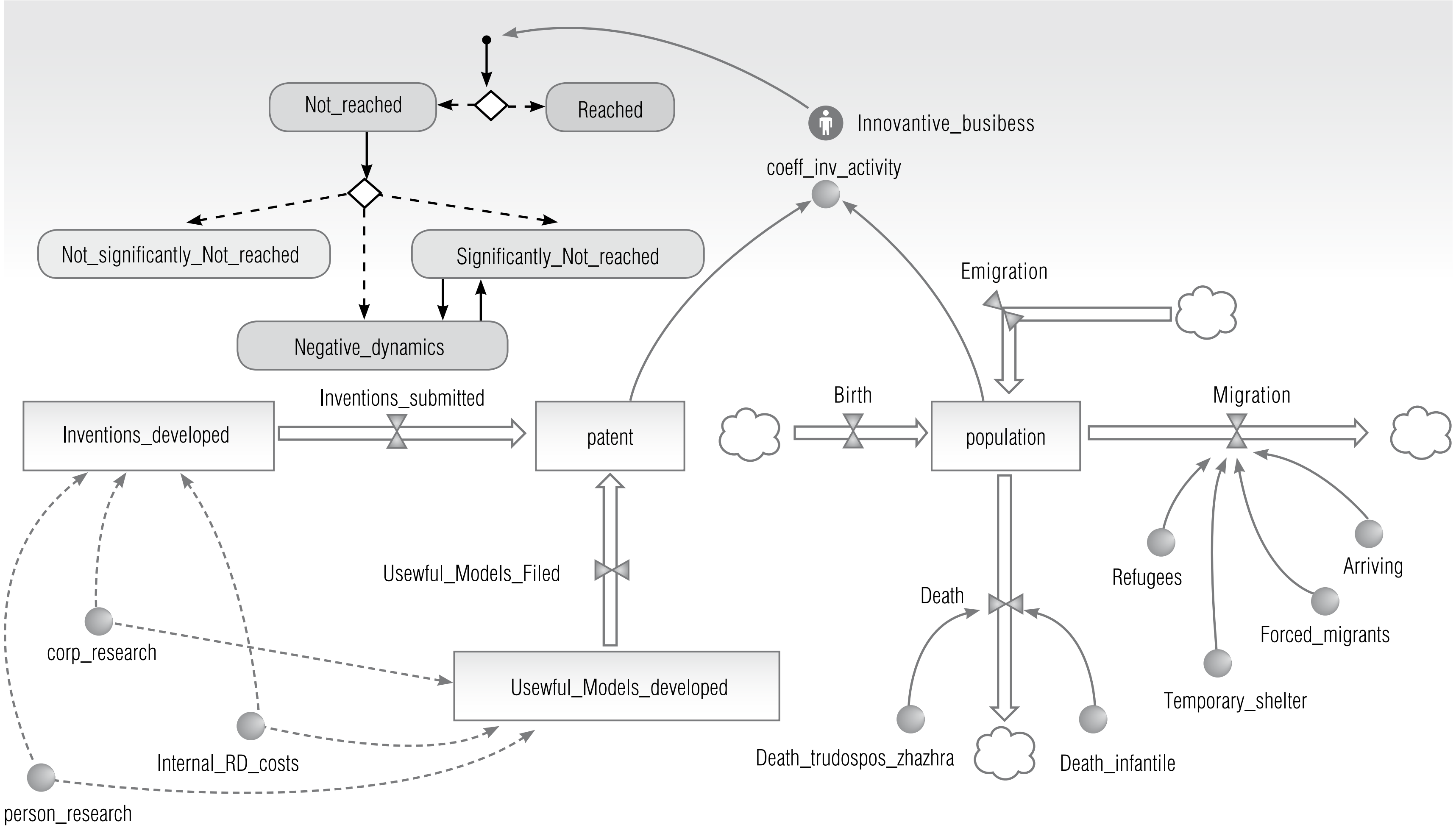
At the first consideration, the question arises of the appropriateness of using simulation to assess the level of inventive activity, since each of its components can be predicted (for example, within the framework of ARIMA or GARCH models that have proven themselves in the field of forecasting demographic and socio-economic indicators). In fact, the dependence of inventive activity on many indicators is stochastic; moreover, in practice, for most of them it is not possible to collect a sufficiently large set of values. Therefore, simulation in this context is considered as a way to build a model of a socio-economic system that describes the complex behavior of objects and processes associated with innovation management at the regional level. This model can be implemented any number of times. In this case, the results will be due to the random nature of the processes [16]. Using these results, one can obtain stable synthetic statistics on the level of inventive activity, which is subsequently used to train the neural network.

2. Experiment

2.1. Synthetic data generation using a system-dynamic model

Simulation modeling was implemented using a set of mathematical tools and AnyLogic special software, which allowed for targeted modeling in the "simulation" mode of the indicator under study, as well as optimization of some of its parameters [17]. In accordance with the results of a study conducted by a group of scientists from Kazan Technical University [18], the reliability of the AnyLogic system was found to be satisfactory, and in the ranking of similar software this system is among the top three.

The configuration settings of the model that graphically describes the user-posed problem in terms of the AnyLogic language are set using experiments. Discrete event modeling implements the possibility of approximating real processes by discrete events that consider the most important moments of the life of the simulated system [19].



coeff_inv_activity	–	coefficient of inventive activity, units;
patent	–	the number of domestic patent applications filed for inventions, units;
population	–	population of the territory, thousand people;
corp_research	–	number of organizations performing research and development, units;
person_research	–	the number of personnel engaged in research and development, people

Fig. 3. System–dynamic model of the level of inventive activity

An experiment "Variation of parameters" was carried out in the AnyLogic system, the essence of which was the repeated launch of the constructed simulation model. For the experiment, a confidence probability of 0.95 and an accuracy of 0.01 were determined. The number of model runs calculated by the Laplace function was 9604 [20]. Varying different parameter values, the model produced a label value ranging from 1.4725 to 2.1105. The mathematical expectation was 1.8114, and the dispersion of values relative to the mathematical expectation was 0.1539, which is acceptable and allows us to conclude that the proposed simulation model was successfully validated. The results of the experiment are presented in *Table 1.*

Table 2 presents the synthetic data obtained as a result of the experiment "Variation of parameters" in the AnyLogic system and used to train the primary neural network.

The initial data set contains five functions, the change in time of which is presented in *Figure 4.*

The figure shows that all time series have the property of seasonality, but we will not take this factor into account explicitly in further training of the original network.

2.2. Source LSTM training and learning transfer

As noted above, the training of the initial LSTM is carried out on synthetic data obtained as a result of simulation. To do this, use the TensorFlow open source machine learning software library, which provides a good helper application programming interface (RNN API) for implementing predictive time series models.

First of all, to complete the training process, we will load the synthetic data and standardize the data set using the mean () and std () functions.

Further, the task is reduced to predicting a multidimensional time series based on some provided history. We will create training and validation data and perform direct training of the original LSTM.

The multivariate_data function performs the window management task. It selects past observations based on a given step size (*Figure 5*). Next, we fix the weights of a pre-trained neural network (*Figure 6*). Then we create a composite neural network based on "Kazakov.h5" and compile it (*Figure 7*).

To select the best neural network hyperparameters, the Keras Tuner optimizer developed by the Google team and included in the Keras open library was used. RandomSearch was defined as

Table 1.

Results of the "Variation of Parameters" experiment of a system-dynamic model of the level of inventive activity in the AnyLogic system

Name	population	patent	corp_research	person_ research	coeff_inv_activity
Parameter Change	[135600…154235]	[21627…30732]	[3317…4384]	[672493…932115]	[1.4725…2.1105]
Expected value	144705.3752	26204.7906	3782.2631	778989.9941	1.8114
Dispersion	6347086.0729	4467795.1931	46155.5052	3873708843	0.0237
Standard deviation	2519.3424	2113.7160	214.8383	62239.1263	0.1539

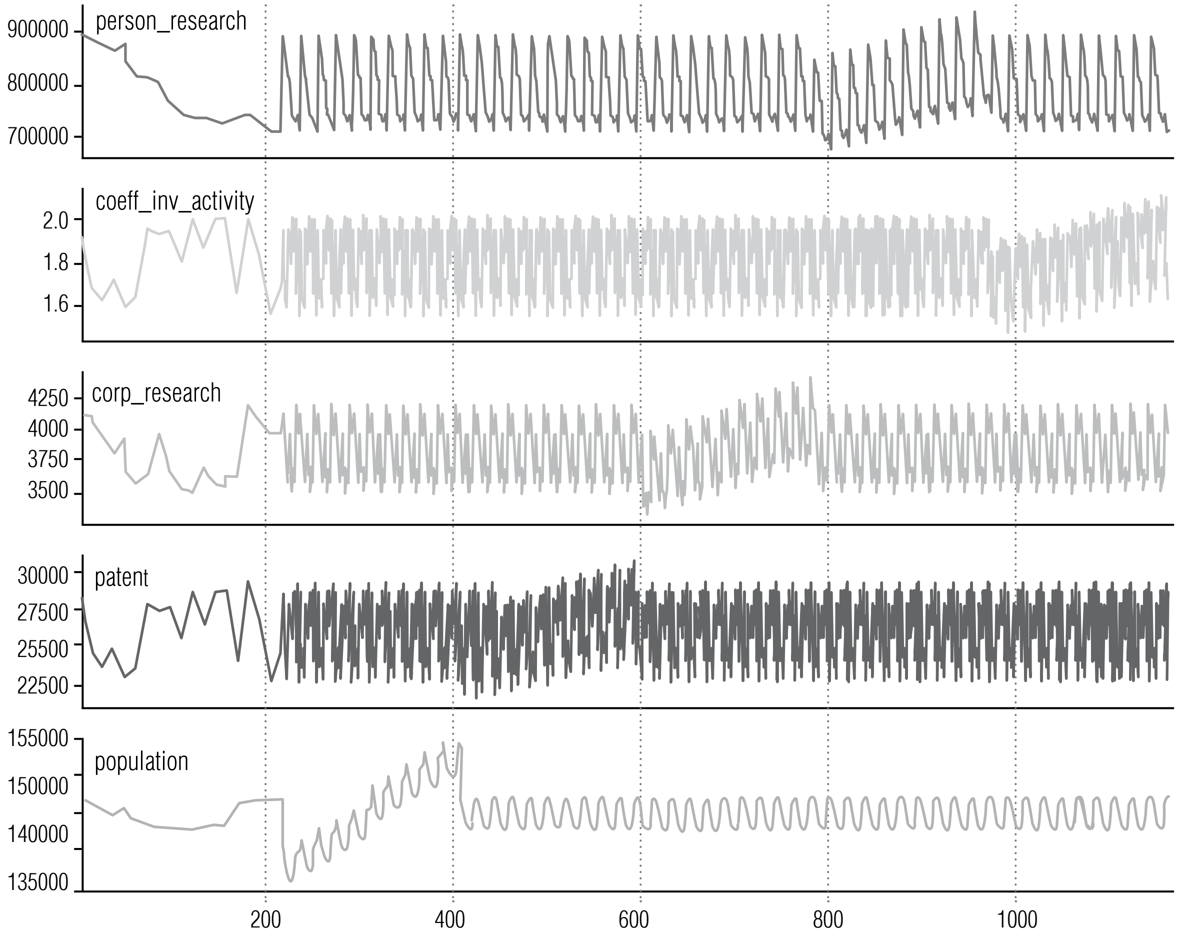
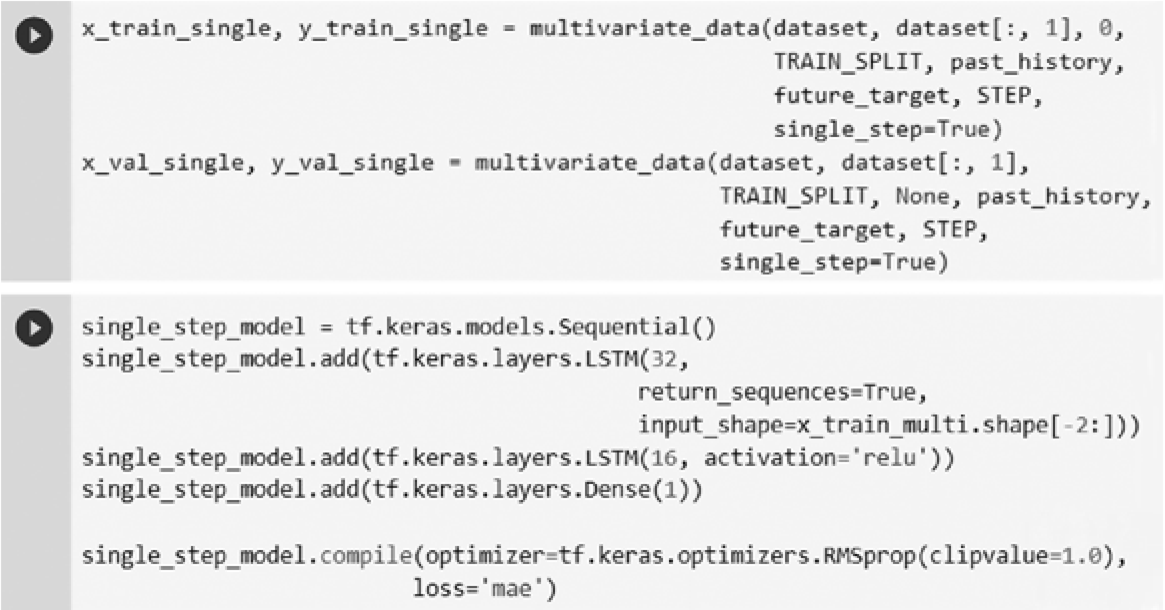


Fig. 4. The change in time of the original functions



```
x_train_single, y_train_single = multivariate_data(dataset, dataset[:, 1], 0,
                                                   TRAIN_SPLIT, past_history,
                                                   future_target, STEP,
                                                   single_step=True)
x_val_single, y_val_single = multivariate_data(dataset, dataset[:, 1],
                                               TRAIN_SPLIT, None, past_history,
                                               future_target, STEP,
                                               single_step=True)
```

```
single_step_model = tf.keras.models.Sequential()
single_step_model.add(tf.keras.layers.LSTM(32,
                                           return_sequences=True,
                                           input_shape=x_train_multi.shape[-2:]))
single_step_model.add(tf.keras.layers.LSTM(16, activation='relu'))
single_step_model.add(tf.keras.layers.Dense(1))

single_step_model.compile(optimizer=tf.keras.optimizers.RMSprop(clipvalue=1.0),
                          loss='mae')
```

Fig. 5. Listing “Training the original LSTM”

Table 2.

Synthetic data for training the original neural network

No	Signs				Mark
	population	patent	corp_research	person_ research	coeff_inv_activity
1	146890	28688	4099	887729	1.950000
2	146841	28362	4098	887553	1.931477
3	146792	28036	4097	887377	1.909913
4	146743	27710	4096	887201	1.888335
5	146694	27384	4095	887025	1.886743
…	…	…	…	…	…
9601	143267	24072	3604	732274	1.732500
9602	146545	29269	4175	738857	2.100000
9603	146804	26795	4032	722291	1.921500
9604	146880	22765	3944	707887	1.627500

the main type of Keras Tuner. Listing the best model with Keras Tuner is as follows (*Figure 8*).

To create a neural network with enumeration of the basic values of hyperparameters, the following function was used (*Figure 9*).

For two fully connected layers, Keras Tuner defined relu, the rectifier, and adam, the stochastic gradient descent method, based on an adaptive estimation of first and second order moments, as an activation function. The root mean square error (mse) is presented as a loss function, and the mean absolute error (mae) is used as a quality metric. In the last era of training, these parameters took the values of 0.3995 and 0.1739, respectively.

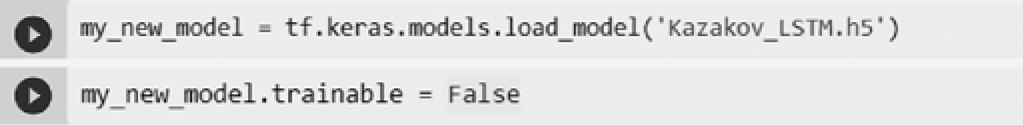
Thus, two fully connected layers were added for the implementation of domain adaptation based on actual data obtained from official statistics and presented in *Table 3*.

It is assumed that the prediction of the dynamics of inventive activity will be carried out by one step, so one neuron will remain at the output of the last layer of the network.

3. Discussion of the results

The developed simulation model of the dynamics of inventive activity allows you to create a potentially unlimited number of records for training the source network. The studied methods of transfer training and domain adaptation in LSTM allowed use of the pre-trained source network in the new mixed architecture. Thus, despite the available critically small set of evidence for training the neural network, it is possible to forecast economic indicators.

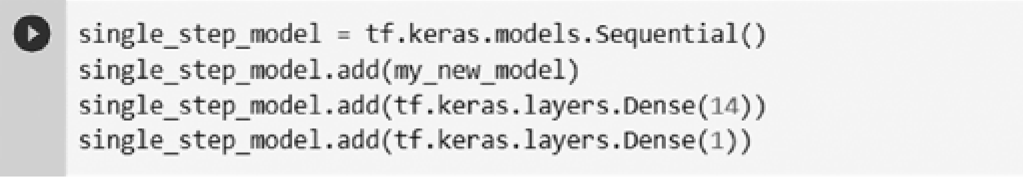
Using a trained neural network, we visualize the predicted values of the innovation activity coefficient for 2012 (based on data for the validation sample) and for 2018 (based on data for the test sample) (*Figure 10*).



```
my_new_model = tf.keras.models.load_model('Kazakov_LSTM.h5')
```

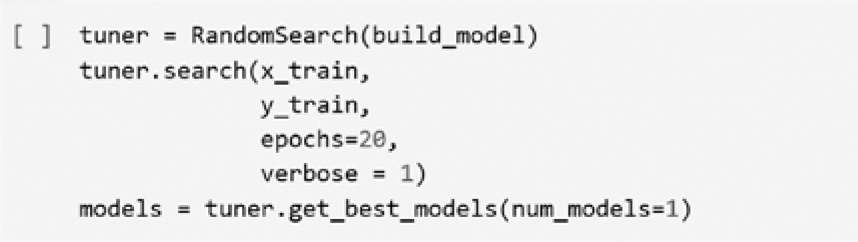
```
my_new_model.trainable = False
```

Fig. 6. Listing “Assigning weights from a pre–trained neural network”



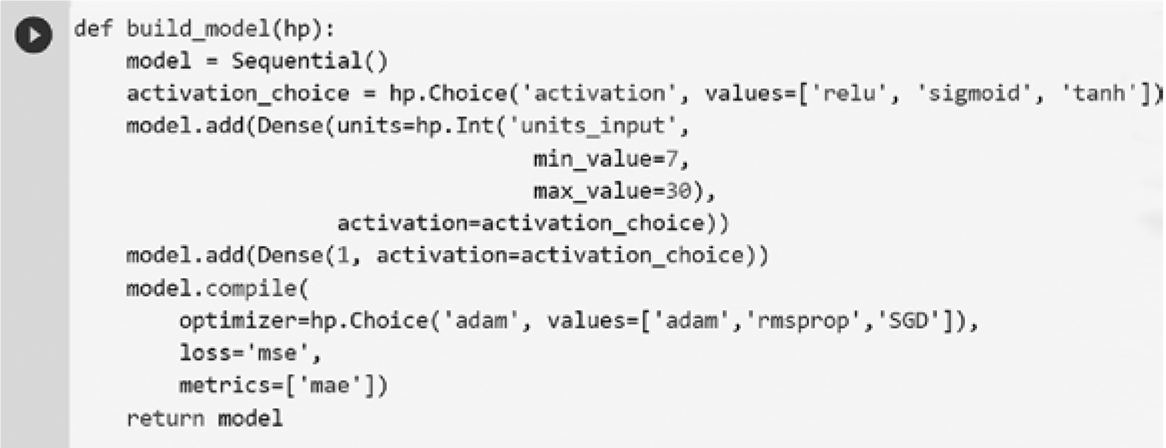
```
single_step_model = tf.keras.models.Sequential()
single_step_model.add(my_new_model)
single_step_model.add(tf.keras.layers.Dense(14))
single_step_model.add(tf.keras.layers.Dense(1))
```

Fig. 7. Listing “Creating the architecture of a composite neural network”



```
[ ]  tuner = RandomSearch(build_model)
     tuner.search(x_train,
                  y_train,
                  epochs=20,
                  verbose = 1)
     models = tuner.get_best_models(num_models=1)
```

Fig. 8. Listing “Selection of neural network hyperparameters”



```
def build_model(hp):
    model = Sequential()
    activation_choice = hp.Choice('activation', values=['relu', 'sigmoid', 'tanh'])
    model.add(Dense(units=hp.Int('units_input',
                                 min_value=7,
                                 max_value=30),
                    activation=activation_choice))
    model.add(Dense(1, activation=activation_choice))
    model.compile(
        optimizer=hp.Choice('adam', values=['adam','rmsprop','SGD']),
        loss='mse',
        metrics=['mae'])
    return model
```

Fig. 9. Listing “Function for selecting neural network hyperparameters”

The data obtained for 2012 showed the value of the mark 1.91 with a value of 2.00 actually recorded during this period (Figure 10a). The inventive activity coefficient determined by the network for 2018 is 1.73 against the actual 1.70 (Figure 10b). Thus, the deviation was 4.5% in 2012 and 1.8% in 2018, respectively. The results can be considered satisfactory, which allows us to broadcast this method in the future.

Conclusion

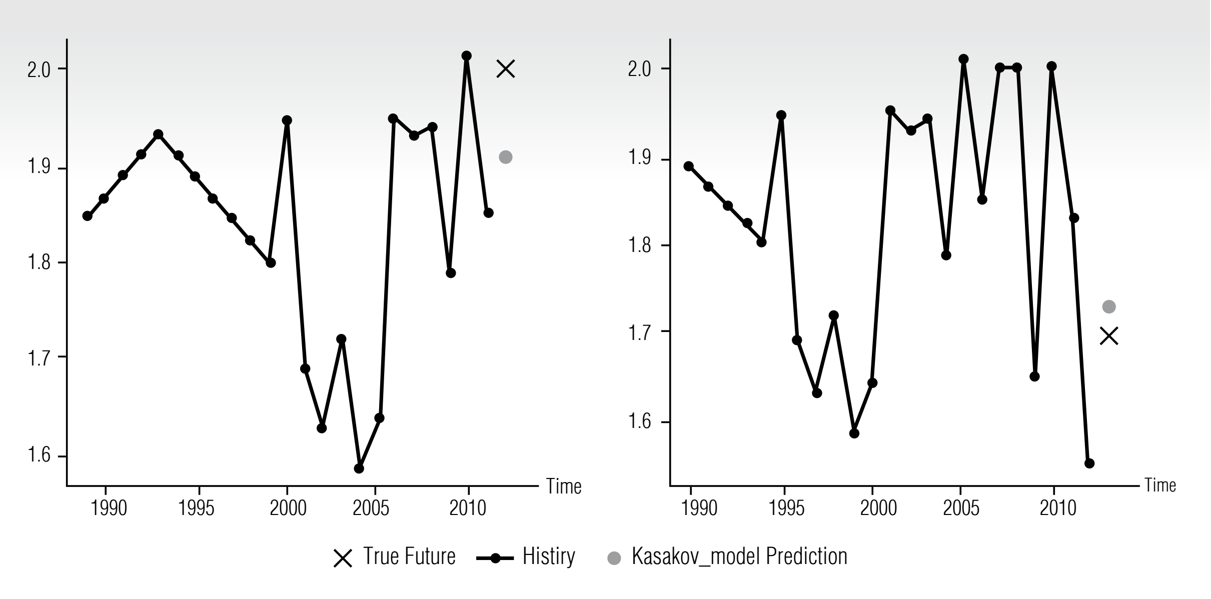
The approach presented in the study, based on the construction of a system-dynamic model and a recurrent neural network, can be adapted

Table 3.

Evidence for implementing domain Adaptation

No	Year	Signs				Mark
		population	patent	corp_research	person_ research	coeff_inv_activity
0	2001	146304	24777.0	4037	885568	1.69
1	2002	145649	23712.0	3906	870878	1.63
2	2003	144964	24969.0	3797	858470	1.72
…	…	…	…	…	…	…
15	2016	146804	26795.0	4032	722291	1.83
16	2017	146880	22765.0	3944	707887	1.55
17	2018	146781	24952.8	3944	707887	1.70

Compiled on the basis of the official website of the Federal State Statistics Service (https://www.gks.ru/)



a) 2012

b) 2018

Fig. 10. Determining the value of the target variable using a trained neural network

to other socio-economic systems and processes in terms of solving problems of predictive analytics. The author's approach to the training and use of LSTM networks in socio-economic systems will significantly increase the effectiveness of management decisions. The undoubted advantage of using this technique, in our opinion, is the possibility of early determination of trends in processes, even under conditions of a limited data set.

The proposed approach can become a universal tool for predictive analytics LSTM, since the studied transfer training and domain adaptation techniques in LSTM allowed using the source network trained on synthetic data and predicting the value of the target variable with a high degree of accuracy. The practical significance of the study is to expand the capabilities of the integrated application of simulation methods for building a neural network. At the same time, the approach we developed can be used by state authorities to justify the development parameters of the socio-economic system and allows us to obtain information about its future status. ■

Acknowledgments

The study was carried out with the financial support of the Russian Federal Property Fund in the framework of scientific project No. 18-41-320003 “Mathematical modeling of the socio-economic development of the region in decision support systems using adaptive methods of machine learning and simulation in the face of uncertainty.”

References

1. Novoselov A.S., ed. (2014) *Regional and municipal management of socio-economic development in the Siberian Federal district.* Novosibirsk: IEIE SB RAS (in Russian).
2. Kantorovich G.G. (2002) Time series analysis. *Economic Journal*, no 1, pp. 87–110 (in Russian).
3. Anderson T. (1976) *Statistical analysis of time series.* Moscow: Mir (in Russian).
4. Community of IT specialists (2013) *Overview of time series forecasting models.* Available at: https://habr.com/ru/post/180409/ (accessed: 15 March 2020) (in Russian).
5. Pan S.J., Yang Q. (2010) A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no 10, pp. 1345–1359. DOI: 10.1109/TKDE.2009.191.
6. Tsaregorodtsev E.I., Barkalova T.G. (2017) Simulation modeling in forecasting of socio-economic systems. *Herald of TISBI*, no 3, pp. 126–134 (in Russian).
7. Mankaev N.V. (2019) Research and modeling of the process of socio-economic systems management. *Soft Measurements and Computing*, no 1 (14), pp. 21–30 (in Russian).
8. Zvyagin L.S. (2015) Practical methods of modeling economic systems. Proceedings of the *IV International Scientific Conference on Problems of the Modern Economy. Chelyabinsk, 20–23 February 2015*, pp. 14–19 (in Russian).
9. Guo H., Zhu H., Guo Z., Zhang X., Wu X., Su Z. (2009) Domain adaptation with latent semantic association for named entity recognition. Proceedings of the *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL – 2009). Boulder, Colorado, USA, 31 May – 5 June 2009*, pp. 281–289. DOI: 10.3115/1620754.1620795.
10. Cortes C., Mohri M. (2014) Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, no 519, pp. 103–126. DOI: 10.1016/j.tcs.2013.09.027.
11. Gafarov F.M., Galimyanov A.F. (2018) *Artificial neural networks and applications.* Kazan: Kazan University (in Russian).
12. Olah C. (2015) *Understanding LSTM networks. GITHUB blog.* Available at: https://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed 12 March 2020).

13. Ganegedara T. (2018) *Stock market predictions with LSTM in Python*. GITHUB blog. Available at: https://www.datacamp.com/community/tutorials/lstm-python-stock-market (accessed 12 March 2020).

14. Kondrashova D.A., Nasyrov R.V. (2019) Comparison of the effectiveness of automatic text classification methods. Proceedings of the *VII All-Russian Scientific Conference on Information Technologies of Intellectual Decision Support. Ufa, 28–30 May 2019*, pp. 146–149 (in Russian).

15. Decree of the Government of the Russian Federation No 2227-R of 8 December 2011. *About the strategy of innovative development of the Russian Federation for the period up to 2020.* Available at: https://www.garant.ru/products/ipo/prime/doc/70006124/#review (accessed: 27 November 2019) (in Russian).

16. *Creating the Monte Carlo experiment*. Available at: https://studme.org/286158/informatika/sozdanie_eksperimenta_monte_karlo (accessed: 27 November 2019) (in Russian).

17. Zvyagin L.S. (2016) Key aspects of complex systems simulation. *Young Scientist*, no 12, pp. 19–23 (in Russian).

18. Drovyannikov V.I., Khaimovich I.N. (2015) Simulation of social cluster management in the AnyLogic system. *Fundamental Study*, no 8–2, pp. 361–366 (in Russian).

19. Yakimov I.M., Kirpichnikov A.P., Isaeva Yu.G., Alyautdinova G.R. (2015) Comparison of simulation results for probabilistic objects in the systems: AnyLogic, Arena, Bizagi modeler, GPSS W. *Bulletin of the Technological University*, vol. 18, no 16, pp. 260–264 (in Russian).

20. Géron A. (2017) *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. Sebastopol, CA: O'Reilly Media.

About the authors

Oleg D. Kazakov

Cand. Sci. (Econ.), Associate Professor;

Head of Department of Information Technology,
Bryansk State Technological University of Engineering,
3, Stanke Dimitrov Avenue, Bryansk 241037, Russia;

E-mail: kod8383@mail.ru;

ORCID: 0000-0001-9665-8138

Olga V. Mikheenko

Cand. Sci. (Econ.);

Associate Professor, Department of Public Administration, Economic and Information Security,
Bryansk State Technological University of Engineering,
3, Stanke Dimitrov Avenue, Bryansk 241037, Russia;

E-mail: miheenkoov@mail.ru;

ORCID: 0000-0003-0917-8406

DOI: 10.17323/2587-814X.2020.2.21.35

Modeling and optimization of plans for railway freight transport performed by a transport operator

Fedor A. Belousov
E-mail: sky_tt@list.ru

Ivan V. Nevolin
E-mail: i.nevolin@cemi.rssi.ru

Nerses K. Khachatryan
E-mail: nerses@cemi.rssi.ru

Central Economics and Mathematics Institute, Russian Academy of Sciences
Address: 47, Nakhimovsky Prospect, Moscow 117418, Russia

Abstract

This paper offers an approach for solving a problem that arises for railway transport operators. The task is to manage the fleet of freight railcars optimally in terms of profit maximization. The source data for the transport operator is a list of requests received from customers, as well as the location of railcars at the beginning of the planning period. The request formed by each customer consists of departure station, destination station, name and volume of cargo that the customer would like to transport. The request also contains the rate that the customer has to pay to the transport operator for each loaded wagon transported. Planning is carried out for a month in advance and consists, on the one hand, in selecting the most profitable requests for execution, on the other hand – in building a sequence of cargo and empty runs that will fulfill the selected requests with the greatest efficiency. Direct transportation of loaded and empty railway cars is carried out by Russian Railways with pre-known tariffs and time standards for each of the routes. At the same time, tariffs for driving loaded wagons are additional costs for the customer of the route specified in the request (customers pay both the transport operator for the use of wagons and Russian Railways); transportation of empty wagons is paid by transport operators. To solve this problem, one of the possible ways to reduce it to a large-dimensional linear programming problem is proposed. An algorithm is proposed, the result of which is a problem written in the format of a linear programming problem. To demonstrate the approach clearly, a simplified problem statement is considered that takes into account only the main factors of the modeled process. The paper also shows an example of a numerical solution of the problem based on simple model data.

Key words: railway cargo transportation; planning of railway cargo transportation; optimal plan of railway cargo transportation; optimal management of railway car fleet; linear programming; schedule theory; operations research; unmanned vehicles.

Citation: Belousov F.A., Nevolin I.V., Khachatryan N.K. (2020) Modeling and optimization of plans for railway freight transport performed by a transport operator. *Business Informatics*, vol. 14, no 2, pp. 21–35. DOI: 10.17323/2587-814X.2020.2.21.35

Introduction

In the mathematical field called operations research, and in its subsection called schedule theory, there are many problems related to optimizing railway management. A whole series of such problems, as well as their classification, can be found in [1–5]. Among the problems of drawing up railway transport schedules, a class of problems related to drawing up schedules for passenger transport can be considered separately. One can find examples of such models in [6–9].

This article deals with railway freight transport. One of the urgent tasks that arise in this area is the organization of the cargo transportation process. In particular, works [10–13] are dedicated to this subject. In these articles, dynamic models of organization of railway freight transportation both between two junction stations and on a closed chain of stations are described and investigated.

This study is devoted to the no less important task which is management of the freight wagon fleet. This problem arises for railway transport operators (hereinafter, transport operators), which manage a fleet of freight rail cars for commercial purposes. Depending on the specifics of regulation and market features, different models can be built for each specific region, taking into account this or that specificity. As an example, one can consider the article [14], which presents a model used by one of the largest railway transport operators in Latin America. Another example is work [15], which examines several models for optimizing cargo delivery by the Swiss railway freight company Cargo Express Service of Swiss Federal Railways. In [16, 17] models designed with the features of the freight transport market in Italy are considered. In [18, 19] models are presented for cost minimization of transporting goods through a railway network that covers several European countries.

There are also models created for the Russian rail transport market; an example of such a model is presented in [20, 21]. In this paper, the problem statement is similar to the statement from [20, 21], the main differences are in the methods of solving the problem. If in [20, 21] solutions are sought using the column generation method [22], as well as using the modified column generation method [23], in this paper the solution is sought based on the search for optimal flows in a network covering all possible routes, by reducing it to a large-dimensional linear programming problem.

In other words, if the column generation method and the generalized column generation method search for solutions iteratively, and each iteration solves a linear programming problem on a subset of all possible routes, then in this paper the search for solution is performed using a single linear programming problem of a sufficiently large dimension on the set of all possible routes. But before getting such large-dimensional linear programming problem, one first needs to build a network of all possible routes (in this case, a wider network that includes a network of all possible routes is built), after that this task can be formalized as a linear programming problem.

The network of all possible routes is an oriented space-time graph without loops. This graph is built with a fixed time step, and one day is taken as a step. Each vertex of the graph contains information about the number of cars at a certain station on a certain day of the planning period. Each edge of this graph characterizes the route by leaving some station on day and arriving at another station on day . In this case, the difference between and corresponds to the number of days it takes to transfer cars from station to station in accordance with the known time standards for transportation by rail. In this interpretation, the task is to search for flows in the constructed graph that provide the maximum gain in total. The flow refers to a chain of cargo and empty runs, and the resulting gain corresponds to the profit for the planning period. For more information about building such models and reducing them to linear programming problems see [24–26].

This approach, which consists in solving the problem of linear programming of large dimensions, in comparison with the approach associated with column generation method, requires more time to find solutions. The advantage of the approach proposed here is the search for an optimal transportation plan on a set of all possible routes, while methods related to column generation solve a series of linear programming problems on subsets of the set of all routes. As a result, the solution obtained using the column generation method may differ from the optimal plan. In practice, in this case the lost profit for the transport operator can be measured in tens of millions of rubles per month.

1. General statement of the problem

The problem of managing a fleet of freight wagons by a transport operator is considered. The goal of the transport operator is to maximize profit. Drawing up a plan is carried out for a month in advance at a time when all the necessary information is known. For planning, one needs information about the initial location of cars in the next month, as well as a ready list of requests for cargo transportation in the next month. The initial position of wagons in the planning month implies information about day and station, in which each of the wagons arrived after being sent in previous month. The list of requests consists of requests from customers, each of which specifies the cargo, its volume (in wagons), stations of departure and destination. Each request also specifies the rate that the customer has to pay for each railway car of the transported cargo. A situation is allowed in which the start of request execution will be in the planning month, but its completion will be in the next month after the planning one. In this case, all profit for the execution of such request will be taken into account in the planning month. It is assumed that customers do not care what day of the planning month his order will be fulfilled; if the operator undertakes to execute this order, it will be executed on the most convenient day for the transport operator (or on several days if the request will be executed by several routes). The transport operator is not required to execute all incoming requests – as a rule, it is physically impossible to do this in the allotted month. Therefore, the operator can either execute the request completely or partially, or reject it. Thus, when creating a plan for the upcoming month, the task of the transport operator is, first, to select those requests that are most profitable to execute, and secondly, to select such chains of cargo and empty runs that will most effectively ensure the implementation of the selected requests.

Direct transportation of wagons is carried out by Russian Railways (JSC "RZD"), which set their own tariffs for both empty and cargo runs. Also time standards for all possible routes are known in advance. In the model, it is assumed that tariffs do not depend on the number of wagons transported on each of the routes. Each customer, if his request is executed, in addition to paying the operator the specified rate for the use of its wagons, pays separately to Russian Railways for the transportation of these wagons. Moving empty railway cars by Russian Railways is an expense item for the transport operator. Since Russian Railways tariffs for transporting loaded wagons are costs for customers and transport operators have nothing to do with them, these tariffs are not considered in the model. In this paper, a simplified statement of the problem is considered. This means that the transport operator manages one type of freight wagons, and the planning horizon in the model is one month. In addition, it will be assumed that direct transportation by Russian Railways is carried out largely by unmanned vehicles, i.e. unmanned locomotives. Their use will reduce the impact of the human factor, the occurrence of which often leads to certain emergencies, which in turn leads to failures in schedules. This assumption about the use of unmanned vehicles makes it possible to consider a deterministic model, without taking into account stochastic components. Otherwise, the model must take into account random factors, which would inevitably lead to a significant complication of the model.

2. Mathematical statement of the problem

This section provides a mathematical statement of the problem in some intermediate format, which is converted to the format of a linear programming problem in explicit form at the next stage. The advantage of this intermediate format is that it is much more visual and convenient for understanding the essence of the proposed approach. A similar format for the mathematical statement of this problem can be found in [3, 4], but in these works the mathematical statement has a rather cumbersome form and can hardly be applied directly in practice. In this paper, to facilitate the presentation, the simplest version of the problem statement is presented.

Let's enter a number of notations.

N – number of stations involved in planning;

T – planning horizon, measured in days, for simplicity one month is taken as the length of the planning horizon in this work (i.e. $T = 30$ or 31). In practice, however, it is more correct to consider a longer planning horizon, such as two or more months, but for simplicity in this work a short and plausible interval is taken;

t – the discrete parameter responsible for time is measured in days and takes values $t = 1, 2, \dots T$;

$C = \{C_{ij}\}_{i,j=1}^{N}$ – ($N \times N$)-matrix, which elements characterize the tariff set by Russian Railways for an empty run of one wagon from station i to station j;

$\Theta 1 = \{\Theta 1_{ij}\}_{i,j=1}^{N}$ – ($N \times N$)-matrix, which elements characterize the time (in days) of movement of loaded wagons from station i to station j in accordance with Russian Railways standards (time is rounded to a larger integer);

$\Theta 2 = \{\Theta 2_{ij}\}_{i,j=1}^{N}$ – ($N \times N$)-matrix, which elements characterize the time (in days) of movement of empty wagons from station i to station j in accordance with Russian Railways standards (time is rounded to a larger integer). Separately, we note that the diagonal elements of this matrix are taken to be equal to one, which means that if a wagon remains at the station until the next day, it is equivalent to the fact that it goes on a loop trip lasting one day, where the departure and destination stations coincide;

$P = \{P_{ij}\}_{i,j=1}^{N}$ – ($N \times N$)-matrix, which elements characterize the rate specified by the customer in the request for transportation of one loaded wagon from station i to station j;

$\bar{Q} = \{\bar{Q}_{ij}\}_{i,j=1}^{N}$ – ($N \times N$)-matrix, which elements characterize the number of loaded wagons specified in the corresponding request for cargo transportation from station i to station j. All elements of the matrix take non-negative integer values;

$\bar{S}^0(t) = \{\bar{S}_i^0(t)\}_{i=1}^{N}$ – vector of dimension N that characterizes the initial location of wagons on day *t*, the *i*-th element of this vector equals to the number of wagons that arrived at station *i* at time $t \in \{1, ..., T\}$, which were dispatched in the previous month. All values of this vector take non-negative integer values.

The transport plan is characterized by the following matrices:

$K1(t) = \{K1_{ij}(t)\}_{i,j=1}^{N}$ – $(N \times N)$-matrix, which elements characterize the number of loaded wagons sent from station *i* to station *j* at time $t \in \{1, ..., T\}$. All elements of the matrix take non-negative integer values;

$K2(t) = \{K2_{ij}(t)\}_{i,j=1}^{N}$ – $(N \times N)$-matrix, *ij* element of which characterizes number of empty wagons sent from station *i* to station *j* at time $t \in \{1, ..., T\}$. All elements of the matrix take non-negative integer values;

Denote by *K*1 and *K*2 the set of corresponding matrices for all moments of time $t \in \{1, ..., T\}$, in other words $K1 = \{K1(t)\}_{t=1}^{T}$, $K2 = \{K2(t)\}_{t=1}^{T}$.

Also of interest is the final distribution of wagons by stations and time in the planning month in accordance with the proposed plan *K*1, *K*2. $\bar{S}(t, K1, K2) = \{\bar{S}_i(t, K1, K2)\}_{i=1}^{N}$ – vector of length *N*, each element of which characterizes the number of wagons at station *i* at time $t \in \{1, ..., T\}$, which is implemented in accordance with the proposed plan *K*1 and *K*2 and the initial distribution of wagons $\bar{S}^0(t)$. It is easy to see that for all $t \in \{1, ..., T\}$ and any station $i \in \{1, ..., N\}$, the value of the element $\bar{S}_i(t, K1, K2)$ is determined by the formula below:

$$\bar{S}_i(t, K1, K2) = \bar{S}_i^0(t) + \\ + \sum_{\tau=1}^{t}\left(\sum_{j=1}^{N} K1_{ji}(\tau) I\left(t-\tau-\Theta1_{ji}\right) + \right. \\ \left. + \sum_{j=1}^{N} K2_{ji}(\tau) I\left(t-\tau-\Theta2_{ji}\right)\right), \tag{1}$$

where the function $I(\cdot)$ is defined by the rule:

$$I(x) = \begin{cases} 1, & if \;\; x = 0; \\ 0, & othewise. \end{cases}$$

In other words, the number of wagons that is observed at station *i* at time *t* is equal to the number of wagons that arrived from the previous month in accordance with the value $\bar{S}_i^0(t)$, as well as the number of wagons that were sent to station *i* by empty or loaded runs on the days τ preceding the current day *t* ($\tau \in \{1, ..., t-1\}$), and that arrive at station *i* on this day *t*.

Now the mathematical statement of this problem can be written. Profit is the criterion of maximization

$$\sum_{t=1}^{T}\left(\sum_{i,j=1}^{N} P_{ij} K1_{ij}(t) - \right. \\ \left. - \sum_{i,j=1}^{N} C_{ij} K2_{ij}(t)\right) \rightarrow \max_{\{K1_{ij}(t), K2_{ij}(t)\}} . \tag{2}$$

The following restrictions must be satisfied:

$$\bar{S}_i(t, K1, K2) = \sum_{j=1}^{N}\left(K1_{ij}(t) + K2_{ij}(t)\right), \\ i = \overline{1, N}, \;\; t = \overline{1, T}; \tag{3}$$

$$\sum_{t=1}^{T} K1_{ij}(t) \le \bar{Q}_{ij}, \quad i = \overline{1, N}, \; j = \overline{1, N}; \tag{4}$$

$$K1_{ij}(t) \in \mathbb{N} \cup \{0\}, \; K2_{ij}(t) \in \mathbb{N} \cup \{0\}, \\ i = \overline{1, N}, \; j = \overline{1, N}, \; t = \overline{1, T}. \tag{5}$$

The target function (2) represents the profit from all freight runs after deduction of the costs associated with empty wagons runs. Optimization is performed by managing amount of loaded runs $K1_{ij}(t)$ and empty runs $K2_{ij}(t)$. Restriction (3) is a balance restriction and means that the number of wagons sent from station *i* on day $t \in \{1, ..., T\}$ is exactly equal to the number of wagons that arrived there on that day. The number of arriving wagons $\bar{S}_i(t, K1, K2)$ is determined in accordance with formula (1). Restriction (4) means that the number of loaded wagons sent from station *i* to station *j* on all days of the planning period must not exceed the number specified in the relevant request.

Problem (2)–(5) can be solved directly using Mixed-Integer Programming. However, due to the large dimension, the task of finding an integer solution may be impossible within a reasonable time. Therefore, instead of the original problem, linear relaxation is considered. This means that the following weaker condition is considered instead of condition (5):

$$K1_{ij}(t) \geq 0,\ K1_{ij}(t) \geq 0,\ i = \overline{1,N},\ t = \overline{1,T} \qquad (6)$$

Thus, instead of problem (2)–(5), the following problem is considered (2), (3), (4), (6). The result of solving such a problem may be a fractional solution, which obviously cannot be applied in practice. In this case, to get an integer solution, you must apply rounding methods to the resulting fractional solution. At the next stage, this statement of the problem will be rewritten in the format of linear programming.

3. Reducing the original problem to a linear programming problem

In order to solve the formulated problem by computer, the above designations must be presented in a different format. To do this, the matrices P, C, $\bar{Q}$, as well as the matrix system $K1(t)$ and $K2(t)$, $t \in \{1, ..., T\}$ must be converted to long vectors. The system of vectors $\bar{S}^0$ and $\bar{S}(t, K1, K2)$, $t \in \{1, ..., T\}$ also must be converted into two single vectors. On the one hand, the matrices P and C are transformed into one PC vector by a special rule; on the other hand, by a similar rule, the matrices $K1(t)$ and $K2(t)$, $t \in \{1, ..., T\}$ are also combined into a vector K of the same dimension consistent with the PC vector. This is done so that the profit of the transport operator, which calculation formula is used in (2), completely coincides with the scalar product of the two new vectors PC and K.

Vector *K*

Since the number of elements in all matrices $K1(t)$ and $K2(t)$, $t \in \{1, ..., T\}$ is equal to $2TN^2$, the dimensions of the vectors PC and K also have to coincide with this value. When forming vector K, the question is in what order to place all the elements of the matrices $K1(t)$ and $K2(t)$, $t \in \{1, ..., T\}$. After this order is defined, the PC vector is compiled accordingly and in the appropriate order. Let the first TN^2 elements of the vector K correspond to all elements of matrices $K1(t)$, $t \in \{1, ..., T\}$. In this case, the first N^2 elements are taken from the matrix $K1(1)$: the first N elements coincide with the first row of this matrix (the first row element is placed at the first position of the vector, the second row element – at the second position, etc.), the next N elements correspond to the second row, and so on. The next N^2 elements of the vector K are written using a similar ordering of the elements of the matrix $K1(2)$. By the same principle, the vector K is filled with elements of the other matrices $K1(t)$, $t = 3, ..., T$. The second half of the vector K, also consisting of TN^2 elements, is filled in the same way only by elements of the matrices $K2(t)$, $t = 1, ..., T$.

Vectors *PC*

In order to match the vector PC with the vector K, the first TN^2 of its elements are taken from the matrix P, the remaining TN^2 are filled from matrix C. The first N elements of the PC vector correspond to the first row of the P matrix (the first row element is placed in the first position of the vector, the second row element - in the second position, etc.), the next N elements correspond to the second row of the P matrix, and so on. After the first N^2 elements of the PC vector are defined by the specified procedure, this part of the vector is copied and put in place of the next N^2 elements. Repeating this operation $T - 1$ times completes the formation of the first half of the PC vector, consisting of

TN^2 elements. The second half of this vector is filled in similarly with elements of the matrix C. The only difference in the fill-in rule is that each element of the second half of the PC vector is additionally multiplied by minus one. Thus, the first TN^2 elements of the PC vector are non-negative, and the next TN^2 elements of this vector are non-positive.

Vector Q, S^0, S

Vector Q is formed from the matrix $\bar{Q}$ by a similar rule (the first N elements of the vector correspond to the first row of the matrix, the next N elements correspond to the second row, and so on). The dimension of the new vector is N^2.

Systems of vectors $\bar{S}^0(t)$ and $\bar{S}(t, K1, K2)$, $t \in \{1, ..., T\}$ are transformed into two vectors by sequential concatenation of the vectors corresponding to each moment of time (the first N elements of each of the new vectors correspond to the vector $\bar{S}^0(1)$ and $\bar{S}(1, K1, K2)$, respectively, and so on). In the end, we get two new vectors of size TN. The first one is denoted by through S^0, the second – through S.

With vectors PC and K, the optimization criterion (2) and condition (6) can be written in the new format. However, in order to write conditions (3) and (4), it is necessary to introduce special matrices.

Matrix A_Q

Let's construct a matrix A_Q, which allows us to rewrite the condition (4). It is easy to see that the dimension of this matrix is equal to $(N^2 \times 2TN^2)$. This is due to the fact that this matrix on the right is multiplied by the vector K, and the result of this multiplication is compared with the vector Q of dimension N^2. Based on the fact that the elements of the matrix $K2$ (correspond to the last TN^2 elements of the vector K) do not participate in this restriction, all the elements of the matrix A_Q located in the last TN^2 columns, are equal to zero. Now the first TN^2 columns of this matrix are necessary to define. Condition (4) means that for each route from station i to station j, the total number of loaded wagons sent on all days of the planning period cannot exceed the amount specified in the request – $\bar{Q}_{ij}$. Based on this, it is easy to see that the first element of the first row is equal to one (it corresponds to the element $K1_{11}(1)$). The next element equal to one located in the first row of matrix A_Q is at position $N^2 + 1$ (it corresponds to the element $K1_{12}(2)$) and so on – all elements equal to one occur in the first row T times with a period of N^2 elements. Further, in the second row of the matrix A_Q, the element equal to one is in the second position (it corresponds to the element $K1_{12}(1)$), the next element equal to one in the second row is at the position $N^2 + 2$ (it corresponds to the element $K1_{12}(2)$) and so on: also, with a frequency of N^2 elements, all elements equal to one occur in the second row T times. For other rows of the A_Q matrix, the filling algorithm is similar. Thus, if one divides matrix A_Q into T blocks, each of size N^2 by N^2 elements, not considering the zero part (the last TN^2 columns), then each block is equal to the identity matrix. After defining the matrix A_Q in the new format, condition (4) takes the form of $A_Q \cdot K \leq Q$, where the sign $\leq$ implies a component-by-component comparison of elements of both vectors.

Matrix A_{in} and A_{out}

It remains to rewrite condition (3). To do this in matrix form, one needs to define two matrices, denoted by A_{in} and A_{out}. The A_{in} matrix is used to calculate the number of incoming wagons to each of the stations at any time $t \in \{1, ..., T\}$. The A_{out} matrix is used to calculate the number of outgoing wagons from each station at any time $t \in \{1, ..., T\}$. The dimension of each of the matrices is $(TN \times 2TN^2)$. This is due to the fact that these matrices are multiplied on the right by the vector K. The result of the product of the vector K on any of these matrices is related to the distribution of wagons by stations and time, so the number of rows in these matrices and the dimension of the resulting product of the vector must be TN. The first N elements of the resulting vector correspond to the distribution of wagons at all stations in the first period, the next N elements correspond to the distribution of wagons at all stations in the second period, and so on until the period T. The matrix A_{in} allows us to represent the formula (1) as $S = A_{in} \cdot K$.

Let's write the A_{out} matrix. As already mentioned, this matrix is designed to count the number of wagons sent from an each station i at any time $t \in \{1, ..., T\}$. Where exactly these wagons are sent, in this case, is of no interest. Taking into account the order of elements of the vector K, it is easy to see that the first N elements of the first row of the A_{out} matrix are responsible for the loaded wagons sent from the first station in the first period of time. So all these elements are equal to one. Then, all elements, starting from the element $N + 1$, up to the element TN^2 inclusive, are equal to zero. The elements, starting from $TN^2 + 1$, before the element $TN^2 + N$, are equal to one. These elements are responsible for outgoing empty wagons. Other elements of the first row of the A_{out} matrix are zero. The second row of the A_{out} matrix has a similar structure, with the difference that everything is shifted by N elements to the right. In other words, the first N elements of the second row are equal to zero, the next N elements are equal to one, then, starting from $TN^2 + N + 1$, up to $TN^2 + 2N$, all elements are equal to one, and the remaining elements of the second row are equal to zero. And so on, in each subsequent row, the coordinates of elements equal to one are shifted by N elements. Repeating this operation for all rows of the A_{out} matrix completes its formation.

It remains to define the A_{in} matrix. For each of the loaded and empty runs from station i to station j, the duration of such a run is known. It is equal to $\Theta 1_{ij}$ for cargo and $\Theta 2_{ij}$ for empty runs. Let's see how this information is reflected in the A_{in} matrix. To do this, consider an arbitrary transfer of loaded wagons in the amount of $K1_{ij}(t)$ from station i to station j on day t of the planning period. This transfer corresponds to vector K element located at the position $TN^2 + (i - 1)N + j$. Travel time on this route equals $\Theta 1_{ij}$. This means that after leaving on day t, the dispatched wagons will end up at station j on day $t + \Theta 1_{ij}$. In terms of equation (1), it can be stated that as a result of this loaded trip, the value of the element $\overline{S}_j\left(t + \Theta 1_{ij}, K1, K2\right)$ will increase by $K1_{ij}(t)$ units. Therefore, if $t + \Theta 1_{ij}$ does not exceed T, then the result of multiplying A_{in} by K is the increase in the coordinates $(t + \Theta 1_{ij} - 1)N + j$ of the vector S by $K1_{ij}(t)$ units. In order for this multiplication to give such a result, it is necessary that the element of the matrix A_{in} with the coordinates $\left(\left(t + \Theta 1_{ij} - 1\right)N + j, tN^2 + \left(i - 1\right)N + j\right)$ be equal to one. An arbitrary empty run in the amount of $K2_{ij}(t)$ wagons from station i to station j on day t of the planning period, in case when $t + \Theta 2_{ij}$ does not exceed T, corresponds to the matrix A_{in} element equals to one with coordinates $\left(\left(t + \Theta 2_{ij} - 1\right)N + j, tN^2 + \left(i - 1\right)N + j\right)$. It can be seen that in the case of empty runs, the value TN^2 is additionally added to the second component of the coordinate of the matrix A_{in} element equals to one. This is due to the fact that the empty runs in the vector K correspond to the second half of this vector, which begins with the coordinate $TN^2 + 1$ and ends with the last element with the coordinate $2TN^2$. Thus, the algorithm for constructing the A_{in} matrix consists of taking a null matrix of size TN by $2TN^2$ and placing elements equal to one in it, iterating over all the elements of the vector K (or, what is the same, iterating over all the elements of the matrices $K1(t)$ and $K2(t), t \in \{1, ..., T\}$).

After the A_{in} and A_{out} matrices are defined, the constraint (3) can be rewritten as $A_{in} \cdot K + S_0 = A_{out} \cdot K$ or, similarly, as $(A_{out} - A_{in}) \cdot K = S_0$.

Linear programming problem

After all vectors and matrices are defined in the new format, the problem (2), (3), (4), (6) may be rewritten in a new format:

$$PC^T \cdot K \to \max_K, \qquad (7)$$

Subject to

$$(A_{out} - A_{in}) \cdot K = S_0; \qquad (8)$$

$$A_Q \cdot K \leq Q; \qquad (9)$$

$$K \geq 0. \qquad (10)$$

The problem statement (7)–(10) is absolutely identical to the task (2), (3), (4), (6), but its advantage is that it is written in the format of a classical linear programming problem, which allows one to solve it using appropriate methods and software tools.

4. Solving transport problem using artificial data as an example

As an illustrative example of solving the transport problem, a simple model example with a small number of stations and a short planning horizon is considered.

Let the number of stations be 4 ($N = 4$), and the planning horizon is 3 days ($T = 3$). The list of received requests consists of five items. We present these requests in *Table 1.*

Based on the list of requests, one needs to create two matrices – the matrix of tariffs P, elements of which are written in conditional units, and the matrix of the volume of requests $\overline{Q}$:

$$P = \begin{pmatrix} 0 & 0 & 2.9 & 0 \\ 1.1 & 0 & 2.3 & 0 \\ 0 & 1.9 & 0 & 2.1 \\ 0 & 0 & 0 & 0 \end{pmatrix}; \ \overline{Q} = \begin{pmatrix} 0 & 0 & 3 & 0 \\ 5 & 0 & 4 & 0 \\ 0 & 7 & 0 & 6 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Travel time of both loaded and empty wagons are given in matrices $\Theta 1$ and $\Theta 2$ bellow:

$$\Theta 1 = \begin{pmatrix} 0 & 2 & 1 & 2 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 2 \\ 1 & 2 & 1 & 0 \end{pmatrix}; \ \Theta 2 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 \\ 1 & 2 & 1 & 1 \end{pmatrix}.$$

Recall that the diagonal elements of the matrix are taken to be equal to one. This is due to the fact that if the wagons are left at the station until the next day, it is equivalent to their being sent from this station to itself on a trip lasting one day.

Table 1.

List of requests for cargo transportation in the model example

№	Departure station	Destination station	Volume of requests (in wagons)	Rate (in conditional units)
1	1	3	3	2.9
2	2	1	5	1.1
3	2	3	4	2.3
4	3	2	7	1.9
5	3	4	6	2.1

The values of Russian Railways tariffs for empty runs, as well as rates expressed in conditional units, are characterized by the values of matrix elements *C*:

$$C = \begin{pmatrix} 0 & 1.9 & 1.3 & 1.9 \\ 1.2 & 0 & 1.8 & 0.9 \\ 1.1 & 1.2 & 0 & 1.6 \\ 1.3 & 1.5 & 1.2 & 0 \end{pmatrix}.$$

It is assumed that wagons can stay at stations until the next day for free, so the diagonal elements of the matrix *C* are zero.

The initial distribution of wagons is characterized by the following vectors:

$$\bar{S}^0(1) = \begin{pmatrix} 0 \\ 2 \\ 1 \\ 3 \end{pmatrix}; \ \bar{S}^0(2) = \begin{pmatrix} 5 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

During the period $t = 3$, wagons do not arrive, which is equivalent to the zero vector $\bar{S}^0(3)$.

Let's rewrite this problem in the new notation. The *PC* vector has a dimension of $2TN^2 = $ $=2 \cdot 3 \cdot 16 = 96$. The representation of such a vector in explicit form will not fit on the page, so let's consider the intermediate vectors *p* and *c* of length $N^2 = 16$, made up on the basis of matrices *P* and *C*:

$$p = \begin{pmatrix} 0 \\ 0 \\ 2.9 \\ 0 \\ 0 \\ 1.1 \\ 0 \\ 2.3 \\ 0 \\ 0 \\ 1.9 \\ 0 \\ 2.1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} c = \begin{pmatrix} 0 \\ 1.9 \\ 1.3 \\ 1.9 \\ 1.2 \\ 0 \\ 1.8 \\ 0.9 \\ 1.1 \\ 1.2 \\ 0 \\ 1.4 \\ 1.3 \\ 1.5 \\ 1.2 \\ 0 \end{pmatrix}.$$

The *PC* vector is obtained by successive concatenation of these vectors, namely:

$$PC = \left(p^T,\ p^T,\ p^T,\ c^T,\ c^T,\ c^T\right)^T.$$

The vector *Q*, obtained from the matrix $\bar{Q}$ and having dimension $N^2 = 16$, takes the following form:

$$Q = \begin{pmatrix} 0 \\ 0 \\ 3 \\ 0 \\ 5 \\ 0 \\ 4 \\ 0 \\ 0 \\ 7 \\ 0 \\ 6 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

The vector S^0, which dimension is $TN = 3 \cdot 4 =$ $= 12$, is constructed on the basis of vectors $\bar{S}^0(1)$ and $\bar{S}^0(2)$ and has the following form:

$$S^0 = \begin{pmatrix} 0 & 2 & 1 & 3 & 5 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}^T.$$

It remains to get the matrices A_{in}, A_{out} and A_Q. Dimension of the matrix A_Q equals to $(N^2 \times 2TN^2) = (16 \times 96)$. Let's write it in the sparse matrix format, i.e. specify the coordinates of non-zero elements equal to one. The list of coordinates of elements equal to one of the A_Q matrix is as follows (here and everywhere else, the numbering of rows and columns begins with one):

(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6), (7, 7), (8, 8), (9, 9), (10, 10), (11, 11), (12, 12), (13, 13), (14, 14), (15, 15), (16, 16), (1, 17), (2, 18), (3, 19), (4, 20), (5, 21), (6, 22), (7, 23), (8, 24), (9, 25), (10, 26), (11, 27), (12, 28), (13, 29), (14, 30), (15, 31), (16, 32), (1, 33), (2, 34), (3, 35), (4, 36), (5, 37), (6, 38), (7, 39), (8, 40), (9, 41), (10, 42), (11, 43), (12, 44), (13, 45), (14, 46), (15, 47), (16, 48).

The same format is used for the A_{in} and A_{out} matrices, which dimension is $(TN \times 2TN^2 = (12 \times 96)$. The list of coordinates of elements equal to one of the A_{in} matrix:

(5, 1), (7, 3), (12,4), (5, 5), (6, 6), (8, 8), (5, 9), (10, 10), (7, 11), (5, 13), (10, 14), (7, 15), (8, 16), (9, 17), (11, 19), (9, 21), (10, 22), (12, 24), (9, 25), (11, 27), (9, 25), (11, 27), (9, 29), (11, 31), (12, 32), (5, 49), (7, 51), (5, 53), (6, 54), (8, 56), (5, 57), (10, 58), (7, 59), (5, 61), (10, 62), (7, 63), (8, 64), (9, 65), (10, 70), (12, 72), (9, 73), (11, 75), (9, 77), (11, 79), (12, 80).

List of coordinates of elements equal to one of the A_{out} matrix:

(1, 1), (1, 2), (1, 3), (1, 4), (2, 5), (2, 6), (2, 7), (2, 8), (3, 9), (3, 10), (3, 11), (3, 12), (4, 13), (4, 14), (4, 15), (4, 16), (5, 17), (5, 18), (5, 19), (5, 20), (6, 21), (6, 22), (6, 23), (6, 24), (7, 25), (7, 26), (7, 27), (7, 28), (8, 29), (8, 30), (8, 31), (8, 32), (9, 33), (9, 34), (9, 35), (9, 36), (10, 37), (10, 38), (10, 39), (10, 40), (11, 41), (11, 42), (11, 43), (11, 44), (12, 45), (12, 46), (12, 47), (12, 48), (1, 49), (1, 50), (1, 51), (1, 52), (2, 53), (2, 54), (2, 55), (2, 56), (3, 57), (3, 58), (3, 59), (3, 60), (4, 61), (4, 62), (4, 63), (4, 64), (5, 65), (5, 66), (5, 67), (5, 68), (6, 69), (6, 70), (6, 71), (6, 72), (7, 73), (7, 74), (7, 75), (7, 76), (8, 77), (8, 78), (8, 79), (8, 80), (9, 81), (9, 82), (9, 83), (9, 84), (10, 85), (10, 86), (10, 87), (10, 88), (11, 89), (11, 90), (11, 91), (11, 92), (12, 93), (12, 94), (12, 95), (12, 96).

After all vectors and matrices are defined, problem (7)–(10) can be solved. The result of solving this problem is vector K, which dimension is $2TN^2 = 96$. Let's give one of the solutions found, writing out the values of only nonzero elements of vector K:

$K_7 = 2$; $K_{10} = 1$; $K^{19} = 2$; $K_{37} = 2$; $K_{42} = 4$; $K_{44} = 2$; $K_{62} = 3$; $K_{65} = 2$; $K_{67} = 1$; $K_{79} = 1$; $K_{81} = 2$.

This solution may be rewritten in the format of matrices $K1(t)$ and $K2(t)$:

$$K1(1) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; \; K1(2) = \begin{pmatrix} 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix};$$

$$K1(3) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 2 & 0 & 6 \\ 0 & 0 & 0 & 0 \end{pmatrix}; \qquad (11)$$

$$K2(1) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 0 \end{pmatrix}; K2(2) = \begin{pmatrix} 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix};$$

$$K2(3) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \qquad (12)$$

The resulting solution (11), (12), due to the small scale of the problem, can also be represented by diagram shown in *Figure 1.*

The profit of the resulting solution can be calculated in two equivalent ways – either based on the formula used in (2), or as the product of the vectors $PC^T \cdot K$. For this problem, its value, expressed in conditional units, is equal to 32.3.

Let's analyze the resulting solution. It can be seen that one of the requests for the transportation of 5 wagons from station 2 to station 1 remained unfulfilled. The request for transportation of 7 wagons from station 3 to station 2 must be partially fulfilled in the amount of 5 wagons. All other requests according to the plan are going to be fulfilled wholly. In accordance with the plan requests from station 1 to station 3, as well as from station 3 to station 4, are going to be executed completely by using one run. Requests from station 2 to station 3 and from station 3 to station 2 have to be executed in stages – two

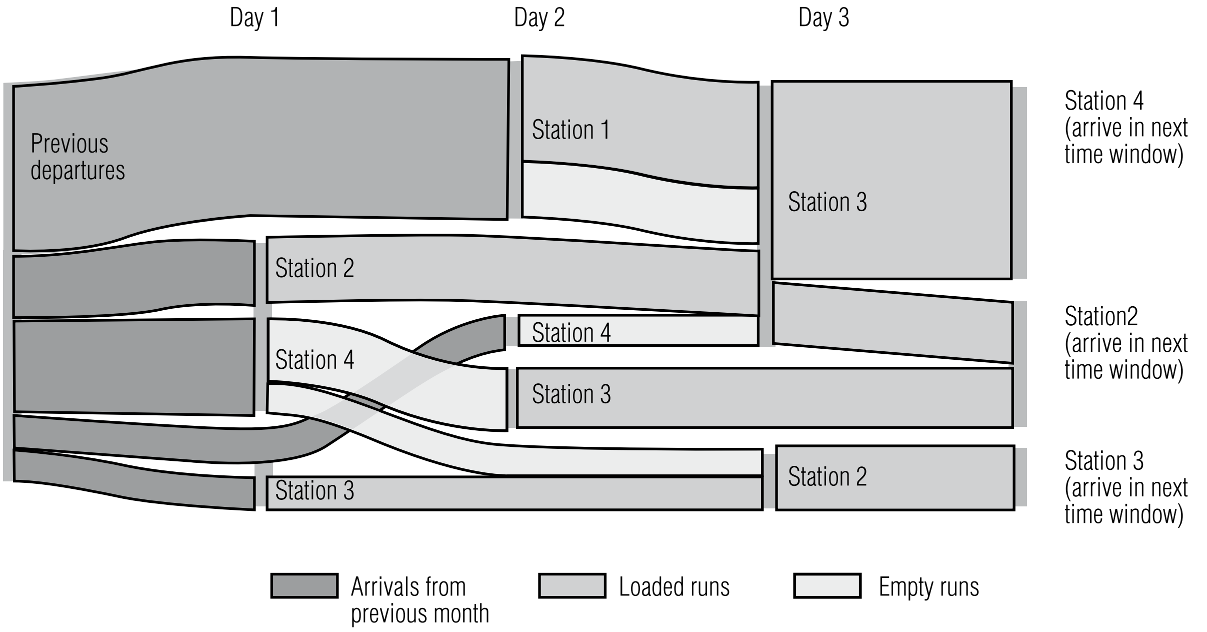

Day 1
Day 2
Day 3
Previous departures
Station 1
Station 3
Station 4 (arrive in next time window)
Station 2
Station 4
Station 4
Station 3
Station2 (arrive in next time window)
Station 3
Station 2
Station 3 (arrive in next time window)
Arrivals from previous month
Loaded runs
Empty runs

Fig. 1. Schematic representation of the solution (11), (12)

runs for the first request and three runs for the second one. Also, the plan found offers execution of several empty runs. So, on the first day, one empty wagon is sent from station 4 to station 2, and two empty wagons are sent from station 4 to station 3. The duration of the first empty run is 2 days (the wagon will arrive at station 2 by the third period), the duration of the second empty run is one day. Similarly, on the second day there are two empty runs: one wagon from station 4 to station 3 and two wagons from station 1 to station 3, the duration of both runs is one day. There are no empty runs scheduled for the final third period.

Conclusion

This paper considers the transport problem that arises for transport operators when managing a fleet of freight rail cars. The approach proposed in this paper allows us to find the optimal plan on the entire set of possible routes, while the methods associated with the column generation method search for optimal plans on subsets of all possible routes. The significant disadvantage of the proposed approach is that it requires significantly more time for calculation then methods associated with generating columns approach. Therefore, it is important to suggest ways to modify this approach, which will lead to acceleration of computing processes.

The large dimension of the problem is mainly due to the fact that statement (2)–(4), (6) takes into account all possible routes, including those that do not actually exist. In particular, this applies to cargo routes. In this statement, all cargo routes are considered, although only a small percentage of them are relevant (only those specified in the requests are relevant).

When solving the problem based on real data, the number of stations can be equal to about 1000, the dimension of the vector K in this case will be about $2TN^2 = 1.2 \cdot 10^8$, in the case when planning horizon is $T = 60$. It is clear that in practice, the linear programming problem with such a dimension cannot be solved in a reasonable time.

Vector K consists of two parts, the first part is based on matrices $K1(t)$; the second part is based on matrices $K1(t)$. If for empty routes it is not known in advance which of them will be used and which will not, then in the case of cargo routes it is known that only those specified in the requests will be in use, so the remaining routes may be removed from consideration. This means that it is more rational to take into account only those elements of the matrices $K1(t)$, $t \in \{1, ..., T\}$ that correspond to the cargo routes specified in the submitted requests. Only these cargo routes should also be taken into account in the K vector, as there is no sense to take into account other cargo routes. Thus, the dimension of the vector K can be significantly reduced. Obviously, it is necessary to exclude such routes not only from the K vector: the PC vector generation algorithm, as well as the A_Q, A_{in} and A_{out} matrix generation algorithms, also must be modified. In this paper, to simplify the presentation, the specified algorithms for forming vectors and matrices are not considered.

To assess how much the size of the problem can be reduced, let's look at the data of a real problem, in which $N = 1126$, and the number of requests equal to the number of demanded cargo routes is 1616. Under these conditions, the first part of the modified vector K, which is responsible for cargo runs, at $T = 60$ has the dimension $T \cdot 1616 = 96960$ instead of $TN^2 = 76072560$. It is easy to see that the dimension of the first part of the vector K is reduced by almost 800 times. If the algorithm for forming the second part of the vector K responsible for empty runs is left unchanged, then the dimension of the vector K in the modified algorithm will be $T \cdot 1616 + TN^2 = 76169520$ instead of $2TN^2 =$ $= 152145120$, i.e. it will be reduced by almost half.

Separately, it can be noted that the planning horizon of one month is too short to solve real problems. This is due to the fact that with such a planning horizon, the result of optimization may be a plan in which a large number of requests with the maximum rate, as well as the maximum distance and duration of routes will be planned for the last days of the planned month. The implementation of this plan will lead to the maximum possible profit in the planning month; however, this is fraught with the fact that in the month following the planning month, the wagons may be distributed in a manner extremely inconvenient both in space and in time of arrival at destination stations. Obviously, this may lead to a noticeable decrease in profit in the next month. Based on this, the planning horizon in the optimization model should be increased. However, if one increases the planning horizon to several months, then each of these months should be planned for its own separate list of requests. Generally speaking, the lists of requests for different months should not coincide, but since only the current list of requests is known for sure and the requests for the following months are unknown, within the model it can be assumed that the lists of requests expected in the future months coincides with the current list. We consider it reasonable to take two months as the planning horizon, such an expansion of the planning horizon will lead to a twofold increase in the dimension of the problem.

Further work on the model should be done in the direction of taking into account more factors, as well as in the direction of optimizing computational processes by reducing the dimension of the problem. All of the above is expected to be taken into account in future works. ■

Acknowledgments

The reported study was funded by RFBR, project number 19-29-06003.

References

1. Lazarev A.A., Musatova E.G., Gafarov E.R., Kvaratskheliya A.G. (2012) *Schedule Theory. Problems of railway planning.* Moscow: IPU RAS (in Russian)
2. Lazarev A.A., Musatova E.G., Kvaratskheliya A.G., Gafarov E.R. (2012) *Schedule Theory. Problems of transport system management.* Moscow: MSU (in Russian).
3. Lusba R., Larsen J., Ehrgott M., Ryan D. (2011) Railway track allocation: models and methods. *OR Spectrum*, vol. 33, no 4, pp. 843–883. DOI: 10.1007/s00291-009-0189-0.
4. Cordean J.-F., Toth P., Vigo V. (1988) A survey of optimization models for train routing and scheduling. *Transportation Science*, vol. 32, no 4, pp. 380–404. DOI: 10.1287/trsc.32.4.380.
5. Ravindra K., Möhring R.H., Zaroliagis C.D., eds. (2009) *Robust and online large-scale optimization: Models and techniques for transportation systems.* Berlin, Heidelberg: Springer. DOI: 10.1007/978-3-642-05465-5.
6. Brucker P., Hurink J., Rolfes T. (1999) *Routing of railway carriages: A case study. Technical Report 1498.* Netherlands: University of Twente.
7. Brucker P., Hurink J., Rolfes T. (2003) Routing of railway carriages. *Journal of Global Optimization*, no 27, pp. 313–332. DOI: https://doi.org/10.1023/A:1024843208074.
8. Cordean J.F., Soumis F., Desrosiers J. (2001) Simultaneous assignment of locomotives and cars to passenger trains. *Operations Research*, vol. 49, no 4, pp. 531–548. DOI: 10.1287/opre.49.4.531.11226.
9. Eidenbenz S., Pagourtzis A., Widmayer P. (2003) Flexible train rostering. Algorithms and Computation. ISAAC 2003. *Lecture Notes in Computer Science*, vol. 2906, pp. 615–624. DOI: https://doi.org/10.1007/978-3-540-24587-2_63.
10. Beklaryan L.A., Khachatryan N.K. (2013) On one class of dynamic transportation models. *Computational Mathematics and Mathematical Physics*, vol. 53, no 10, pp. 1649–1667 (in Russian). DOI: 10.7868/S0044466913100037.
11. Khachatryan N.K., Akopov A.S. (2017) Model for organizing cargo transportation with an initial station of departure and a final station of cargo distribution. *Business Informatics*, no 1, pp. 25–35. DOI: 10.17323/1998-0663.2017.1.25.35.
12. Khachatryan N.K., Akopov A.S., Belousov F.A. (2018) About quasi-solutions of traveling wave type in models for organizing cargo transportation. *Business Informatics*, no 1, pp. 61–70. DOI: 10.17323/1998-0663.2018.1.61.70.
13. Beklaryan L.A., Khachatryan N.K. (2019) Dynamic models of cargo flow organization on Railway Transport. *Economics and Mathematical Methods*, vol. 55, no 3, pp. 62–73 (in Russian). DOI: 10.31857/S042473880005780-7.
14. Fukasawa R., Aragão M.P., Porto O., Uchoa E. (2002) Solving the freight car flow problem to optimality. *Electronic Notes in Theoretical Computer Science*, vol. 66, no 6, pp. 42–52. DOI: 10.1016/S1571-0661(04)80528-0.
15. Ceselli A., Gatto M., Lübbecke M., Nunkesser M., Schilling H. (2008) Optimizing the cargo express service of Swiss Federal Railways. *Transportation Science*, vol. 42, no 4, pp. 450–465. DOI: 10.1287/trsc.1080.0246.
16. Lulli G., Pietropaoli U., Ricciardi N. (2011) Service network design for freight railway transportation: the Italian case. *Journal of the Operational Research Society*, vol. 62, no 12, pp. 2107–2119. DOI: 10.1057/jors.2010.190.
17. Campetella M., Lulli G., Pietropaoli U., Ricciardi N. Fright service design for the Italian railways company. Proceedings of the *6th Workshop on Algorithmic Approach for Transportation Modelling, Optimization, and Systems (ATMOS 2006), Zurich, Switzerland, 14 September 2006*, pp. 1–13. DOI: 10.4230/OASIcs.ATMOS.2006.685.

18. Andersen J., Christiansen M. (2009) Designing new European rail freight services. *Journal of the Operational Research Society*, no 60, pp. 348–360. DOI: 10.1057/palgrave.jors.2602559.

19. Jeong S.-J., Lee C.-G., Bookbinder J. (2007) The European freight railway system as a hub-and-spoke network. *Transportation Research, Part A: Policy and Practice*, vol. 41, no 6, pp. 523–536. DOI: 10.1016/j.tra.2006.11.005.

20. Sadykov R., Lazarev A., Shiryaev V., Stratonnikov A. (2013) Solving a freight railcar flow problem arising in Russia. Proceedings of the *13th Workshop on Algorithmic Approach for Transportation Modelling, Optimization, and Systems (ATMOS'13), Sophia Antipolis, France, 5 September 2013*, pp. 55–67. DOI: 10.4230/OASIcs.ATMOS.2013.55.

21. Lazarev A.A., Sadykov R.R. (2014) Management problem of railway cars fleet. Proceedings of the *XII All-Russian Meeting on Management Issues (VSPU 2014), Moscow, IPU RAS, Russia, 16–19 June 2014*, pp. 5083–5093 (in Russian).

22. Desaulniers J., Desrosiers J., Solomon M. (2005) *Column generation*. New York: Springer.

23. Sadykov R., Vanderbeck F. (2013) Column generation for extended formulations. *EURO Journal on Computational Optimization*, vol. 1, no 1–2, pp. 81–115. DOI: 10.1007/s13675-013-0009-9.

24. Ahuja R., Magnanti T., Orlin J. (1993) *Network flows: Theory, algorithms, and applications*. Prentice Hall.

25. Williamson D. (2019) *Network flow algorithms*. Cambridge: Cambridge University Press. DOI: 10.1017/9781316888568.

26. Evans J.R., Minieka E. (1992) *Optimization algorithms for networks and graphs*. New York: Marcel Dekker.

About the authors

Fedor A. Belousov

Cand. Sci. (Econ.);

Researcher, Laboratory of Dynamic Models of Economy and Optimization,
Central Economics and Mathematics Institute, Russian Academy of Sciences,
47, Nakhimovsky Prospect, Moscow 117418, Russia;

E-mail: sky_tt@list.ru

ORCID: 0000-0002-3040-3148

Ivan V. Nevolin

Cand. Sci. (Econ.);

Leading Researcher, Laboratory of Experimental Economics,
Central Economics and Mathematics Institute, Russian Academy of Sciences,
47, Nakhimovsky Prospect, Moscow 117418, Russia;

E-mail: i.nevolin@cemi.rssi.ru

Nerses K. Khachatryan

Cand. Sci. (Phys.-Math.);

Senior Researcher, Laboratory of Dynamic Models of Economy and Optimization,
Central Economics and Mathematics Institute, Russian Academy of Sciences,
47, Nakhimovsky Prospect, Moscow 117418, Russia;

E-mail: nerses@cemi.rssi.ru

ORCID: 0000-0003-2495-5736

DOI: 10.17323/2587-814X.2020.2.36.47

The advantages of cognitive approach for enterprise management in modern conditions

Robert A. Karayev
E-mail: karayevr@rambler.ru

Natella Yu. Sadikhova
E-mail: natella5@rambler.ru

Institute of Control Systems, National Academy of Sciences of Azerbaijan
Address: 9, B. Vahabzade Street, Baku AZ 1141, Azerbaijan

Abstract

The paper provides a brief description of cognitive management, which opens up unique opportunities for the effective management of enterprises in modern complex and unstable conditions. The problems of commercializing this promising paradigm are discussed. It is pointed out that the main, critical one of these problems is due to the lack of developed engineering of cognitive management. A conceptual framework for solving this problem is proposed, based on the convergence of the ideas and methods of the "cognitive school" and the empirical experience gained in knowledge engineering. The results of using the conceptual framework in four research projects of different industry orientations, with different internal conditions and different dynamics of the external environment are presented. The engineering prospects of the proposed framework are discussed in terms of the commercialization of the cognitive school identified by H. Mintzberg, B. Ahlstrand and D. Lampel 30 years ago.

Key words: cognitive management; conceptual framework; analysis and choice technology; cognitive approach.

Citation: Karayev R.A., Sadikhova N.Yu. (2020) The advantages of cognitive approach for enterprise management in modern conditions. *Business Informatics*, vol. 14, no 2, pp. 36–47.
DOI: 10.17323/2587-814X.2020.2.36.47

Introduction

Research into strategic management is usually said to have begun in the mid-1960s. Sometimes 1951 is cited [1]. But studies on military strategic construction appeared much earlier: for instance, Sun Tzu's famous treatise on the art of war is attributed to the 5th century BC. The range of studies on strategic management began to expand rapidly in the early 1980s, when business leaders faced the growing complexity of the business world and realized the need for strategic analysis of the prospects for their business success.

In recent years, business has come up against new challenges that require a substantial revision of established concepts and traditional strategic management tools [2]. One of the most serious problems that managers have been facing recently is understanding complex causal chains that determine the impact of the external and internal conditions of an enterprise on the goals and properties of the strategy being developed. Today, this problem is compounded by the growing complexity and instability of the economic environment, leading to numerous uncertainties and risks.

In the new environment, the use of traditional strategic management tools encounters serious limitations everywhere, dictating the need to create new tools that are appropriate to the research nature of modern management practice [3]. Ideas and methods of the cognitive approach open up great opportunities for creating such tools [4, 5]. Cognitive approach-based support tools can be regarded as tools of a new generation forming within the framework of the research paradigm of the cognitive science [6]. This paradigm today is focused on solving a wide range of complex problems of the modern world in a variety of fields: economics, sociology, politics, ecology, industry, business, emergency situations, etc., including strategic management.

There are many studies devoted to cognitive management. However, most of them have been conducted by management theorists, mathematicians or psychologists, and they are very far from the demands of management practice. More than thirty years have passed since the birth of the cognitive school of management [7]. Numerous publications widely discuss the tempting and promising prospects of this school, however, to date, there is still no developed cognitive management engineering, which (as in the case of all other knowledge-based technologies) is a critical link in cognitive analytics [8].

It is significant that of all the publications on cognitive management we have analyzed (academic journals, monographs, conference proceedings) we have not come across a single one written by a specialist with experience in the practical development of commercial models. Most of those publications analyze the problems of the cognitive process and discuss theoretical problems for future research.

At the same time, long-term practice convincingly indicates that an important prerequisite for creating the engineering of all knowledge-based technologies without exception [8] is critical analysis and generalization of the results of applied developments that form the empirical "axiomatics" of the problem area, which is no less important than numerous theoretical constructions.

Even thirty years ago, the authors of [7] pointed out that the body of work of cognitive school followers "forms not so much a tight school of thought as a loose collection of research, which seems, nonetheless, to be growing into such a school. If it can deliver on its intentions, it could very well transform the teaching and practice of strategy as we know it today" [7].

In our opinion, the solution to this complex "stagnant" problem can be facilitated by the creation of advanced cognitive management engineering, which includes two basic components: a single conceptual framework that combines the entire variety of existing strategic concepts and a systematic library of cognitive tools that allow these concepts to be implemented in concrete strategic projects.

Apparently, the possibly most effective way to solve this problem is a convergence of ideas and methods of the cognitive school of strategic management and the empirical experience gained in knowledge engineering [9]. In this paper, we propose one of the possible versions of the conceptual framework constructed in this way, presenting the results of the development and testing of cognitive models for four research projects differing in industry orientation, internal conditions and different dynamics of the macroeconomic environment obtained within this framework. The goal in these projects was not to create commercial support tools. The main goal pursued by the authors of the projects was to test the possibility of using a single conceptual framework when creating applied models for enterprises of various configurations. The developments were of a research nature and were carried out from 2004 to 2018 by four independent groups.

In the following paragraphs, we provide:

(1) a general description of the cognitive approach to the problem of strategic management modeling reflecting the characteristics of enterprise management in an unstable economic environment;

(2) the proposed conceptual framework of cognitive management, integrating modern cognitive experience in a single ontological project;

(3) a brief description of the cognitive models developed during the implementation of these projects;

(4) some most significant, in our opinion, results obtained by us in the development and testing of models, which, taking into account the specifics of concrete projects, can nevertheless be of interest to a wide range of specialists concerned with engineering issues.

The findings presented in the conclusion reflect not only the opinions of the model makers, but also the opinions of the project stakeholders, as well as the experience in developing cognitive models in other areas.

1. General description of the cognitive approach

Modern ideas of the cognitive approach to the management of complex problem situations have been described in numerous publications, e.g. [4, 5, 10]. Within the framework of these ideas, cognitive modeling is a cognitive map-based modeling aimed at studying a problem situation and finding the optimal (in one sense or another) strategy for managing this situation. The main components of cognitive models are the cognitive map and methods of its analysis.

1.1. Cognitive map

A cognitive map is an explicit representation of "mental models" [11] of management subjects about the conceptual structure, laws or patterns of a problem situation.

The main components of a cognitive map are:

(1) a set of basic factors characterizing the problem situation (the management object and its external environment);

(2) cause-and-effect (causal) relations between basic factors represented by certain formalisms.

Currently, the formalism of digraphs is used to build cognitive maps in most cases [12]. The vertices of the digraph are the factors of the problem situation, and the arcs are the causal relations between them.

On the set of basic factors, we set:

✦ a subset of target factors reflecting the desired state of the management object,

✦ subsets of uncontrollable and controllable factors of the management object,

✦ a subset of external environment factors that may affect the problem situation.

For each of the factors, its current value or the tendency of its variation is established. For causal relations, the nature of the impact of the cause factors on the effect factors is established.

Situation management consists in such a change in the controllable factors of the management object that would lead to the desired changes in the target factors.

It should be noted that digraph-based maps are currently the most popular in cognitive research, however the range of formalisms that can be used in the building of cognitive maps is much wider. Along with the digraph formalism, cognitive maps can also employ such formalisms as M. Minsky's frames, genetic networks, relational matrices of the American Society for Quality, scenario networks, etc.

1.2. Cognitive map analysis methods

Cognitive map analysis methods are developed to conduct model experiments on a cognitive map in order to solve a wide range of management problems, e.g. choosing the strategic goals of an enterprise, strategic diagnostics of the external and internal environment of an enterprise, developing an "optimal" (in one sense or another) enterprise strategy in an uncertain and dynamic economic environment, auditing a strategy, assessing its stability and effectiveness, adjusting strategy in a changing environment, etc.

2. Conceptual framework of cognitive management

The problem of finding the best strategy in today's business environments is extremely complex. It can be fully regarded among the class of complex problems, the solution of which is beyond the competence of the traditional theory of strategic management.

The "phenomenon of complexity" is due to five major features of modern strategic management, which management practice faces everywhere nowadays. Such features are: the uniqueness of each strategic project, multifactoriality, multidimensionality (multidisciplinarity), dynamism and uncertainty of the problem of strategic choice, the substantial role of strategy developers' mentality, as well as strategic decision makers.

There have been known attempts to create one-size-fits-all cognitive support tools for strategic problem solvers [13–16]. However, practice shows that in the complex and diverse environment in which modern enterprises operate, the creation of universal tools is futile and does not justify itself. What is needed is not universal tools but some unified methodology that allows building cognitive models for concrete enterprises for a concrete period of time, taking into account the strategic vision of the owners and managers of this enterprise [17].

The first step in the formation of this kind of methodology is the development of a conceptual framework of cognitive management that is common to all its applications. The fundamental fact here is that a conceptual framework in such a complex problematic environment cannot be developed on the basis of traditional formal axiomatic approaches. The empirical approach widely used in knowledge-based technologies [18] is apparently more appropriate here.

Within this approach, the conceptual framework of cognitive management can be represented as follows:

$$P(CM)\text{: } S^{o}(C) \Rightarrow S^{c}(C)\,|_{U(P)}$$

where $P(CM)$ is the full knowledge of the problem area of cognitive management;

$S^{o}(C)$ is the current state of the analyzed business situation set on the cognitive map;

$S^{c}(C)$ is the target state of the analyzed business situation set on the cognitive map;

$U(P)$ is the management strategy establishing the sequence of strategic steps that ensure the transition of the business situation from So to Sc.

It is clear that the full knowledge $P(CM)$ should reflect the accumulated theoretical and practical experience of the problem area.

The study and critical analysis of the vast material devoted to cognitive modeling of management problems [19–26], as well as the authors' personal experience suggest that the full knowledge P(CM) can be represented as an ontological project including the following sections:

1. Applied problems that can be solved with the help of cognitive management engineering. The problems can be presented in the form of separate questions to which answers can be obtained and, thus, they are the subject of engineering.

2. A set of postulates or axioms that show what assumptions were made during the development of engineering. The axioms or postulates describe the conditions and limits of applicability of engineering.

3. The list of concepts of strategic management that are currently in place in management science and practice.

4. A systematized library of applied tools to implement these concepts in specific strategic projects.

4.1 Support tools for choosing the concept of strategic enterprise management.

4.2. Support tools for choosing the language of presentation (formalism) of cognitive maps.

4.3. Support tools for structural and functional identification and parameterization of cognitive maps.

4.5. Many methods of analysis of cognitive maps for solving applied problems.

The structure of P(CM) presented is essentially a "protoframe" of engineering and solves two important problems.

First, it expands the traditional idea of schemes for solving applied problems of strategic management by including the stages of postulating cognitive analysis, choosing the concept of strategic management, choosing an appropriate language for representing cognitive maps, carrying out structural and functional identification and parameterization of cognitive maps. Without addressing these issues, cognitive business analysis in most cases loses all practical meaning.

Second, it systematizes the directions of efforts aimed at the development and accumulation of applied capabilities of cognitive analysis in solving issues that are important in the context of management practice, including but not limited to:

✦ identification of contradictions between the goals set by the management subjects;

✦ analysis of the effectiveness of the controllable factors of the cognitive map and their importance according to the degree of impact on the established goals;

✦ designing various options for management strategies ("self-development strategy" and various versions of "controlled development strategies");

✦ modeling the dynamics of alternative management strategies in various scenarios of the development of the external environment and choosing the optimal (in one sense or another) strategy;

✦ studying the stability of the selected strategy in critical situations caused by possible external environment threats;

✦ monitoring the strategy in the process of its implementation;

✦ retrospective analysis of the adequacy of the cognitive map and its adjustment.

The production part of the ontological project today includes a lot of publications related to solving the main problem of strategic management — choosing the enterprise development strategy.

A preliminary systematization of these studies can be done using descriptors characterizing their engineering efficiency:

1. the horizon of analysis, which the tool is focused on (short-term, medium-term, long-term);

2. the nature of the external environment of enterprises for which the tool is developed (static, dynamic);

3. the stage of development of the strategy for which the tool is intended (strategic identification of the external and internal environment of the enterprise, conceptualization of the strategic vision of the owners and top management of the enterprise, formalization of the strategic vision in the form of a cognitive map, development (selection) of map analysis methods, testing of the cognitive model [27]);

4. the level of development of the tool (theoretical proposal, demonstration prototype, research prototype, functional prototype, industrial version [9]).

We use this systematization to characterize the cognitive models developed during the implementation of the four mentioned projects.

3. Cognitive analysis projects

Project 1. An assessment of the strategic prospects of an offshore oil company in the Caspian Sea region, the operating conditions of which are characterized by typical modern trends: fluctuations in world oil prices caused by geopolitical factors, the global financial crisis, growing government regulation, tightening environmental standards. The study was carried out jointly with the Petroleum Engineering Department of the Texas A&M University (College Station, Texas, USA) under a creative cooperation agreement with BP Azerbaijan.

Project 2. Identification of key competencies for the strategic development of a telecommunications company – a regional dealer of Microsoft, Cisco Systems, HP, Intel. The company is engaged in the development, installation and maintenance of industrial, administrative and medical information systems, as well as IP and CTI telephony systems for telecom service providers and corporate clients.

Project 3. Analysis of management effectiveness at a poultry enterprise of a holding company in Baku. The problem was caused by the critical situation due to an increase in prices for raw materials and feeds, a decrease in prices for finished products, an increase in competitive pressure, difficulties in obtaining commercial loans, insufficient qualifications of managerial personnel, as well as a decrease in the share of profits allocated to the refinancing of the enterprise (in particular, due to the corruptive pressure of the regulatory authorities).

Project 4. Selection of regional ICT management strategy. The project was implemented with financial support from the Ministry of Communications and Information Technologies of Azerbaijan. Its relevance was due to the fact that the transition of the country to the information (digital) economy made the issue of the impact of information and communication technologies (ICT) on the economic growth of various economic entities vital. The issue was most acute at the regional level. This problem is quite complicated. Until now, it has been the subject of numerous discussions and has been known in economic practice for over thirty years as R. Solow's paradox ("the productivity paradox") [28].

The problem is multifactorial, uncertain and dynamic and cannot be solved using traditional econometric methods. Those interested in this issue can browse the Internet to see how many unsuccessful attempts to tackle it have been made in international practice. In particular, the authors of such attempts used such methods as correlation and regression analysis, estimation methods based on the Cobb–Douglas production function, research based on the methodology of economic impact analysis. Mention may be made of the iSociety project supported by Microsoft and Pricewaterhouse-Coopers that examined the impact of ICT on the efficiency of everyday business in the UK, as well as the McKinsey study that analyzed the channels of the impact of ICT on productivity based on the study of relevant business processes.

In our case, a cognitive approach was used to solve the problem posed [29]. The results of a large-scale Economist Intelligence Unit study (EIU, 2004), which is a combination of system analysis methodology and business review, were used as a conceptual basis at the stage of strategic concept selection.

Model experiments conducted during the implementation of this project have shown that cognitive modeling opens up fundamentally new opportunities not only for assessing the economic impact of ICT, but also for managing this impact, i.e. for carrying out an effective ICT management strategy.

4. Brief description of the developed cognitive models

The main characteristics of the models developed in the above projects were:

✦ the adopted concept of strategic management;

✦ the cognitive map formalism reflecting the logic of the adopted concept;

✦ methods for building and analyzing a cognitive map;

✦ scales for assessing the values of basic factors and causal relations of the map.

The characteristics of the models developed in these projects are shown in *Table 1.*

Examples of building cognitive maps can be found in [29, 30].

5. Discussion

The experience of developing and testing cognitive models in these projects has shown the following:

1. In practical business problems, cognitive modeling faces the same difficulties as other intelligent (knowledge-based) technologies. This is an issue of high-quality expert knowledge used in building cognitive models, all kinds of "traps" and "expert paradoxes" [9] that arise when working with experts, difficulties in choosing an appropriate detail level for cognitive maps, and an urgent need for an intermediary specialist (knowledge engineer, meta-interpreter cognitologist), etc.

2. Cognitive modeling acquires practical significance only within the framework of the strategic diagnostic methodology, which includes the stages of macroeconomic and marketing analysis in the context of the external environment dynamics. Ignoring these stages reduces cognitive models to mathematical objects that are far removed from the realities of business practice.

Table 1.

Characteristics of the cognitive models developed during the implementation of the projects

Project No. (project development level)	Adopted concept of business strategy	Structural type of CM	CM building and analysis methods	Scales for assessing factors and trends
Project 1. (DP)	"Resource–oriented" (R. Grant, 1987)	Digraph	***Building methods:*** SWOT analysis. PESTLE analysis. Implicit repertoire lattices of S. Hinckle. Interaction matrix (L. Jones, 1982) ***Analysis methods:*** Models of linear dynamics F. Roberts. Scenario analysis method.	5–point bipolar linguistic
Project 2. (RP)	"School of competencies" (C. Prahalad, G. Hamel, 1990)	Relational Matrices (ASQ Matrix XL Diagram) [8]	***Building methods:*** Ishikawa diagrams. VRIO analysis (J. Barney, 1987). eTOM.4.0 business process model (Intern. Telecommunication Union/Stand–ardization Sector). "Consumer opinion model" (IBM, USA). "Resource model" (American Society for Quality, USA). ***Analysis methods:*** Fuzzy causal grid analysis [8]. T. Saati hierarchy analysis method.	10–point linguistic
Project 3. (RP)	"Migration strategy" [1]	Digraph	***Building methods:*** SWOT analysis. PESTLE analysis. "Competitive analysis model" (M. Porter, 2003). Methods of psychosemantics and multidimensional non–metric scaling (M. Davidson, 2003). ***Analysis methods:*** Qualitative dynamics models based on rules [5] and temporal logic (Pospelov, 1986).	5–point bipolar linguistic
Project 4 (RP)	Economist Intelligence Unit concept (EIU, 2004)	Minsky frames	***Building methods:*** System analysis + Business review PESTLE analysis ***Analysis methods:*** Scenario modeling method	10–point linguistic

Legend: DP – demonstration prototype; RP – research prototype; BM – CM building methods; AM – CM analysis methods.

3. One of the key issues of cognitive management is that of choosing a reasonable level of detail ("granulation") of cognitive maps. It stems from the fact that the use of cognitive maps with a low detail level often leads to the loss of details wherein the devil is known to be. A critical review of relevant literature and discussions with colleagues has led us to conclude that the issue requires further study and can be regarded as one of the areas for further improvement of cognitive analysis, in particular, the development of an effective "multiscaling" mechanism similar to the design proposed by the RAND Corporation [31].

4. Serious difficulties arise in the scenario analysis of the strategies developed. Non-monotonic dynamics of the external environment and the need for a multivariate analysis of strategies in a broad "scenario corridor" require the development of an engineering modeling technique that is different from that offered by higher mathematics [19, 22].

5. Collaboration with employees of enterprises has shown that they did not have a clear opinion on the effectiveness of cognitive modeling. At the same time, 12 out of 17 respondents showed active interest and felt that cognitive modeling is a promising and useful technology. At the same time, our observations have revealed a very important latent feature of cognitive analysis: it stimulates the cognitive and creative activity of strategy developers in the most complex and critical phase of "strategic thinking" – the phase of afferent synthesis of strategic decisions.

6. The work in the projects has convincingly confirmed the fact that the suitability of cognitive models depends primarily on the "quality" of knowledge in its foundation, the carriers of which are the strategy developers, their competencies and professional experience. Cognitive modeling only enhances and expands their analytical potential.

7. The high changeability of the business environment today imposes stringent requirements on the timeframe of constructing and testing cognitive models. This is the fundamental difference between cognitive management and many other cognitive applications. In this regard, it becomes critically important to create effective support tools (high-level languages, system shells, scenario networks) for the "quick" development and testing of cognitive models – an issue that is currently completely dismissed by theorists of the cognitive school.

Conclusion

Note some general considerations regarding the advantages and shortcomings of cognitive modeling, which, according to the analysis of the few applied studies, are relevant for other applications of the cognitive approach.

1. As in all "knowledge-based" technologies, the necessary condition for success is, first, the experts involved (top managers, business consultants) possessing high-quality knowledge and, second, the involvement of a knowledge engineer playing a key role in the initial stages of the cognitive process.

2. The fundamental advantages of the cognitive approach are:

- the capability to study the dynamics of a business situation in a complex, rapidly changing PEST environment, when the available data is not enough to build a complete simulation model;
- the capability to study business situations, taking into account the multi-factorial "institutional shell" of the enterprise [17];
- the capability to study business situations in the presence of multiplicative and feedback connections between environmental factors, as well as in the presence of various threshold effects.

None of the many traditional strategic management support tools have such capabilities.

3. Cognitive analysis opens up new perspectives for decision theory. The possibility of purposeful generation of optimal strategies, which is not available in well-known commercial DSS packages, determines a fundamentally new approach to decision-making: not "choosing the best solution from the many available alternatives" (RAND Corporation paradigm), but purposefully generating the "best solution".

4. Testing of cognitive models has revealed a number of challenges:

- due to the discrete structure of models, only a rough approximation of continuous processes is possible. One has to carefully consider the sequence of factors in causal networks and take into account whether the effects of some factors on others are in sync, or whether they are offset from each other;
- one has to be careful when parameterizing cognitive maps, especially when evaluating the strength of causal relations that can change during scenario transformations;
- parameterization of cognitive models in the case of complex (multi-factorial) cognitive maps faces an "integrity issue." The widespread belief that the values of each factor and each causal relation can be determined individually is, in our opinion, deeply erroneous. According to our observations, although these assessments are made individually, they are highly correlated with the gestalt of the problem situation formed by strategy developers on an intuitive level [32]. This fact severely limits the working formats of cognitive maps and makes it unproductive to deal with popular large-format maps, in which basic factors and causal relations number in tens or even hundreds.

5. The complex economic environment in which enterprises operate today significantly limits the possibilities of traditional economic and mathematical modeling. In the problems of strategic choice, knowledge-based modeling becomes relevant. Modern strategic research virtually turns into a complex engineering art [18], shaping a new promising trend of managerial business analytics. Critical analysis shows that cognitive management is currently the only uncontested paradigm that can ensure the successful implementation of this trend [33]. However, it also shows that the field of cognitive management engineering today is riddled with issues that must be solved lest the cognitive school of management, like thirty years ago, remain a tempting and promising potential rather than a practical tool. ■

References

1. Newman W. (1951) *Management action: Organization and management technique*. Columbian University (in Russian).
2. Thompson Jr. A.A., Striklend III A.G. (2009) *Strategic management: concepts and situations for the analysis*. 12 ed. Moscow: Williams (in Russian).
3. Drucker R. (2012) *Management challenges for the 21st century*. London, New York: Routledge. Taylor & Francis Group.
4. Narayanan V.K., Zane L.K., Kemmerer B. (2011) The cognitive perspective in strategy: An integrative review. *Journal of Management*, vol. 37, no 1, pp. 305–323. DOI: 10.1177/0149206310383986.

5. Hodginson G. (2011) Cognitive process in strategic management: Some emerging trends and future direction. *Handbook of industrial, work & organizational psychology*. London: SAGE Publication, vol. 2, pp. 401–441.
6. Ross D. (2005) *Economic theory and cognitive science: Microexplanation*. MIT Press.
7. Mintzberg H., Lampel J., Ahlstrand B. (2005) *Strategy safari: A guide tour through the wilds of strategic management*. New York: Free Press.
8. Lee J.D., Kirlik A., Dainoff M.J., eds. (2013) *The Oxford handbook of cognitive engineering. Oxford library of psychology*. New York: OUP USA.
9. Waterman D.A. (1986) *Guide on expert systems*. Addison-Wesley.
10. Hodgkinson G.P., Healey M.P. (2007) Cognition in organizations. *Annual Review of sychology*, no 59, pp. 387–417. DOI: 10.1146/annurev.psych.59.103006.093612.
11. Johnson-Laird P.N. (1980) Mental models in cognitive science. *Cognitive Science*, no 4, pp. 71–115. DOI: 10.1207/s15516709cog0401_4.
12. Roberts F. (1976) *Discrete mathematical models with application to social, biological and environmental problems*. Englewood Cliffs, New Jersey: Rutgers University, Prentice-Hall.
13. Schwenk C.R. (1988) The cognitive perspective on strategic decision making. *Journal of Management Studies*, vol. 25, no 1, pp. 41–55. DOI: 10.1111/j.1467-6486.1988.tb00021.x.
14. Haleblian J., Rajagopalan N. (2006) A cognitive model of CEO dismissal: Understanding the influence of board perceptions, attributions and efficacy beliefs. *Journal of Management Studies*, no 43, pp. 1009–1026. 10.1111/j.1467-6486.2006.00627.x.
15. Porac J.F., Thomas H. (2002) Managing cognition and strategy: Issues, trends and future directions. *Handbook of strategy and management* (Eds. A. Pettigrew, H. Thomas, R. Whittington). London: SAGE Publisher, pp. 165–181.
16. Eden C., Ackermann F. (2001) SODA – The principles. *Rational analysis for a problematic world revisited: problem structuring methods for complexity, uncertainty and conflict* (Eds. J. Rosenhead, J. Mingers). Chichester: Wiley, pp. 21–41.
17. Balatsky E. (2006) The dialectic of cognition and the new paradigm of economic science. *World Economy and International Relations*, no 7 (in Russian).
18. Pospelov D.A. (2001) *Theory and practice of situational management*. Moscow: Nauka (in Russian).
19. *IEEE Proceedings of the International Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA) (2011–2019) series*. Available at: https://edas.info/web/cogsima2019/home.html (accessed 20 February 2020).
20. *Best value business & consulting toolkits. Created by management consultants previously from*: Deloitte, McKinsey&Company, BCG Strategy Consultants. Available at: https://www.slidebooks.com/products/strategy-toolkit?gclid=EAIaIQobChMIhfSMj8qz4gIVgobVCh1xTwW2E AMYASAAEgJ8XfD_BwE&variant=17648810693 (accessed 20 February 2020).
21. Gallopin G.C. (1977) Modeling incompletely specified complex systems. *Third International Symposium on Trends in Mathematical Modelling, S.C. Bariloche, December 1976*, UNESCO-Fundacion Bariloche.
22. *IEEE Proceedings of the International Conference on Cognitive Modeling (ICCM) series* (1996–2019). Available at: https://iccm-conference.github.io/previous.html (accessed 20 February 2020).

23. Avdeeva Z.K., Kovriga S.V. (2011) *Formation of a strategy for the development of socio-economic objects based on cognitive maps*. Saarbrucken: LAP LAMBERT Academic Publishing.

24. Cummins S., Wilson D., Angwin D, Bilton C., Brocklesby J., Doyle P., Galliers B., Legge K., McGee J., Newell S., Pettigrew A., Smith C., Wensley R. (2003) *Images of strategy*. Oxford: Blackwell Publishing.

25. Rodin D.V. (2008) Conceptual approaches to the formation and implementation of organizational strategies. *System Management*, issue 1(2). Available at: http://sisupr.mrsu.ru/wp-content/uploads/2014/11/21-rodin.pdf (accessed 20 February 2020) (in Russian).

26. Aithenhead A.M., Slack J.M., Eds. (1994) *Issues in cognitive modeling*. Lea: Hove.

27. Buchanan B., Bechtal R., Bennett J., Clancey W., Kulikowski C., Mitchell T., Waterman D.A. (1983) Constructing an expert system. *Building expert system* (F. Hayes-Rath, D. Waterman, D. Lenet, eds.). Addison-Wesley, 1983.

28. Solow R. (1987) We'd better watch out. Book review. *New York Times*, 12 July 1987.

29. Karayev R.A. (2013) Choice of a strategy of regional ICT-management. Cognitive paradigm. *International Journal of Managing Information Technology*, vol. 5, no 3, pp. 17–30. DOI: 10.5121/ijmit.2013.5303.

30. Karayev R.A. (2015) Cognitive approach and its application to the modeling of strategic management of enterprises. *Knowledge engineering: Principles, methods and applications* (Ed. A. Perez Gama). New York: Nova Science, pp. 79–101.

31. Davis P.K., Kahan J.P. (2007) *Theory and methods for supporting high-level decision making*. Santa Monica, CA: RAND Corporation, TR-422-AF.

32. Bays P.M., Husain M. (2008) Dynamic shifts of limited working memory resources in human vision. *Science*, vol. 321, no 5890, pp. 851–854. DOI: 10.1126/science.1158023.

33. Polterovich V.M. (1999) Institutional traps and economic reforms. *Economics and Mathematical Methods*, vol. 35, no 2, pp. 3–20 (in Russian).

About the authors

Robert A. Karayev

Dr. Sci. (Tech.);

Professor, Head of Ecosystems Modeling Laboratory,
Institute of Control Systems, Azerbaijan National Academy of Sciences,
9, B. Vahabzade Street, Baku AZ1141, Azerbaijan;

E-mail: karayevr@rambler.ru

Natella Yu. Sadikhova

Researcher, Institute of Control Systems, Azerbaijan National Academy of Sciences,
9, B. Vahabzade Street, Baku AZ 1141, Azerbaijan

E-mail: natella5@rambler.ru

DOI: 10.17323/2587-814X.2020.2.48.63

Multidimensional log-normal distribution in real estate appraisals

Michael B. Laskin
E-mail: laskinmb@yahoo.com

St. Petersburg Institute for Informatics and Automation, Russian Academy of Sciences
Address: 39, 14 Line, St. Petersburg 199178, Russia

Abstract

The purpose of the research was to develop a market value appraisal methodology based on a set of a joint logarithmically normal distribution of price-forming factors. Joint logarithmically normal distribution means random vector component logarithms are distributed together jointly normally. This article suggests a method for appraising the real estate market value based on the statistical hypothesis of a joint logarithmically normal distribution and conditional distribution of prices with fixed values of pricing factors. The article suggests a method of offer price analysis from the point of view of its relevance to pricing factor values. We consider the features of the coefficient of development depending on the area of the land plot. Additional arguments are given in favor of estimating market value as a mode of conditional laws of price distribution. An example of a multidimensional log-normal distribution of prices and pricing factors such as the area of the improvements (improvements mean buildings and constructions) area and the land area in real data, i.e. for the case of a three-dimensional random vector. We present a formula for determining the absolute maximum density point of a multidimensional logarithmically normal random vector. The proof is given in the Appendix. The results obtained can be used to create information systems to support decision-making in valuation activities for real estate properties.

Key words: market value; logarithmically normal law of price distribution; multidimensional logarithmically normal distribution, valuation of real estate.

Citation: Laskin M.B. (2020) Multidimensional log-normal distribution in real estate appraisals. *Business Informatics*, vol. 14, no 2, pp. 48–63. DOI: 10.17323/2587-814X.2020.2.48.63

Introduction

One of the most common methods in market value real estate appraisals is the linear regression model of prices with some price-forming factors as regressors. The factors can be qualitative (type of home, encumbrance, floors, window view, the condition of the apartment/room, etc.) and quantitative (area of the object or the land plot, distance to the city center, to the metro, to other infrastructure objects, etc.).

There are various views on division the pricing factors into classes. In the context of this article, we mean splitting into qualitative and quantitative factors in terms of the possibility of representing the values of the factor as a real number (if such a possibility exists, then the factor is quantitative). Combining quantitative and qualitative factors in a single regression model presents a certain difficulty for analysts. However, this problem goes beyond the scope of the present article: here we will limit ourselves only to quantitative (real) factors. Very often such factors, considered as random variables on a set of objects of comparison, can usually be approximated by a logarithmically normal distribution. There are reasons to assume that the prices of properties formed by sequential comparisons follow the logarithmically normal distribution law.

The theoretical reason for the formation of a lognormal general population for prices formed by successive comparisons was given in [1]. The fact of subordination of rental rates in real estate to the logarithmically normal distribution was pointed out by Aitchinson and Brown in 1963 [2]. More recent researchers have also pointed to the logarithmic distribution of prices in real estate [3]. This approach is not yet traditional from the point of view of the existing practice of real estate valuation, since it requires the use of special applied statistical packages that are not used by practicing appraisers, who use a small number of objects for comparison. At the same time, the changing information environment encourages researchers to look for new, non-traditional approaches to real estate valuation. As an example, we can cite the works [4–9] devoted to the method of hedonistic pricing, i.e. the identification of a statistical relationship between the average or median cost of housing, internal and external price-forming factors.

Statistical dependence is usually estimated using models of linear, logarithmic, or partially logarithmic dependence. In general, this same ideology is the basis for the report on cadastral value [10] made by the St Petersburg government department “Cadastral assessment” in 2018. A number of works use non-regression models for estimating residential real estate objects: for example, in [11, 12] neural networks are used to predict the value of residential property, in [13, 14] machine learning methods (random forest, support vector method) are used, and in [15] the results of using such methods as decision trees, naive Bayesian classifier, and AdaBoost are compared. These methods require the use of large data samples. Another approach is to use price indices. For example, the Case-Shiller housing price index is considered in [16]. Articles [17–19] study the re-sale index, which predicts changes in the value of a resold property based on the difference in time and changes in its attributes between the initial sale and subsequent resale. The authors of [20–23] consider a hybrid method that combines a hedonistic approach and a method of re-selling.

The main approach to the study of price bubbles is to use variations of auto regression methods applied to average prices, for example, in [24–28]. Thus, the use of multidimensional logarithmically normal distributions is also in line with current trends in the search for non-traditional methods in real estate valuation.

1. Estimation of market value based on conditional distributions with fixed values of price-forming quantitative factors

Let V – be the price of the offer (or transaction), X_1, ..., X_n – are the quantitative (real) price – forming factors. Let $W = \ln(V)$, $Y_i = \ln(X_i)$, $i = 1, n$ (then $v = e^W$, $X_i = e^{Y_i}$).

Consider a multidimensional normal random vector (W, Y_1, ..., Y_n) with a mean vector (μ_W, μ_{Y_1}, ..., μ_{Y_n}). Let's write the covariance matrix in block form:

$$CV = \begin{pmatrix} \sigma_W^2 & cov\left(W,\vec{Y}\right) \\ cov\left(W,\vec{Y}\right)^T & COV \end{pmatrix},$$

where COV – covariance matrix of a random vector $\vec{Y} = (Y_1, ..., Y_n)$;

$cov\left(W,\vec{Y}\right)$ – vector $\left(\rho_{WY_1}\sigma_w\sigma_{Y_1}, \ldots, \rho_{WY_n}\sigma_w\sigma_{Y_n}\right)$;

$\sigma_W^2, \sigma_{Y_1}^2 \ldots, \sigma_{Y_n}^2$ – variances of random variables W, Y_1, ..., Y_n;

$\rho_{WY_1}, \ldots, \rho_{WY_n}$ – corresponding correlation coefficients.

Then, the conditional expectation of W, if $Y_1 = y_1$, ..., $Y_n = y_n$ equals to

$$E\left(W|Y_1 = y_1, \ldots, Y_n = y_n\right) =$$
$$= \mu_W + \left(COV^{-1} \times cov\left(W,\vec{Y}\right)^T, \left(\vec{Y} - \overline{\mu_Y}\right)\right),$$

where $\overline{\mu_Y} = \left(\mu_{Y_1}, \ldots, \mu_{Y_n}\right)$. Conditional variance of W, if $Y_1 = y_1$, ..., $Y_n = y_n$ is equal to

$$D\left(W|Y_1 = y_1, \ldots, Y_n = y_n\right) =$$
$$= \sigma_W^2 - \left(COV^{-1} \times cov\left(W,\vec{Y}\right)^T, cov\left(W,\vec{Y}\right)\right)$$

For fixed values of price-forming factors $X_1 = x_1$, ..., $X_n = x_n$ the most probable value of the offer price (or transaction, depending on what prices were in the source data) V is calculated using the conditional mode formula:

$$Mode(V|X_1 = x_1 = e^{y_1}, \ldots, X_n = x_n = e^{y_n}) =$$
$$\exp\left(\mu_W + (COV^{-1} \times cov(W,\vec{Y}))^T, \left(\vec{Y} - \overline{\mu_Y}\right) - \right. \quad (1)$$
$$\left. -\sigma_W^2 + \left(COV^{-1} \times cov(W,\vec{Y})^T, cov(W,\vec{Y})\right)\right).$$

Under the terms of Federal Law No 135 [29], the market value is the most probable price at which the evaluation object can be alienated on the open market in conditions of perfect competition. In practice, appraisers tend to use an average or median estimation. Such estimations can be based on conditional expectation and a conditional median:

$$E(V|X_1 = x_1 = \exp(y_1), \ldots, X_n = x_n = \exp(y_n)) =$$
$$= \exp(\mu_W + (COV^{-1} \times cov\left(W,\vec{Y}\right)^T, \left(\vec{Y} - \overline{\mu_Y}\right)) + \quad (2)$$
$$+ \frac{1}{2}\sigma_W^2 - \frac{1}{2}\left(COV^{-1} \times cov\left(W,\vec{Y}\right)^T, cov\left(W,\vec{Y}\right)\right)).$$

$$Median\left(V|X_1 = x_1 =\right.$$
$$\left.\exp(y_1), \ldots, X_n = x_n = \exp(y_n)\right) = \quad (3)$$
$$= \exp(\mu_W + COV^{-1} \times cov(W,\vec{Y})^T, (\vec{Y} - \overline{\mu_Y}).$$

Thus, if with respect to some ensemble of quantitative pricing factors and the prices of objects of comparison can be adopted as a working hypothesis on the joint log-normal distribution (joint normal distribution of the logarithms) component of a random vector, then the valuation can be accepted in the evaluation by the formula (1). Estimates according to the formulas (2) and (3) can also be taken; but it should be noted that they do not follow the definition of market value in accordance with Federal Law No 135.

Let's consider an example that uses real data collected by well-known Russian appraisers and was published on the resource [30]. The data set includes 40 real estate objects of industrial and warehouse use with a location in the same region (St Petersburg), on offer for sale in the same time period. Since the authors of the example justified the rejection of a number of adjustments, in our example we will also consider the data comparable and comparable without additional adjustments. Industrial and warehouse purpose real estate is considered as a unit complex consisting of a land plot and improvements (buildings). The data set is presented in *Table 1.*

The items compared are considered as existing industry and warehouse properties that are offered for sale in the current use. We will build a general method for estimating the market value (without auction discount), if the area

Table 1.

Source data

Building area (sq. m)	Land area (sq. m)	Offer prices (rubles)	Price to improvements square ratio (rubles per 1 sq. m of improvements)
400	2 500	20 500 000	51 250
750	5 000	18 000 000	24 000
1 081	3 378	26 000 000	24 052
1 130	6 638	27 500 000	24 336
1 320	4 167	31 500 000	23 864
1 440	10 000	160 000 000	111 111
1 790	3 462	93 000 000	51 955
1 900	13 000	85 000 000	44 737
2 125	5 623	85 000 000	40 000
2 642	5 183	75 000 000	28 388
2 700	6 800	59 000 000	21 852
1 820	2 737	32 000 000	17 582
2 250	9 252	84 000 000	37 333
2 973	5 388	90 000 000	30 272
3 513	10 000	80 000 000	22 773
3 600	5 000	95 000 000	26 389
4 000	13 558	140 000 000	35 000
4 124	12 866	91 000 000	22 066
4 167	5 000	125 000 000	29 998
4 257	6 861	128 500 000	30 186
5 292	11 143	56 000 000	10 582
5 300	16 000	220 000 000	41 509
6 011	11 319	135 000 000	22 459
6 013	20 781	90 000 000	14 968
6 060	21 790	179 000 000	29 538
6 123	2 390	152 490 000	24 904
6 479	7 337	119 000 000	18 367
6 756	4 220	90 000 000	13 321
10 000	12 000	420 000 000	42 000
10 300	17 000	312 000 000	30 291
10 672	12 194	350 000 000	32 796
10 990	30 000	480 000 000	43 676
12 000	30 000	300 000 000	25 000
13 000	55 000	200 000 000	15 385
14 428	33 000	385 000 000	26 684
15 000	37 000	840 000 000	56 000
18 924	20 600	800 000 000	42 274
22 312	40 162	338 541 000	15 173
34 082	478 000	2 500 000 000	73 353
35 000	160 000	2 400 000 000	68 571

of improvements and land are fixed (of course, at the same time period, same real estate class, and the same region).

In this case, there are random variables V – the offer price per 1 sq. m of improvements, SB – the area of improvements, SP – the area of the land plot. They form a three-dimensional random vector (V, SB, SP). Let $W = \ln(V)$, $Y = \ln(SB)$, $Z = \ln(SP)$ (then $v = e^{W}$, $SB = e^{Y}$, $SP = e^{Z}$). For a three-dimensional normal random vector (W, Y, Z) the mean vector is equal to (μ_W , μ_Y, ..., μ_Z). The covariance matrix looks like:

$$CV = \begin{pmatrix} \sigma_W^2 & \rho_{WY}\sigma_W\sigma_Y & \rho_{WZ}\sigma_W\sigma_Z \\ \rho_{YW}\sigma_W\sigma_Y & \sigma_Y^2 & \rho_{YZ}\sigma_Y\sigma_Z \\ \rho_{ZW}\sigma_W\sigma_Z & \rho_{ZY}\sigma_Y\sigma_Z & \sigma_Z^2 \end{pmatrix},$$

or:

$$CV = \begin{pmatrix} \sigma_W^2 & cov\left(W,\vec{Y}\right) \\ cov\left(W,\vec{Y}\right)^T & COV \end{pmatrix},$$

where $COV = \begin{pmatrix} \sigma_Y^2 & \rho_{YZ}\sigma_Y\sigma_Z \\ \rho_{ZY}\sigma_Y\sigma_Z & \sigma_Z^2 \end{pmatrix}$; $\vec{Y} = (Y, Z)$

$$cov\left(W,\vec{Y}\right) = \left(\rho_{WY}\sigma_W\sigma_Y,\ \rho_{WZ}\sigma_W\sigma_Z\right);$$

$\sigma_W^2, \sigma_Y^2, \sigma_Z^2$ – variances of random variables W, Y, Z;

$\rho_{WY} = \rho_{YW}, \rho_{WZ} = \rho_{ZW}, \rho_{YZ} = \rho_{ZY}$ – corresponding correlation coefficients.

Conditional expectation of W, if Y, $Z = z$:

$$E\left(W|Y = y, Z = z\right) =$$
$$= \mu_W + \left(COV^{-1} \times cov\left(X,\vec{Y}\right)^T,\ \left(y - \mu_Y, z - \mu_Z\right)\right).$$

Conditional variance of W, if Y, $Z = \mathrm{z}$:

$$D\left(W|Y = y, Z = z\right) =$$
$$= \sigma_W^2 - \left(COV^{-1} \times cov\left(X,\vec{Y}\right)^T,\ cov\left(X,\vec{Y}\right)\right).$$

Let's set the values of the area of improvements $SB = sb$ and the area of the land plot $SP = sp$. In accordance with the above notation $Y = \ln(SB)$, $Z = \ln(SP)$, $y = \ln(sb)$, $z = \ln(sp)$. The most probable value of the offer price V for known values of the area of improvements and land area is calculated using the formula [31]:

$$Mode(V|SB = sb, SP = sp) =$$
$$= \exp(\mu_W + (COV^{-1} \times cov\left(X,\vec{Y}\right)^T, (y - \mu_Y, z - \mu_Z) - \tag{4}$$
$$-\sigma_W^2 + \left(COV^{-1} \times cov\left(W,\vec{Y}\right)^T,\ cov\left(W,\vec{Y}\right)\right))\,.$$

Conditional expectation:

$$E(V|SB = sb,\ SP = E) =$$
$$\mu_W + (COV^{-1} \times cov\left(X,\vec{Y}\right)^T, (y - \mu_Y, z - \mu_Z)) + \tag{5}$$
$$+\frac{1}{2}\sigma_W^2 - \frac{1}{2}\left(COV^{-1} \times cov\left(W,\vec{Y}\right)^T,\ cov\left(W,\vec{Y}\right)\right)).$$

Conditional median:

$$Median(V|SB = sb,\ SP = sp) = \exp(\mu_W +$$
$$+ \left(COV^{-1} \times cov\left(W,\vec{Y}\right)^T,\ \left(y - \mu_Y, z - \mu_Z\right)\right)). \tag{6}$$

Before applying formulas (4)–(6) to the data in *Table 1*, let's check whether there are grounds to assume the lognormality of distributions of components of a random vector (V, SB, SP) (joint normality of logarithms of their components). The Kolmogorov–Smirnov parametric test was used to check marginal distributions. The following p-value figures are received:

V – price of 1 sq. m of improvements: with parameters meanlog = 10.3 and sdlog = 0.43 p-value is equal 0.7016;

SB – improvements area: with parameters meanlog = 8.45 and sdlog = 1.02 p-value is equal to 0.9761;

SP– plot of land area: with parameters meanlog = 9.3 and sdlog = 1.01 p-value is equal to 0.8963.

Let's check the three studied random variables for the joint normality of logarithms. To do this, we use a well-known condition of joint normality: in order for a multidimensional random vector to have a multidimensional normal distribution, it is necessary and sufficient that any linear combination of its components is distributed normally. The following procedure was implemented in the statistical package R environment:

✦ we take the logarithm from variables

✦ the resulting logarithmic values of the variables are centered and normalized, each with its own standard deviation;

✦ using the standard function R unif(3,0,1), three weight coefficients are generated, and the coefficients are normalized by their sum;

✦ a linear combination of centered and normalized logarithms is formed with random positive coefficients equal to one;

✦ the resulting linear combination is tested using the Kolmogorov-Smirnov normality test, and the test result is written as a *p*-value in the array;

✦ the procedure with a random linear combination is repeated a specified number of times, each time the *p*-value is written. Then the total *p*-value array is compared to the critical level (0.05).

Figure 1 shows a histogram in which *p*-values were obtained when the test was repeated 100 000 times.

The minimum of *p*-value is equal to 0.2867691; it is more than 0.05. The mentioned procedure of 100 000 time test repeating of random linear combinations of components of random vector (W, Y, Z) seems like a reason for keeping the joint logarithmically component distribution hypothesis as the working one.

For the logarithms of the variables "Ratio of price to area of improvements," "Area of buildings," "Area of land" specified in *Table 1*, the following values of the mean vector and covariance matrix are obtained (*Table 2*).

The means are the following: $\mu_W = 10,2993$;
$\mu_Y = 8,4469$; $\mu_Z = 9,3506$, $\sigma_W^2 = 0,2381$,
$\rho_{WY}\sigma_W\sigma_Y = \rho_{YW}\sigma_W\sigma_Y = 0,0108$;
$\rho_{WZ}\sigma_W\sigma_Z = \rho_{ZW}\sigma_W\sigma_Z = 0,1467$;
$\sigma_Y^2 = 1,0635$, $\rho_{YZ}\sigma_Y\sigma_Z = \rho_{ZY}\sigma_Y\sigma_Z = 0,8978$;
$\sigma_Z^2 = 1,2140$.

In the statistical package R, a program code was implemented that allows us to calculate the market value estimation based on the specified values of the parameters "Building (improvements) area" and "Land area" (formula (4)).

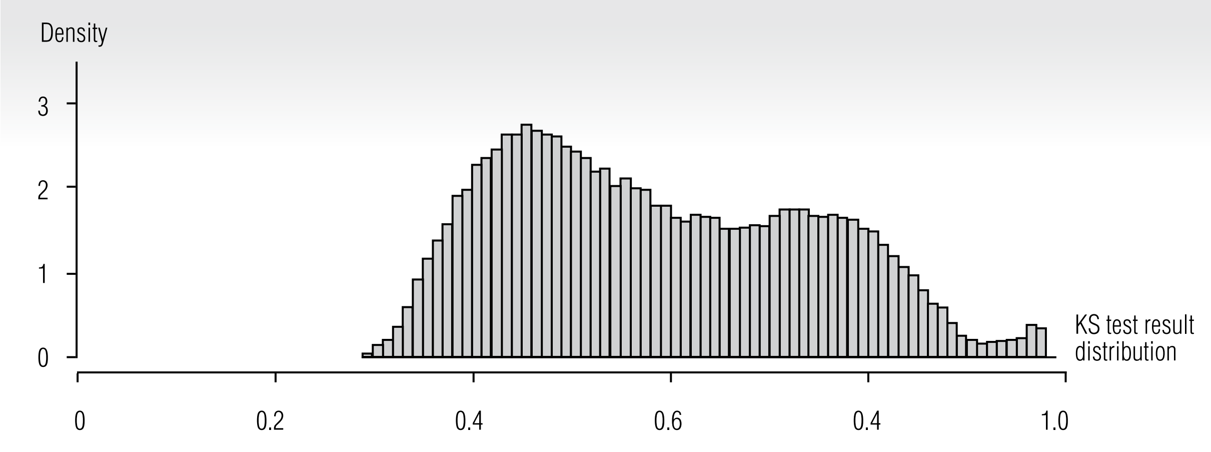


Fig. 1. Results of testing random linear combinations of centered and normalized vector components (W, Y, Z) = (ln(V), ln(SB), ln(SP)) on joint normality by the Kolmogorov–Smirnov test

Table 2.

Means of the logarithms

Ratio of price to area of improvements (*V*)	Area of buildings (*SB*)	Area of land (*SP*)
Means of logarithm		
10.2993	8.4469	9.3506
Covariance matrix		
0.2381	0.0108	0.1467
0.0108	1.0635	0.8978
0.1467	0.8978	1.2140

Similar calculations can be performed for estimates based on median values or on mathematical expectations (formulas (5) and (6)). The results are shown in *Table 3.*

This table shows the following:

✦ the mode estimate is always lower than the median estimate, and the median estimate is always lower than the mathematical expectation estimate (author's opinion: the market value should be defined as a mode estimate, in accordance with the terms of Federal Law No 135, taking into account the asymmetric distribution of prices, areas and distances observed in the market);

✦ if the area of the land plot is constant, the market value (1 sq. m of improvements) decreases as the area of improvements increases;

✦ if the area of improvements is constant, the market value (1 sq. m of improvements) increases as the land area increases;

✦ the formula (4) can be used to calculate the market value of a property of the same class as the comparison items for any values of improvement areas and land (on the same date, for the same location). Since there is no general consensus in the evaluation community regarding the numerical characteristics used for estimating market value (mode, median, mathematical expectation), formulas (5) and (6) can be applied, but, strictly speaking, these do not follow the definition of market value in accordance with Federal Law No 135.

2. The ratio of the area of improvements to land square if the price offers are fixed

In [30] the authors considered the question of pricing trends in the property market for industrial and storage purposes, and the dependence of the market value on the "density factor" (development coefficient) of the land, which is defined as the ratio of the area of improvements to the area of land. The model of joint logarithmically normal distribution of components of a random vector (*V*, *SB*, *SP*) considered in this article also allows us to look at the problem of forming price trends. The difference is that all the components of the random vector (*V*, *SB*, *SP*) are distributed on a positive half-axis; for each given value $V = v$, we can specify the most probable values of the components *SB* (improvement area) and *SP* (land area) corresponding to the offer price. In contrast to the previous case (estimation of market value based on the specified values *SB* and *SP*), the area of possible deviations from the most probable (median, average) values is not on the numeric axis, but on the plane and consists (as will be shown below) of nested sets obtained from the scattering ellipses of logarithmic values *SB* and *SP* in the inverse exponential transformation of the plane.

Let the offer price $V = v$ be known. It is necessary to estimate the ratio of the area of improvements and land for a class of objects with such an initial offer price, i.e. to select objects with lower, middle and upper price trends [30]. Denote the former: V – bid price, SB – area of improvements, SP – area of land, $W = \ln(V)$, $Y = \ln(SB)$, $Z = \ln(SP)$ (then $V = e^W$, $SB = e^Y$, $SP = e^Z$).

Table 3.

Estimates of market value per 1 sq. m of improvements for various values of improvement areas, land plots

Moda estimation		Plot of land in sq. m.									
		2 000	7 000	12 000	17 000	22 000	27 000	32 000	37 000	42 000	47 000
Improvements area in sq. m.	400	26 247	38 298	45 058	50 049	54 096	57 543	60 568	63 279	65 745	68 014
	2 400	16 938	24 714	29 076	32 297	34 909	37 133	39 085	40 835	42 426	43 890
	4 400	14 605	21 310	25 072	27 849	30 101	32 019	33 702	35 211	36 583	37 845
	6 400	13 327	19 445	22 877	25 411	27 466	29 216	30 752	32 129	33 381	34 533
	8 400	12 469	18 194	21 406	23 777	25 700	27 337	28 775	30 062	31 234	32 312
	10 400	11 835	17 269	20 317	22 567	24 392	25 947	27 311	28 533	29 645	30 668
	12 400	11 337	16 542	19 462	21 618	23 366	24 855	26 161	27 332	28 397	29 377
	14 400	10 930	15 948	18 763	20 842	22 527	23 962	25 222	26 351	27 378	28 323
	16 400	10 588	15 449	18 176	20 189	21 822	23 212	24 433	25 527	26 521	27 436
	18 400	10 294	15 021	17 672	19 629	21 217	22 569	23 755	24 818	25 786	26 675

Median estimation		Plot of land in sq. m.									
		2 000	7 000	12 000	17 000	22 000	27 000	32 000	37 000	42 000	47 000
Improvements area in sq.m.	400	31 947	46 615	54 843	60 918	65 844	70 039	73 722	77 021	80 023	82 784
	2 400	20 616	30 081	35 391	39 311	42 490	45 197	47 573	49 703	51 640	53 421
	4 400	17 777	25 938	30 517	33 897	36 638	38 972	41 021	42 858	44 528	46 064
	6 400	16 221	23 668	27 846	30 930	33 431	35 561	37 431	39 106	40 630	42 032
	8 400	15 177	22 146	26 055	28 941	31 281	33 274	35 023	36 591	38 017	39 329
	10 400	14 405	21 019	24 729	27 468	29 690	31 581	33 242	34 730	36 083	37 328
	12 400	13 799	20 134	23 688	26 312	28 440	30 252	31 843	33 268	34 564	35 757
	14 400	13 304	19 412	22 838	25 368	27 419	29 166	30 700	32 074	33 324	34 473
	16 400	12 887	18 804	22 123	24 574	26 561	28 253	29 739	31 070	32 281	33 395
	18 400	12 530	18 283	21 510	23 892	25 824	27 470	28 914	30 208	31 385	32 468

Expectation estimation		Plot of land in sq. m.									
		2 000	7 000	12 000	17 000	22 000	27 000	32 000	37 000	42 000	47 000
Improvements area in sq. m.	400	35 246	51 428	60 506	67 208	72 643	77 271	81 334	84 974	88 285	91 332
	2 400	22 744	33 187	39 045	43 370	46 877	49 864	52 485	54 835	56 971	58 937
	4 400	19 612	28 616	33 668	37 397	40 421	42 996	45 257	47 283	49 125	50 820
	6 400	17 895	26 112	30 721	34 124	36 883	39 233	41 296	43 144	44 825	46 372
	8 400	16 744	24 432	28 745	31 929	34 511	36 710	38 640	40 369	41 942	43 389
	10 400	15 893	23 189	27 283	30 305	32 755	34 842	36 674	38 315	39 809	41 182
	12 400	15 224	22 213	26 134	29 029	31 377	33 376	35 130	36 703	38 133	39 449
	14 400	14 677	21 416	25 196	27 987	30 250	32 178	33 869	35 385	36 764	38 033
	16 400	14 218	20 746	24 408	27 111	29 304	31 171	32 810	34 278	35 614	36 843
	18 400	13 824	20 170	23 731	26 359	28 491	30 306	31 899	33 327	34 626	35 821

As before, we consider a three-dimensional normal random vector (*W*, *Y*, *Z*) with a mean vector (μ_W, μ_Y, μ_Z) and covariance matrix

$$CV = \begin{pmatrix} \sigma_W^2 & \rho_{WY}\sigma_W\sigma_Y & \rho_{WZ}\sigma_W\sigma_Z \\ \rho_{YW}\sigma_W\sigma_E & \sigma_Y^2 & \rho_{YZ}\sigma_Y\sigma_Z \\ \rho_{ZW}\sigma_W\sigma_Z & \rho_{ZY}\sigma_Y\sigma_Z & \sigma_Z^2 \end{pmatrix},$$

or:

$$CV = \begin{pmatrix} \sigma_W^2 & cov\left(W,\vec{Y}\right) \\ cov\left(W,\vec{Y}\right)^T & COV \end{pmatrix},$$

where $COV = \begin{pmatrix} \sigma_Y^2 & \rho_{YZ}\sigma_Y\sigma_Z \\ \rho_{ZY}\sigma_Y\sigma_Z & \sigma_Z^2 \end{pmatrix}$;

$\vec{Y} = (Y, Z)$

$cov\left(W,\vec{Y}\right) = \left(\rho_{WY}\sigma_W\sigma_Y,\ \rho_{WZ}\sigma_W\sigma_Z\right)$;

$\sigma_W^2, \sigma_Y^2, \sigma_Z^2$ – variances of random variables *W*, *Y*, *Z*;

$\rho_{WY=}\rho_{YW}, \rho_{WZ=}\rho_{ZW}, \rho_{YZ=}\rho_{ZY}$ — corresponding correlation coefficients.

Conditional expectation of vector $\vec{Y} = (Y, Z)$ if $W = w$:

$$E\left(\vec{Y}|W = w\right) = \vec{\mu} + \frac{cov\left(W,\vec{Y}\right)^T}{\sigma_W^2}\left(w - \mu_E\right) =$$

$$\begin{pmatrix} \mu_Y \\ \mu_Z \end{pmatrix} + \begin{pmatrix} \rho_{YW}\dfrac{\sigma_Y}{\sigma_W}\left(w - \mu_W\right) \\ \rho_{ZW}\dfrac{\sigma_Z}{\sigma_W}\left(w - \mu_W\right) \end{pmatrix} = \tag{7}$$

$$\begin{pmatrix} \mu_Y + \rho_{YW}\dfrac{\sigma_E}{\sigma_W}\left(w - \mu_W\right) \\ \mu_Z + \rho_{ZW}\dfrac{\sigma_Z}{\sigma_W}\left(w - \mu_W\right) \end{pmatrix}.$$

Conditional covariance matrix if $W = w$:

$$COV\left(\vec{Y}|W = w\right) =$$

$$= COV - \frac{cov\left(W\vec{Y}\right)^T \times cov\left(W,\vec{Y}\right)}{\sigma_W^2} =$$

$$= \begin{pmatrix} \sigma_Y^2 & \rho_{YZ}\sigma_Y\sigma_Z \\ \rho_{ZY}\sigma_Y\sigma_Z & \sigma_Z^2 \end{pmatrix} -$$

$$- \begin{pmatrix} \rho_{YW}^2\sigma_Y^2 & \rho_{YW}\rho_{ZW}\sigma_Y\sigma_Z \\ \rho_{YW}\rho_{ZW}\sigma_Y\sigma_Z & \rho_{ZW}^2\sigma_Z^2 \end{pmatrix} = \tag{8}$$

$$= \begin{pmatrix} \sigma_Y^2\left(1 - \rho_{YW}^2\right) & \sigma_Y\sigma_Z\left(\rho_{YZ} - \rho_{YW}\rho_{ZW}\right) \\ \sigma_Y\sigma_Z\left(\rho_{YZ} - \rho_{YW}\rho_{ZW}\right) & \sigma_Z^2\left(1 - \rho_{ZW}^2\right) \end{pmatrix}.$$

Let *V* = *v*. In accordance with the notation introduced above *W* = ln(*V*), *w* = ln(*v*). The most probable combination of *SB* and *SP* in the condition if $V = v$:

$$Mode(\vec{Y}|V = v) =$$

$$= \exp\left(\vec{\mu} + \frac{cov\left(W,\vec{Y}\right)^T}{\sigma_W^2}\left(w - \mu_W\right) - \right.$$

$$\left. - COV + \frac{cov\left(W\vec{Y}\right)^T \times cov\left(W,\vec{Y}\right)}{\sigma_W^2}\right).$$

Table 4 shows the results: most probable combination of SB and SP in a few cases of bid prices.

It should be noted that for each value of the offer price, the most probable pair of *SB, SP* values is the only one (the building density coefficient in this case corresponds to the most probable pair *SB, SP*). Trying to present as the most convenient another pair of components of *SB* and *SP* means choosing a point when there are many other equally probable points, with a density less than the maximum, and for which the building density coefficients will obviously be different. *Figure 2* shows images of two-dimensional distributions of *SB* and *SP* for the offer price of 7.000 rubles, 28.000 rubles (from left to right), 100.000 rubles per 1 sq. m of improvements.

It is possible to see that any other points in plane *SP, SB* have any set of equal-probability points. The sets of such points shown on *Figure 3* (for *V* = 28 000 rubles per 1 sq. m of improvements).

Table 4.

Most probable combination of *SB* and *SP*, density factor in a few cases of bid prices

Bid price on 1 sq.m. of improvements	7 000	12 000	21 000	28 000	40 000	60 000	80 000	100 000
Most probable pair:								
Area of improvements	619	634	650	659	669	682	691	698
Plot of land in sq.m.	630	878	1 239	1 479	1 843	2 365	2 824	3 240
Dencity factor	0.98	0.72	0.52	0.45	0.36	0.29	0.24	0.22

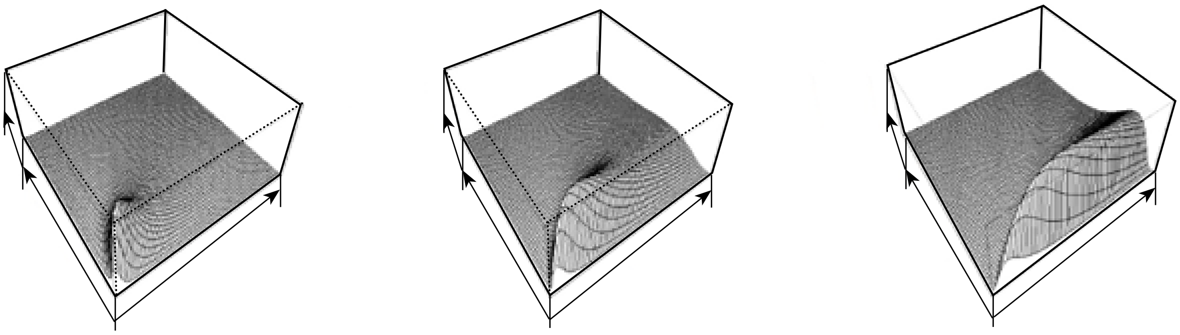


Fig. 2. Two–dimensional distributions of *SP* (improvements area)

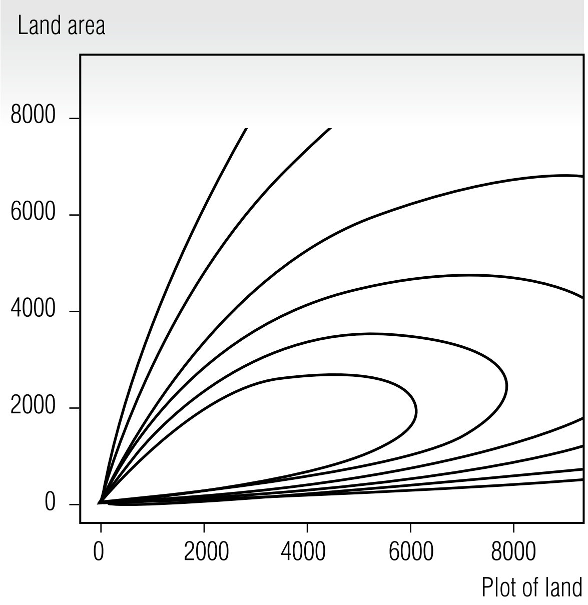


Fig. 3. Equal–probability level lines for *SP* and *SB*

3. The density factor (ratio of the improvements area to land area)

Let’s assume that the price of the offer (or transaction) is known. Our goal is to estimate what the coefficient of building density should be at a given price and the given area of the land plot. Let’s use the formulas (7) and (8). For a fixed offer price ($V = v$), formulas (7) and (8) give the calculated values of the conditional mathematical expectations of the improvement area logarithms ($SB| V = v$), the land area ($SB| V = v$), and the conditional covariance matrix. Additionally let’s assume that the area of the land plot is also known. We introduce new notation for conditional logarithms of the improvement area ($SB| V = v$) and the land area ($SB| V = v$):

$$\mu_{condSB} = \mu_Y + \rho_{YW}\frac{\sigma_Y}{\sigma_W}\left(w - \mu_W\right),$$

$$\mu_{condSP} = \mu_Z + \rho_{ZW}\frac{\sigma_Z}{\sigma_W}\left(w - \mu_W\right),$$

$$\sigma^2_{condSB} = \sigma^2_Y\left(1 - \rho^2_{YW}\right),$$

$$\sigma^2_{condSP} = \sigma^2_Z\left(1 - \rho^2_{ZW}\right),$$

$$\rho = \sigma_Y\sigma_Z\left(\rho_{YZ} - \rho_{YW}\rho_{ZW}\right)$$

(”cond” subscripts mean “conditional”). Consider a two-dimensional random vector ($SB | V = v, SP | V = v$) with the specified parameters. For a given value of the land plot area and a given value of the price (in the example of the offer price) ($SP = sp, V = v$), the conditional mode of *SB* (improvement area) is equal to (by analogy with the proof given in [32]):

$$Mode\left(SP\middle|\,\mathrm{SP} = \mathrm{sp}, \mathrm{V} = \mathrm{v}\right) =$$

$$= \exp(\mu_{condSP} + \rho\times\frac{\sigma^{\;\;2}_{condSP}}{\sigma^{2}{}_{\alpha dSB}}\left(\ln\left(sb - \mu_{condSB}\right)\right) - \quad (9)$$

$$-\sigma^2_{condSP}\left(1 - \rho^2\right)).$$

Conditional median of *SP* is equal to:

$$Median(SP\,|\,\mathrm{SP} = \mathrm{sp}, \mathrm{V} = \mathrm{v}) =$$

$$= \exp(\mu_{condSP} + \rho\times\frac{\sigma^2_{condSP}}{\sigma^2_{condSB}}\left(\ln\left(sb - \mu_{condSB}\right)\right).$$

Conditional expectation of *SP* is equal to:

$$E(SP\,|\,\mathrm{SP} = \mathrm{sp}, \mathrm{V} = \mathrm{v}) = \exp(\mu_{condSP} +$$

$$+\rho\times\frac{\sigma^2_{condSP}}{\sigma^2_{condSB}}\left(\ln\left(sb - \mu_{condSB}\right)\right) + \frac{1}{2}\sigma^2_{condSP}\left(1 - \rho^2\right)).$$

Let’s assume the need to estimate the density factor in a group of items in the lower, middle, or upper price category. Such estimates can be constructed depending on the area of the land plot by modal, median or average values. However, the appearance of the surfaces shown in *Figure 2* suggests that the most conservative estimates will be based on modal values. Estimates for the median or average values seems overestimated (for $V = 28.000$ rubles/sq. m approximately 1.4 and 1.7 times, *Figure 4*). Let’s assume that we are interested in the following question: if the offer price is 28.000 rubles per sq. m and if the area of the land plot is equal to 30,000 square meters, then what area of improvements (and, accordingly, what coefficient of building density) should be considered adequate for such a price and land area. Under the development coefficient (density factor), we will understand the ratio of the estimated value of the area of improvements to the area of the land plot, i.e.

$$\frac{Mode\left(SB|SP = sp\right)}{sp}$$

(alternatively, $\frac{Median\left(SB|SP = sp\right)}{sp}$ or

$\frac{E\left(SB|SP = sp\right)}{sp}$).

Figure 4 shows that estimates for modal, median, and average values can differ significantly. Applying formula (9) to the results of calculating conditional parameters at the price of 28.000 rubles/sq. m and the value of the land area equal to 30.000 sq. m gives the result of 7.165 sq. m of improvements, and then the building density coefficient (density factor) is 7 165 / 30 000 = 0.24. Thus, based on the example data (table 1), the other coefficient of the building area at the price of 28,000 rubles/sq. m, the land area of 30.000 sq. m may be understood as not appropriate to the price set. The same result could be obtained by applying a formula similar to formula (1). In this section, sequential accounting of conditions (first prices $V = v$, then land area $SP = sp$) is used to show that the coefficient of building density is not a constant within one price group or even for one specific price, and has a power-law dependence on the land area. Left part of *Figure 4* shows lines of modal, median and average values of the area of improvements, depending on the area of the land plot for the case when

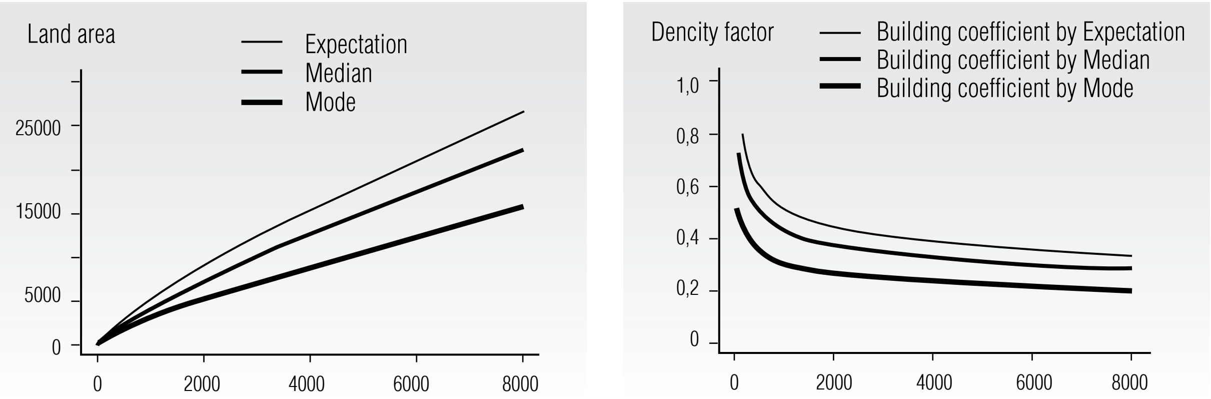


Fig. 4. Values of the area of improvements and building coefficients, depending on the area of the land plot

the offer price is equal to 28.000 rubles/sq. m of the area of existing improvements. The right figure shows lines of building coefficients for corresponding estimates of the area of improvements. *Figure 4* shows that at a given price (price group), the coefficient of development with acceptable accuracy for evaluation purposes can be estimated as a constant only if the land area is large enough. For plots with a small area, the development coefficient cannot be estimated as a constant and must be studied individually taking into account the area of the plot.

4. A note regarding the form of the joint logarithmically normal distribution of the vector (*V*, *SB*, *SP*)

Multidimensional distribution of vector components (W, Y, Z) = (ln(V), ln(SB), ln(SP)) it is normal and has symmetry. The scattering clouds of empirical observations will take the form of three-dimensional ellipsoids. The density maximum point has coordinates equal to the mean values of the components W, Y, Z. The distribution of the components of the vector (V, SB, SP) is asymmetric, the density maximum point is not the center of symmetry and can be calculated (see Appendix) using the following formulas:

$$V_{max} = \exp(\mu_W - \sigma_W^2 - \rho_{WY}\sigma_W\sigma_Y - \rho_{WZ}\sigma_W\sigma_Z),$$
$$SB_{max} = \exp(\mu_Y - \sigma_Y^2 - \rho_{YW}\sigma_W\sigma_Y - \rho_{YZ}\sigma_Y\sigma_Z),$$
$$SP_{max} = \exp(\mu_Z - \sigma_Z^2 - \rho_{ZY}\sigma_Z\sigma_Y - \rho_{ZW}\sigma_W\sigma_Z).$$

Figure 5 shows the following: the scattering of source data and the scattering of logarithms of source data, the point of maximum density in space (V, SB, SP) with coordinates V_{max} = 20 004 rubles per 1 sq. m, SB_{max} = 649 rubles per 1 sq. m, SP_{max} = 1 202 rubles per 1 sq. m and the point of maximum density in logarithmic space (W, Y, Z) = (ln(V), ln(SB), ln(SP)) with coordinates μW = 10.30; μY = 8.45; μZ = 9.35. Black marks the points of maximum density: on the left – in the space (V, SB, SP), on the right – in the space (W, Y, Z) = (ln(V), ln(SB), ln(SP)).

Figure 6 shows the result of 1000 generations three-dimensional random vectors with the same parameters.

It is obvious that (see Appendix) the maximum density point of a multidimensional vector (mode) whose logarithms are normally distributed together is unique. All other density values correspond to the sets described in the logarithmic dimension by hollow three-dimensional ellipsoids, and in the original coordinates, the sets corresponding to a single density value represent the result of distortion (stretch-

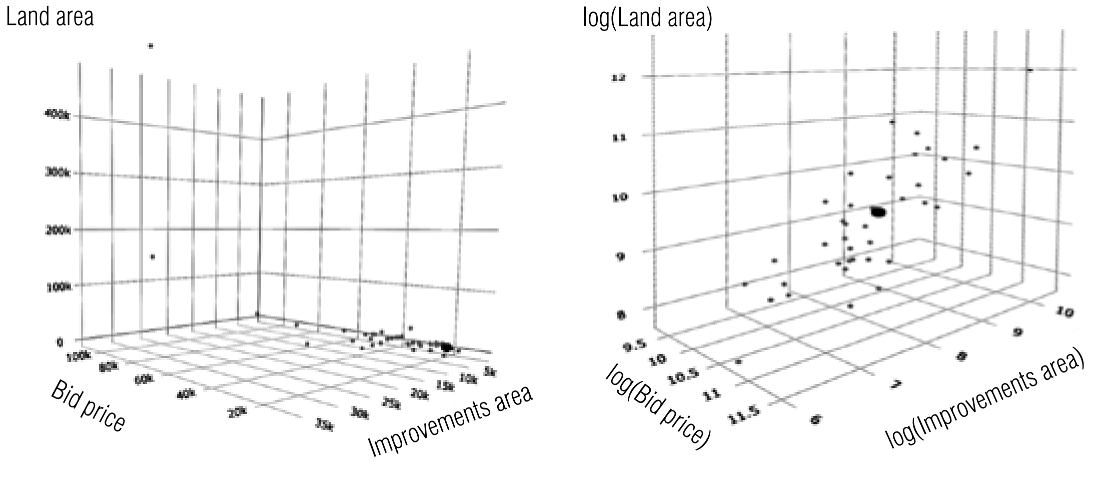


Fig. 5. The scattering of original data (left), the scattering of the logarithms of the original data (right)

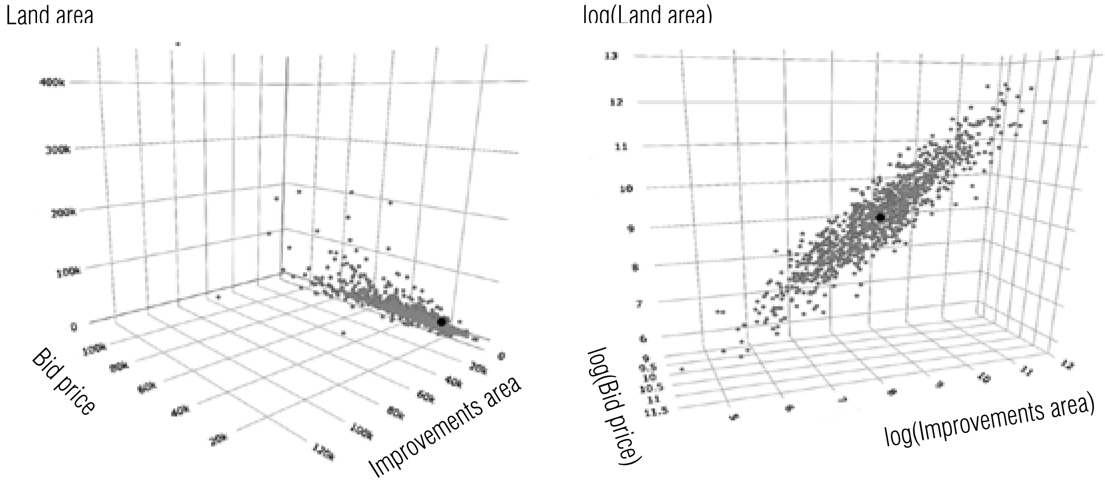


Fig. 6. Result of 1000 generations three–dimensional random vectors

ing) of the hollow ellipsoids during the inverse exponential transformation of space. Thus, it is the modal assessment of the market value that should lead to a correct result that does not create conflict situations. All other (non-modal) market value estimates are potentially a source of constant disputes about the market value of the object of valuation.

Conclusion

Considering the prices of objects of comparison and the values of price-forming factors as multidimensional random variables opens up new opportunities in the assessment of real estate. It often turns out that empirical observations of prices and their corresponding values of price-forming factors are well approximated by the logarithmically normal distribution law, including the multidimensional one, which allows us to derive calculation formulas for various estimation problems. The bulkiness of these formulas is compensated by the capabilities of modern applied statistical packages (in particular, R). In addition, the ability to reduce calculations to a well-studied multidimensional normal law by logarithm of components makes this choice of model distribution preferable.

Conditional price distributions with known values of price-forming factors make it possible to estimate the market value in full accordance with its definition fixed in Russian legislation and foreign standards, as the maximum point of the density of the conditional price distribution.

Conditional distributions of price-forming factors at a given offer price allow us to assess the adequacy of the offer price in terms of a set of price-forming factors.

It is hardly to be expected that practicing appraisers are prepared to apply the formulas given in this article in their daily practice of valuation and business analysis. This is not required. Once written and debugged, the script (in the statistical package R or in other specialized packages) will allow is to easily solve such problems practically in real time. It should be recognized that in the period of digital transformation of the economy and business analysis, it is time for the valuation business to move to advanced statistical packages and automatic data processing. ■

Appendix

Statement. The absolute maximum (mode) density of a random logarithmically normal vector $\bar{\boldsymbol{x}}$ is reached at the point with coordinates exp($\bar{\mu} - \Sigma \times \mathbf{1}$), where $\bar{\mu}$ is the vector of mathematical expectations of the logarithms of the component, Σ is the covariance matrix of the logarithms of the component, and $\mathbf{1}$ is a vector consisting of units.

Proof. Consider the density of a multidimensional normal distribution of a centered random vector $\bar{\boldsymbol{y}}$:

$$f(\bar{\boldsymbol{y}}) = \frac{1}{(2\pi)^{\frac{n}{2}} \sqrt[2]{\det\Sigma}} \exp\left(-\frac{1}{2}\left(\Sigma^{-1}\bar{\boldsymbol{y}}, \bar{\boldsymbol{y}}\right)\right).$$

When replacing variables $\bar{\boldsymbol{y}} = \ln(\bar{\boldsymbol{x}})$, the density of the lognormal distribution of the random vector $\bar{\boldsymbol{x}}$:

$$f(\bar{\boldsymbol{x}}) = \frac{1}{(2\pi)^{\frac{n}{2}} \sqrt[2]{\det\Sigma}} \times \frac{1}{\prod_{i=1}^{n} x_i} \times$$
$$\times \exp\left(-\frac{1}{2}\left(\Sigma^{-1}\overrightarrow{\ln(\boldsymbol{x})}, \overrightarrow{\ln(\boldsymbol{x})}\right)\right),$$

where $\frac{1}{\prod_{i=1}^{n} x_i}$ – coordinate transformation Jacobian,

Σ – covariance matrix, $\overrightarrow{\ln(\boldsymbol{x})}$ – centered random vector. At the point of absolute maximum density of the joint logarithmically normal distribution, the derivative in any direction must be zero, which means that all partial derivatives are equal to zero.

$$\frac{\partial f(\bar{\boldsymbol{x}})}{\partial \boldsymbol{x}_j} = \frac{1}{(2\pi)^{\frac{n}{2}} \sqrt[2]{\det\Sigma}} \times \frac{1}{\prod_{i=1}^{n} x_i}\left(-\frac{1}{x_j}\right) \times$$
$$\times \exp\left(-\frac{1}{2}\left(\Sigma^{-1}\overrightarrow{\ln(\boldsymbol{x})}, \overrightarrow{\ln(\boldsymbol{x})}\right)\right) + \frac{1}{2\pi)^{\frac{n}{2}} \sqrt[2]{\det\Sigma}} \times$$
$$\times \frac{1}{\prod_{i=1}^{n} x_i} \exp\left(-\frac{1}{2}\left(\Sigma^{-1}\overrightarrow{\ln(\boldsymbol{x})}\ \overrightarrow{\ln(\boldsymbol{x})}\right)\right) \times$$
$$\times \left(-\frac{2}{2}\,\Sigma^{-1}\overrightarrow{\ln(\boldsymbol{x})}\right) \times \frac{1}{x_j} = 0$$

After removing the common multipliers from brackets, the condition remains:

$$-\mathbf{1} + \left(-\Sigma^{-1}\overrightarrow{\ln(\boldsymbol{x})}\right) = 0 \text{ or } \left(-\Sigma^{-1}\overrightarrow{\ln(\boldsymbol{x})}\right) = \mathbf{1},$$

where **1, –1** – vectors with dimension n, consisting from units/negative units.

Let multiply the last equality on the left by Σ:

$$\Sigma\Sigma^{-1}\,\overrightarrow{\ln(\boldsymbol{x})} = -\Sigma \times \mathbf{1},$$
$$\boldsymbol{E} \times \overrightarrow{\ln(\boldsymbol{x})} = -\Sigma \times \mathbf{1}.$$

Here $\boldsymbol{E}$ is a unit matrix (on the main diagonal – units, the other elements are zero), **1** – a vector consisting of units. I.e., the values of the vector $\overrightarrow{\ln(\boldsymbol{x})}$ in which all partial derivatives are zero, are equal to the line-by-line sums of the covariance matrix, taken with the reverse sign.

It remains to remember that $\bar{\boldsymbol{y}} = \ln(\bar{\boldsymbol{x}})$ is a centered random vector. If the expectation vector $\bar{\mu}$ contains non-zero values, then the final solution is:

$$\overline{\ln(\boldsymbol{x})} = \bar{\mu} - \Sigma \times \mathbf{1} \text{ or } \bar{\boldsymbol{x}} = \exp(\bar{\mu} - \Sigma \times \mathbf{1}).$$

Taking into account negative definiteness of the quadratic form composed of second partial derivatives in point $\bar{\boldsymbol{x}} = \exp(\bar{\mu} - \Sigma \times \mathbf{1})$ (the author omits this bulky record since the result is obvious), the point $\bar{\boldsymbol{x}} = \exp(\bar{\mu} - \Sigma \times \mathbf{1})$ is a point of maximum density of lognormal random vector $\bar{\boldsymbol{x}}$.

The statement is proven.

References

1. Rusakov O.V., Laskin M.B., Jaksumbaeva O.I. (2016) Pricing in the real estate market as a stochastic limit. Log Normal approximation. *International Journal of Mathematical Models and Methods in Applied Sciences*, no 10, pp. 229–236.
2. Aitchinson J., Brown J.A.C. (1963) *The Lognormal distribution with special references to its uses in economics*. Cambridge: University Press.
3. Ohnishi T., Mizuno T., Shimizu C., Watanabe T. (2011) *On the evolution of the house price distribution*. Columbia Business School. Center of Japanese Economy and Business. Working Paper Series, no 296.
4. Anselin L., Lozano-Gracia N. (2008) Errors in variables and spatial effects in hedonic house price models of ambient air quality. *Empirical Economics*, vol. 34, no 1, pp. 5–34. DOI: 10.1007/s00181-007-0152-3.
5. Benson E.D., Hansen J.L., Schwartz Jr. A.L., Smersh G.T. (1998) Pricing residential amenities: The value of a view. *Journal of Real Estate Finance and Economics*, vol. 16, no 1, pp. 55–73. DOI: 10.1023/A:1007785315925.
6. Debrezion G., Pels E., Rietveld P. (2011) The impact of rail transport on real estate prices: an empirical analysis of the Dutch housing market. *Urban Studies*, vol. 48, no 5, pp. 997–1015. DOI: 10.1177/0042098010371395.
7. Jim C.Y., Chen W.Y. (2006) Impacts of urban environmental elements on residential housing prices in Guangzhou (China). *Landscape and Urban Planning*, vol. 78, no 4, pp. 422–434. DOI: 10.1016/j.landurbplan.2005.12.003.
8. Rivas R., Patil D., Hristidis V., Barr J.R., Srinivasan N. (2019) The impact of colleges and hospitals to local real estate markets. *Journal of Big Data*, vol. 6, no 1, article no 7 (2019). DOI: 10.1186/s40537-019-0174-7.
9. Wena H., Zhanga Y., Zhang L. (2015) Assessing amenity effects of urban landscapes on housing price in Hangzhou, China. *Urban Forestry & Urban Greening*, no 14, pp. 1017–1026. DOI: 10.1016/j.ufug.2015.09.013.
10. Saint Petersburg State Budget Department “Cadastral Valuation City Department” (2018) *Report on determining the cadastral value of real estate objects on the territory of Saint Petersburg*, no 1. Available at: http://www.ko.spb.ru/interim-reports/ (accessed 05 June 2019).
11. Peterson S., Flanagan A.B. (2009) Neural network hedonic pricing models in mass real estate appraisal. *Journal of Real Estate Research*, vol. 31, no 2, pp. 147–164.
12. Rafiei M.H., Adeli H. (2018) Novel machine-learning model for estimating construction costs considering economic variables and indexes. *Journal of Construction Engineering and Management*, vol. 144, no 12, article no 04018106. DOI: 10.1061/(asce)co.1943-7862.0001570.
13. Antipov E.A., Pokryshevskaya E.B. (2012) Mass appraisal of residential apartments: An application of Random forest for valuation and a CART-based approach for model diagnostics. *Expert Systems with Applications*, no 39, pp. 1772–1778. DOI: 10.1016/j.eswa.2011.08.077.
14. Kontrimas V., Verikas A. (2011) The mass appraisal of the real estate by computational intelligence. *Applied Soft Computing*, no 11, pp. 443–448. DOI: 10.1016/j.asoc.2009.12.003.
15. Park B., Baem J.K. (2015) Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data. *Expert Systems with Applications*, no 42, pp. 2928–2934. DOI: 10.1016/j.eswa.2014.11.040.
16. Case K.E., Shiller R.J. (1987) Prices of single-family Homes since 1970: New indexes for four cities. *New England Economic Review*, September, pp. 45–56. DOI: 10.3386/w2393.

17. Englund P., Quigley J.M., Redfearn C.L. (1999) The choice of methodology for computing housing price indexes: comparison of temporal aggregation and sample definition. *Journal of Real Estate Finance and Economics*, vol. 19, no 2, pp. 91–112. DOI: 10.1023/A:1007846404582.
18. Epley D. (2016) Assumptions and restrictions on the use of repeat sales to estimate residential price appreciation. *Journal of Real Estate Literature*, vol. 24, no 2, pp. 275–286. DOI: 10.5555/0927-7544.24.2.275.
19. Malpezzi S. (2002) Hedonic pricing models: A selective and applied review. *Housing economics and public policy: Essays in honor of Duncan Maclennan* (T. O'Sullivan, K. Gibb, eds.). Oxford, UK: Blackwell Science, pp. 67–89. DOI: 10.1002/9780470690680.ch5.
20. Case B., Quigley J.M. (1991) The dynamics of real estate prices. *Review of Economics and Statistics*, vol. 73, no 1, pp. 50–58.
21. Englund P., Quigley J.M., Redfearn C.L. (1998) Improved price indexes for real estate: Measuring the course of Swedish housing prices. *Journal of Urban Economics*, vol. 44, no 2, pp. 171–196.
22. Jones C. (2010) House price measurement: The hybrid hedonic repeat-sales method. *Economic Record*, vol. 86, no 272, pp. 95–97. DOI: 10.1111/j.1475-4932.2009.00596.x.
23. Wang F., Zheng X. (2018) The comparison of the hedonic, repeat sales, and hybrid models: Evidence from the Chinese paintings. *Cogent Economics & Finance*, no 6, pp. 1–19. DOI: 10.1080/23322039.2018.1443372.
24. Brunnermeier M.K. (2009) Bubbles. *The new Palgrave dictionary of Economics* (L.E. Blume, S.N. Durlauf, eds.). New York: Palgrave Macmillan.
25. Fabozzi F.J., Xiao K. (2019) The timeline estimation of bubbles: The case of real estate. *Real Estate Economics*, vol. 47, no 2, pp. 564–594. DOI: 10.1111/1540-6229.12246.
26. Fernandez-Kranz D., Hon M.T. (2006) A cross-section analysis of the income elasticity of housing demand in Spain: Is there a real estate bubble? *Journal of Real Estate Finance and Economics*, vol. 32, no 4, pp. 449–470. DOI: 10.1007/s11146-006-6962-9.
27. Phillips P.C.B., Shi S.-P., Yu J. (2015) Testing for multiple bubbles: Historical episodes of exuberance. *International Economic Review*, vol. 56, no 4, pp. 1043–1078. DOI: 10.1111/iere.12132.
28. Phillips P.C.B., Shi S.-P., Yu J. (2015) Testing for multiple bubbles: Limit theory of real time detectors. *International Economic Review*, vol. 56, no 4, pp. 1079–1134. DOI: 10.1111/iere.12131.
29. The Federal Law of 29.07.1998 No 135-FZ (edition of 29.07.2017) "*About assessment activity in the Russian Federation*". Available at: http://www.consultant.ru/document/cons_doc_LAW_19586/ (accessed 14 March 2020).
30. Slytsky A.A., Slytskaya I.A. (2020) *The modified extraction method and the generalized modified method of allocation. Use for analyzing the market segment that the item is being evaluated belongs to.* Available at: http://tmpo.su/sluckij-a-a-sluckaya-i-a-mmv-i-ommv-primenenie-dlya-analiza-rynka-3/ (accessed 14 March 2020).
31. Laskin M.B. (2014) Logarithmically normal distribution of prices and market value in the real estate market. *Saint Petersburg State Technological Institute Review*, no 25 (51), pp.102–106.
32. Laskin M.B. (2017) Market value adjustment for the pricing factor "square". *Property Relations in the Russian Federation*, no 8 (191), pp. 86–99.

About the author

Michael B. Laskin

Cand. Sci. (Phys.-Math.), Associate Professor;

Senior Researcher, St. Petersburg Institute for Informatics and Automation,
Russian Academy of Sciences (SPIIRAS),
39, 14 Line, St. Petersburg 199178, Russia

E-mail: laskinmb@yahoo.com

ORCID: 0000-0002-0143-4164

DOI: 10.17323/2587-814X.2020.2.64.83

Demand for skills on the labor market in the IT sector

Andrei A. Ternikov
E-mail: aternikov@hse.ru

Ekaterina A. Aleksandrova
E-mail: ea.aleksandrova@hse.ru

National Research University Higher School of Economics
Address: 3, Kantemirovskaya Street, Saint-Petersburg 194100, Russia

Abstract

One of the most dynamically changing parts of the labor market relates to information technologies. Skillsets demanded by employers in this sphere vary across different industries, organizations and even certain vacancies. The educational system in the most cases lags behind such changes, so that obsolete skillsets are being taught. This article proposes an algorithm of skillsets identification that allows us to extract skills that are needed by companies from different occupational groups in the information technologies sector. Using the unstructured online-vacancies database for the Russian regional labor market, skills are extracted and unified with the use of TF-IDF and *n*-grams approaches. As a result, key specific skillsets for various occupations are found. The proposed algorithm allows us to identify and standardize key skills which might be applicable to create a system of Russian classification for occupations and skills. In addition, the algorithm allows us to provide lists of the key combinations of skills that are in high demand among companies inside each particular occupation.

Key words: job vacancies in IT sector; online vacancies; unstructured data analysis; labor market; demand on skills of job candidates; combinations of skillsets.

Citation: Ternikov A.A., Aleksandrova E.A. (2020) Demand for skills on the labor market in the IT sector. *Business Informatics*, vol. 14, no 2, pp. 64–83. DOI: 10.17323/2587-814X.2020.2.64.83

Introduction

The process of employment in the labor market involves several parties: employers, employees, the educational system and state authorities. One of the most informative indicators for demand assessment is skills, which provide extensive information about competences and abilities demanded from the potential job candidate. However, sets of such skills are dynamically changing in different industries, organizations and even certain vacancies. These changes are connected to economic system fluctuations and labor market restructuring. Moreover, the professional standards that are formed with the help of the educational system become obsolete and inflexible to such changes. The particular interest of this study relates to the problem of identifying key skillsets on the labor market for occupational groups in information technologies (IT).

Several authors highlight issues of skills determination in the IT sphere. Firstly, this branch of the labor market has high volatility of technical and soft skills required, and must be analyzed in a time perspective [1–8]. Secondly, skills, especially technical skills, have an outstanding structure due to the precise formulation of programming languages, technological stack, interface instruments, etc., so that it is easier to classify them in attribution to several job positions [9–11]. Thirdly, the adoption of new technologies requires changing combinations of skills of workers in order to perform newly created tasks [12–16].

IT has penetrated a large part of the labor market. Technical specialists with certain sets of competences and knowledge are hired in spheres of economics and finance, public management, retail industry, etc. Thus, such specialists are also required to be competent in the professional activities of a particular company. Unique combinations of skills in certain areas can be formed in the education system not only in IT specialties. The rethinking of educational policy regarding the formation of skills cannot be separated from the demand from the labor market, which needs effective tools for identifying combinations of professional skills that are required by employers.

The present work concentrates on the creation of the algorithm of key skillsets, determining the particular occupational group, extraction in the IT sphere. The main question is: which skills are needed by companies from different occupational groups in the IT sector.

The paper is structured as follows. The first section contains the overview of related works and methods that are used for classification and clusterization purposes of online labor market data. The second section relates to the main algorithm representation which allows us to extract and classify information from an unstructured job advertisements database and its implementation to real data obtained for local labor market. The third section provides results of the work in terms of proposed key skills and combinations relating to different occupations of the IT sector of the labor market. Finally, in the concluding remarks we discuss the theoretical and practical implementation of the proposed algorithm and extended results are presented in the last section.

1. Related works and methods of skill demand analysis

Many researchers create various algorithms in order to extract information about occupations and particular skills from online job advertisement databases [17–25]. Such sources provide an extensive amount of information about the labor market. However, this data is, in general, unstructured. The main methods of processing this information are based on Natural Language Processing (NLP) techniques such as TF-IDF (Term Frequency – Inverse Document Frequency) matrices, *n*-grams (contiguous sequence of *n* items), classification techniques based on manual mark-up of data sample (LDA, KNN, SVM, etc.), clusterization of data [17–25].

Online vacancies databases, in general, have unstructured text fields that contain information about an occupation and competences required. However, such fields are manually filled by the companies' representatives, and that demands that data preparation procedures and algorithmic techniques be implemented in order to extract the appropriate information in standardized form. Some research papers try to resolve the classification task of how to match job titles and job descriptions from online advertisements with widely used classifications of occupations and skills such as ISCO[1], ESCO[2] and O*NET[3] [17–20]. Others implement classification models on the basis of expert markups [2, 4, 11, 13, 21]. In other words, the sample portion of data is analyzed and labelled by the domain experts and, then, this information is used to transfer this knowledge to the whole dataset. In addition, researchers use clusterization approaches for the occupations and skills determination in the data preparation process, thereby formulating the separate groups of jobs and competences after machine-based separation [19, 21–25]. Thus, the combination of different approaches and algorithms of data preparation and standardization allows us to provide the basis for an analytical research of labor market issues. A brief description of data, approaches and pipelines which are used in related works is presented in *Table 1.*

The information presented in the table allows us to summarize and systematize approaches for data organization, its processing and selection of criteria for identifying combinations of skills.

All authors present their algorithms of information extraction and systematization on the basis of online job advertisements. However, the manner of their implementation differs from the stated research task. For example, if the main research objective relates to the process of matching the unstructured text fields from job advertisements with the official classification for occupations and skills [17, 18, 20, 21] classification algorithms are implemented on the basis of finding similar patterns in job title description with the extended text information from official classifications and a significant amount of expert manual mark-up data.

The other approach relates to the data-driven approach where obtained data is manually corrected by domain experts in order to provide the appropriate systematization [19, 22–25]. These works focus on the data preparation part and clusterization algorithms. Despite the difference in research objectives, the common techniques of data preparation and extraction of standardized information are, in general, applied. All authors use the TF-IDF approach and tokenization (including stopword removal and stemming procedure) in order to process a wide amount of unstructured textual information. In addition, *n*-grams are used for more than one-word extraction. As a result, a set of unified patterns of information (e.g. occupations and skills) is obtained. However, the authors do not provide a generalized algorithm for matching different variants of the same pattern notation within the noisy data management process.

The choice of groups of occupations and skills is highly dependent on the official classifications and the volume of data. Thus, the level of such groups' aggregation demands an expert view based on the data characteristics. In general, the data is available for a one-year period and the search for appropriate patterns for the unstructured fields are simplified only for job advertisements published in one language.

[1] International Standard Classification of Occupations, https://www.ilo.org/public/english/bureau/stat/isco/

[2] European Skills/Competences, Qualifications and Occupations, https://ec.europa.eu/esco/portal/home

[3] The Occupational Information Network, https://www.onetonline.org/

Table 1.

Related works on online job advertisements analysis

Main direction	Data				Number of extracted groups		Methods of data processing				Number of manually processed data entries	Similarity indexes for terms' matching	Authors
	Volume	Period	Source(s)	Language	Occupations	Skills (terms)	TF-IDF	*n*-grams	Clusterization	Classification			
Clusterization of occupations	1.460	4 months (April – July 2018)	LinkedIn	English	8	96	+	+	Unweighted Pair Group Mean Average method	–	>900	Jaccard	[24]
	12.849	7 months (July – November 2015 and October – November 2016)	5 sources	English	69	NS*	+	–	Latent Semantic Indexing. Singular Value Decomposition	–	750	Cosine	[19]
Classification of occupations	75.546	27 months (February 2013 – April 2015)	WollyBI	Italian	9	NS	+	+	–	SVM (linear & RBF Kernel); Random Forest; NN	57.740	Levenshtein. Jaccard. Sørensen–Dice	[17]
	40.000	1 month (year is not specified)	12 sources	Italian	62	542	+	–	Weighted Word Pairs (WWP) extraction	LinearSVC & Perceptron classifier	412	Levenshtein	[21]
Clusterization of occupations and key skills extraction	2.786	3 months (Fall 2015)	10 sources	English	4	180	+	+	Latent Dirichlet Analysis	–	180	Centrality degree	[22]
	2.638	3 months (May – July 2018)	Indeed. com	English	48	480	+	–	Latent Dirichlet Analysis	–	480	% of occurrences	[23]
	1.050	5 months (July – November 2017)	6 sources	English	2	2.335	+	–	Latent Class Analysis. Singular Value Decomposition	–	NS	VARIMAX Rotation	[25]
Classification of occupations and key skills extraction	6.222	4 months (June – September 2015)	3 sources	Italian	6	NS	+	+	–	SVM (linear & RBF Kernel); Random Forest; NN	1.007	Random Forest importance	[20]
	~2 mln	24 months (2016–2017)	WollyBI	Italian	22	8	+	+	–	SVM	NS	Levenshtein. Jaccard. Sørensen–Dice	[18]

*NS stands for "Not specified"

Different research pipelines address different metrics of classification and clusterization model assessment. The point of interest here is how several patterns and terms could be matched with the stated domain. The authors use tokenization for raw texts and *n*-grams for construction of the set of terms. Similarity is found by implementing the Similarity indexes. In the case of preserving the word order the Levenshtein distance is the appropriate measure but if only intersection of common terms is valuable for detecting similarity – the Jaccard index is preferable.

2. Algorithm of skills demand analysis and related data

The proposed algorithm which allows us to conduct skills demand analysis is organized for online job advertisements data. These data are obtained from the open-source Application Programming Interface of HeadHunter[4], the largest Russian online recruitment platform[5]. The typical structure of an online vacancy is presented in *Table 2.*

Along with the presented data structure and methods used in the related works, the main interest of the work is to organize the process of knowledge extraction (groups of occupations and unified skills) from unstructured data. Then the opportunity to determine highly demanded skills and skillsets within occupational groups can be performed in an accurate manner. Despite the use of classification algorithms for aggregation of job occupations in related works, the current dataset has already been codified by the data producer. Thus, we assume at the preliminary stage of analysis that occupational groups already exist. In order to organize the algorithm description, several concepts need to be formalized and simplified for research purposes.

Definition 1. **Online vacancy.** Let I be the set of vacancy identifiers; H is the set of specialization codes; S is the set of key skills in text format. Suppose $V = \{v_1, \ldots, v_n\} : n \in \mathbb{N}$ is the set of vacancies; an online vacancy v is a 6-tuple $v = (i, C, d, p, g, K)$, where $i \in \mathrm{I}$, $C \subseteq H : |C| \in \{\overline{1,6}\}$ is the subset of specialization codes, d is the vacancy published date, p is the text name of vacancy's position, g is the

Table 2.

Structure of a typical HeadHunter online job advertisement

Field	Field type		Description
	Structured	Unstructured	
Vacancy ID	+		Numeric code
Specialization ID	+		Set (from 1 to 6 including) of numeric codes[6]
Published date	+		Long date format
Position Name		+	Text
Job description		+	Text
Key skills		+	Set of texts (30 at maximum: each up to 100 symbols)

[4] HeadHunter API, https://dev.hh.ru

[5] SimilarWeb: websites ranking, https://www.similarweb.com/top-websites/russian-federation/category/jobs-and-career/jobs-and-employment

[6] HeadHunter API: Specializations, https://api.hh.ru/specializations

text description of the vacancy, $K \subseteq S : |K| \leq 30$ is the subset of skills (in text format) for a particular vacancy.

Current research is concentrated on the IT sector and the methodology is tested on the local job market (the city of Saint Petersburg). Online-vacancies from the IT sphere in Saint Petersburg were extracted from 2015 till 2019 (the official HeadHunter classification is used to obtain IT-related vacancies by Specialization ID). Each data entry of a particular vacancy (v) contains of the ID-code of the vacancy (i), HeadHunter specialization codes (C) and the list of skills required for a particular employer (K). The research objectives are concentrated on the process of skills' determination, thus, the portion of data where skills are given in the separate data entry are used. Such a sample consists of 63.869 vacancies from May 2015 till September 2019. Each vacancy includes from 1 to 6 professional area codes (HeadHunter professional area classifier). The distribution along 36 areas (inside the group of IT sphere) is presented in *Table 3.*

Despite the HH classifiers' distribution, some spheres could be deleted and merged in onw bigger subgroup. In accordance with the classification introduced in [20] we define 6 + 1 groups of IT specialists. After deleting vacancies with not a purely IT profile and regrouping, 56.000 observations (vacancies) are obtained. The share of deleted professional areas is 10.2%. The new distribution among the rest of aggregated occupational groups of vacancies and their names are presented in the *Table 4.*

***Definition 2.* Job occupation (occupational group).** Let H be the set of specialization codes (identifiers). Suppose O is the ordered set of aggregated job occupations, a job occupation $o \in O$, is a 2-tuple $o = (L, a)$, where $L \subseteq H$ is the subset of specialization codes attributing the particular aggregated group with text name a.

Table 3.

Distribution of vacancies by HeadHunter specializations in data sample

HeadHunter Specialization ID	Share, %	Name
1.221	20.37	Software Development
1.82	7.99	Engineer
1.9	4.90	Web Engineer
1.89	4.52	Internet
1.10	4.18	Web Master
1.327	3.83	Project Management
1.225	3.42	Sales
1.137	3.21	Marketing
1.272	3.11	System Integration
1.295	3.04	Telecommunication
1.211	2.99	Support, Helpdesk
1.117	2.90	Testing
1.270	2.86	Networks
1.273	2.73	System Administrator
1.25	2.69	Analyst
1.172	2.62	Entry Level, Little Experience
1.50	2.25	ERP
1.400	2.11	SEO
1.536	2.10	CRM Systems
1.474	2.04	Startups
1.359	1.75	E–Commerce
1.116	1.58	Content
1.475	1.51	Video Games Development
1.113	1.50	Consulting, Outsourcing
1.246	1.42	Business Development
1.420	1.39	Database Administrator
1.395	1.04	Banking Software
1.203	1.01	Data Communication and Internet Access
1.110	0.98	IT Security
1.161	0.86	Multimedia
1.296	0.67	Technical Writer
1.274	0.66	Computer Aided Design Systems
1.3	0.64	CTO, CIO, IT Director
1.277	0.61	Mobile, Wireless Technology
1.30	0.34	Art Director
1.232	0.18	Producer

Table 4.

Distribution of vacancies among IT occupational groups

Name	Short name	Share, %	HeadHunter Specialization ID
High–level IT specialists	high	13.10	1.327, 1.272, 1.25, 1.113, 1.3
Low–level IT specialists	low	3.66	1.172, 1.296
Engineering professionals	engineers	16.18	1.82, 1.295, 1.117, 1.277
Software developers	soft	22.67	1.221
Web and multimedia developers	web	20.13	1.9, 1.89, 1.10, 1.400, 1.475, 1.161
Administrators and database designers	admin	19.30	1.211, 1.270, 1.273, 1.50, 1.536, 1.420, 1.395, 1.203, 1.110
Others	others	4.96	1.474, 1.359, 1.274

To simplify the further analysis of key skills extraction for particular occupational groups (O, where $|O| = 7$ for proposed dataset), all data processing is organized on the whole sample during the 5 years of data presented. As an assumption for such aggregation, the relative distribution of vacancies in occupational groups is used (*Figure 1*). Thus, the proportion of data in the sample is relatively the same for each occupational group in a time perspective.

Following the research objectives of the current work, key skills and their combinations should be unified and extracted along the set of vacancies (V). However, before introducing the skills' extraction algorithm, each online vacancy that may relate to several job occupations should be mapped in the new data structure (IT online vacancy).

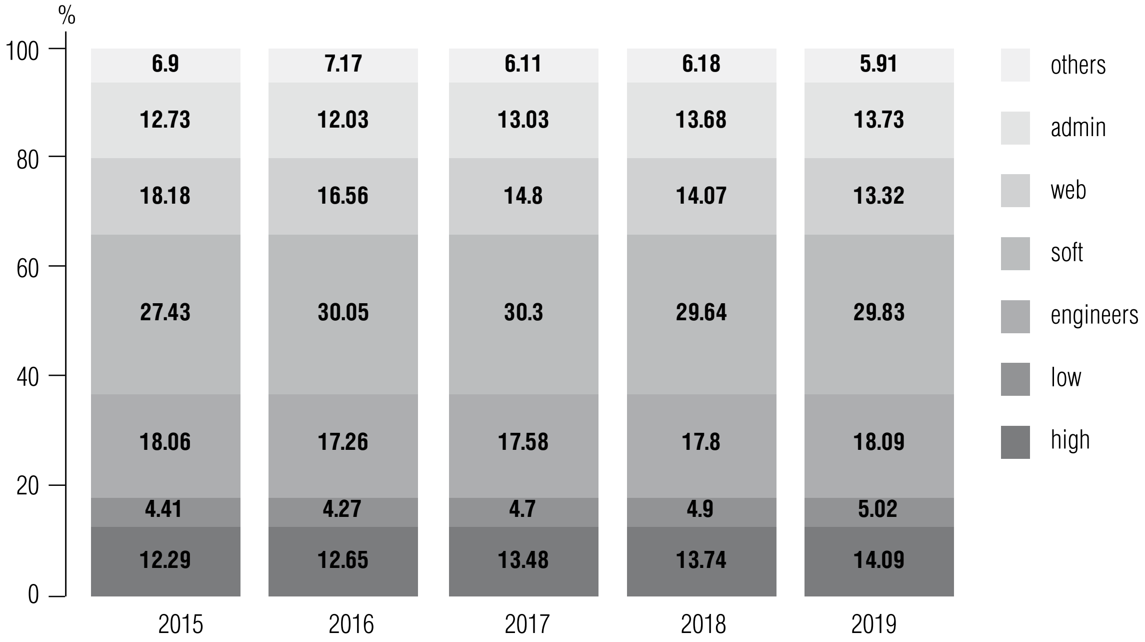


Fig. 1. Distribution of vacancies by occupational groups in time perspective

In order to provide the results of finding key skills and skillset by job occupations, the skill extraction algorithm is organized. Hereinafter, the algorithm of skills' extraction is implemented to IT online vacancies.

Input: IT online vacancies J.
Output: the set of standardized terms (skills) $\tilde{K}$, matched to database of online-vacancies.

```
let $\tilde{S} \subseteq S$ denote the set of unique text descriptions (skills) obtained from $J$
let $B$ denote the set 2-tuples: text description (skill) and its frequency (number of occurrences) in $J$
foreach $\tilde{s}_i \in \tilde{S}$ do
    $b_i \leftarrow$ occurrences of $\tilde{s}_i$ in $J$
    $B[i] = (\tilde{s}_i, b_i)$
end foreach
sort $B$ in descending order by $b$
procedure FrequentTerms($h$, $t$)
    $\tilde{h} \leftarrow$ subset of $h$ if $h_i > t, \forall h_i \in h$
    return $\tilde{h}$
end procedure
introduce threshold $t$
$\tilde{B} \leftarrow$ FrequentTerms($B(b)$, $t$)
$T \leftarrow$ 3-tuple set of manually standardized terms $T = (u, x, xs)$, where $u$ denotes identifier of standardized term (skill), $x$ – the name in text format, $xs$ – the set of synonyms in text format for particular pair $(u, x)$
function Tokenizer($j$)
    white space normalizer
    punctuation removal
    lowercase
    stemming (English & Russian)
    stopwords removal (English & Russian)
end function
procedure NGrams($J$, $n$)
    for $j$ in $J$ do
        $G \leftarrow n$-grams of size $n$ for Tokenizer($j$)
        add $G$ to ngramterms
    end for
    return ngramterms
end procedure
introduce thresholds $t_1$, $t_2$, $t_3$
ngram1:= FrequentTerms(NGrams($B(s)$, 1), $t_1$)
ngram2:= FrequentTerms(NGrams($B(s)$, 2), $t_2$)
ngram3:= FrequentTerms(NGrams($B(s)$, 3), $t_3$)
for obtained databases with $n$-grams provide manual processing (clear uninformative terms)
each entry in these $n$-grams databases has the set of  identifiers
procedure MatchTerms($X$, $Y$)
```

```
        let $L$ is the set of unique combinations from $X$ and $Y$, where $L=\{l_1|l_1\in X, l_2|l_2\in Y\}$; $l_1$, $l_2$ are sets of identifiers ($i$, $\tilde{s}$ )
        for $l_1$, $l_2$ in $L$ do
            $M\leftarrow$ Jaccard Similarity: $\frac{|l_1(i)\cap l_2(i)|}{|l_1(i)\cup l_2(i)|}$
            if >0.5 do
                add ($l_1$, $l_2$) to termsmatched
            end if
        end for
        return termsmatched
    end procedure
    for each pair of datasets: ngram1, ngram2, ngram3 provide MatchTerms($X$, $Y$) $\rightarrow$ M1, M2, M3
    for each pair of datasets $T$, M1, M2, M3 provide MatchTerms($\tilde{X},\tilde{Y}$) procedure, where $\tilde{X}:=T$, $\tilde{Y}:=T$ {M1,M2,M3}
    $\tilde{K}\leftarrow X$ left-join $Y$
    $\tilde{K}$ – is output database, with standardized terms, their synonyms, unigrams, bigrams, trigrams
```

Definition 3. **IT online vacancy.** Let $J\subseteq V$ be the set of IT online vacancies, where $J=\{j_1, ..., j_m\}$: $m\le n$, $m\in\mathbb{N}$. Let $\mathbf{c}$ denote the classification codes from online vacancy as follows: $\mathbf{c}=(c_1, ..., c_z)\in C$, where $z\le 6$. Let $\mathcal{L}$ denote the ordered set of labels obtained from job occupations with relation $(L, a)\in O\mapsto\mathcal{L}$ in the following form $\mathcal{L}=(\lambda_1, ..., \lambda_q)$, where $q=|O|$. Introduce the function

$$f(\mathbf{c},\lambda_q)=\begin{cases}1,\mathbf{c}\subseteq L_q\\0\end{cases},$$

that associates the occupational classification codes from job occupation o with codes from aggregated job occupations $\mathcal{L}$. Introduce mapping relation H: $C\mapsto\mathcal{L}$ that provides the multi-label classification and maps the set of aggregated job occupations $\mathcal{L}$ on the basis of the occupational classification codes as follows $\tilde{O}=H(\mathbf{c})=\{\lambda\in\mathcal{L}\ f(\mathbf{c},\lambda)=1\}$, where $\tilde{O}$ is the set of aggregated group names. Thus, an IT online vacancy o is a 3-tuple $j=(i, \tilde{O}, K)$.

In *Table 5*, the distribution of obtained vacancies by aggregated groups (job occupations) is presented. The distribution of job occupations assigned by companies in the database is not homogeneous. In other words, a portion from 6 to 30% in each job occupation is strongly related to the occupation itself. The other major part of vacancies relates to more than one aggregated group. Thus, in the following analysis the diversification of skills that are related to a particular job occupation is needed.

Specifically, assigning the algorithm to the dataset of IT online vacancies the procedure of extracting skills (skillsets) is as follows. In the data sample 13.347 raw unique skills are presented. The descriptions of such skills are not unified in general. In other words, each company can enter its own text string from 1 to 100 characters. For example, one skill's entry may contain one word/phrase or a sentence containing such words separated with punctuation symbols or spaces (no generic format). In order to automate the extraction of certain skills and unify different forms of notation of one term, text mining techniques are used.

According to [20], on the first stage of data preprocessing n-grams (contiguous sequence of items) of words can be constructed. From the vector corpus (TF–IDF) of skills pre-

Table 5.

Distribution of aggregated occupations in data sample

Short name	Number of vacancies	% of non-mixed vacancies by occupational groups
high	19.266	16.28
low	5.383	17.28
engineers	23.787	10.15
soft	33.333	29.78
web	19.312	9.29
admin	20.825	6.18
others	7.293	15.80

sented in vacancies' descriptions, uni-, bi- and tri-grams were constructed with the use of the following tokenizer: removing all punctuation and extra spaces, lowercase to all letters, words' stemming both in English and Russian language, stop words removal. Within these extracted terms, the initial structure and formulation of skills were saved. The first step is the extraction of meaningless words (non-informative itself) and messy data separation from informative patterns. For the unigram database, 348 entries were extracted from 5.234 non-unique terms; for bigram – 577 out of 1.090; for trigram – 110 out of 303. The second step is the creation the database of synonyms for already obtained patterns. HeadHunter API: Suggestions (Key skills suggestions) allows us to obtain a portion of synonyms for manual processing of obtained terms. After such synonym extraction, the term matrix for 707 terms was obtained (1.296 entries for manual checking). The third step is the addition of terms from the Stack Overflow Developer Survey[7] (108 terms for the most popular IT technologies names) and final correction of appropriate terms (database with reference terms and their synonyms).

As a result, 435 standardized terms were obtained within 420 synonyms for them. Such dataset contains both technical (hard) and non-technical (soft) skills for the given sector of the labor market. The last stage is composed through intersection of raw skills (codified with unique identification codes), matched exactly with specific terms obtained from a manually corrected list of HeadHunter synonyms and the results from TF-IDF matrices (for uni-, bi-, tri-grams), resulted within the pairs: skill ID and term. In order to automatically define the closeness between several terms (on the basis of the unique set of IDs for each term) and match the rest of the data with the given standardized terms, the Jaccard distance measure is used. For example, the similarity between two sets of words (terms) $A$and B could be found with the following formula:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}.$$

[7] Stack Overflow Annual Developer Survey, https://insights.stackoverflow.com/survey/

This measure is appropriate for categorical data closeness comparison and its value is in the range from 0 to 1 inclusive. However, the choice of the cutoff-point highly depends on the data and research objectives. As the threshold for identifying close terms, the level above 0.5 is used (after manual processing of obtained datasets). Thus, 53.672 vacancies (95.8% of the initial sample) contain at least one of standardized skill from the previously obtained dataset of terms and their synonyms. The percentage representation of the Top-20 standardized skills (by the number of occurrences in the sample) among the whole dataset is presented in *Table 6.*

However, following the objectives of the current research, the particular analysis of relevant and highly-demanded (from employer's side) skills (and their combinations) lies behind the determination of relevant skills that are at the same time strongly related to a particular occupational group (specific skills) and supported by a relatively large number of employers.

After the data preparation of the dataset of vacancies, the following data structure is obtained: the dataset of 305.217 observations (particular skill/term from the vacancy), where each observation has the ID of a standardized skill, the ID of the vacancy and the occupational group code. In order to provide the classification of skills in accordance with the stated groups of vacancies, the process of finding pairs and triplets of skills was conducted for each vacancy group. After obtaining the pairs and triplets of skills (non-zero by Jaccard Similarity), the highly matched (threshold by Jaccard Similarity) skills were extracted. The general scheme of the proposed algorithm implemented to the dataset is presented in *Figure 2.*

On the first step, all 435 skills, pairs of them (C_{435}^{2} = 94.435) and triplets (C_{435}^{3} = 1.362.345) were used for finding the Jaccard Similarity for each of 7 groups of vacancies (occupations).

Table 6.

Top-20 skills by their occurrence in the sample dataset for the IT sector

Skill Name	% of occurrences in database of skills
HTML/CSS	6.73
JavaScript	4.69
1C	3.48
SQL	3.25
PHP	2.63
Git	2.53
Linux	2.32
Java	2.28
MySQL	1.86
Negotiation skills	1.61
Sales Skills	1.52
Business communication	1.51
English	1.45
Testing Framework	1.43
Python	1.40
jQuery	1.36
C/C++	1.30
OOP	1.29
C#	1.28
.net	1.27

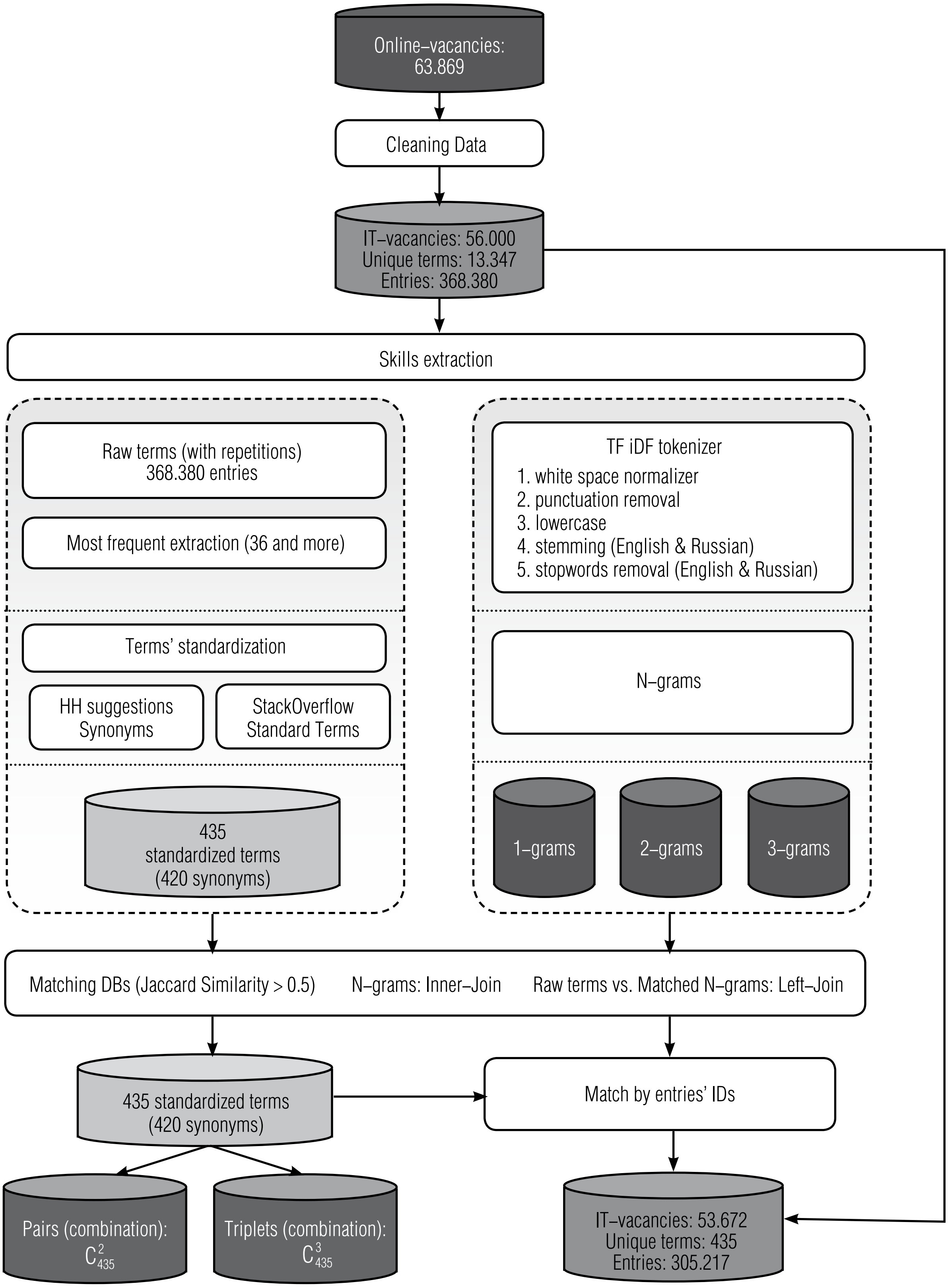
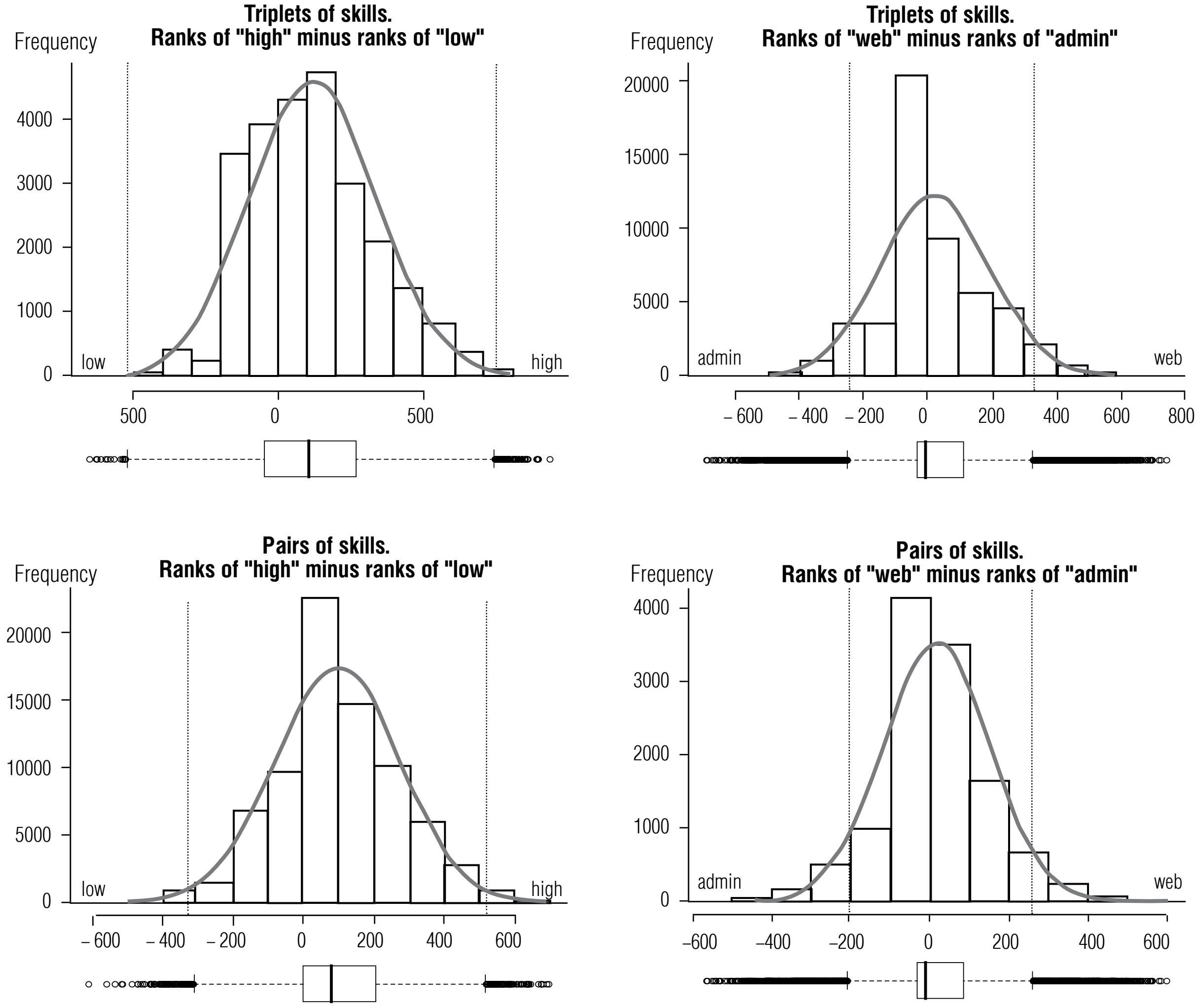


Fig. 2. Baseline algorithm of skills extraction from unstructured database

Secondly, using pairs and triplets of skills (unique combinations without repetitions), each dataset with terms was ranked by Jaccard Similarity (after removing such observations, where Jaccard Similarity equals zero) within their quantiles (permilles: 0.1% step for pairs and triplets in order to produce the variability).

For each pair and triplet of skills, such a measure was calculated on the basis of the number of vacancies that include such combinations. Thirdly, the differences in ranks for each pair of vacancies' groups were found. Fourthly, in order to extract the specific features (set of skills), the outliers in such distribution were found (as a provision for highly diverse skills and skillsets that can describe and separate groups of vacancies. The statistical logic behind this shows that the distribution of ranks' differences is quite close to normal and the detection of outliers (too vast difference in ranks of skills and skillsets) allows us to provide the appropriate selection of skills which can separate different groups of vacancies. For example, several pairs of such groups are represented in *Figure 3*


Triplets of skills.
Ranks of "high" minus ranks of "low"

Triplets of skills.
Ranks of "web" minus ranks of "admin"

Pairs of skills.
Ranks of "high" minus ranks of "low"

Pairs of skills.
Ranks of "web" minus ranks of "admin"

Fig. 3. Distribution of rank differences by Jaccard Similarity for occupational groups

[8] IQR – interquartile range

(cutoffs for skills detection are boundaries of whiskers in boxplot: 1.5 IQR[8] below and upper for relatively appropriate quantile rank difference).

Fifthly, for extracted pairs and triples, the procedure of addition of unique skillsets (different sets of skills) for each pair of an occupation's groups is provided. Thus, in the cross-intersection of skills (technically, with zero value of Jaccard Similarity) only those in above 95% (by quantile difference) are added in order to detect initially key (and different) skillsets. Sixthly, for each pair of vacancies' groups ($C_7^2 = 21$), core and determinant skills (and their combinations) were determined (and skills, which are unique in certain class in the last decile, were added for them unique skills by cross-intersection). Thus, three matrices 7×7 were obtained, where on the cross-section of i-th row and i-th column ($i \neq j$) the unique sets of skills (codified separately for unique skills, their pairs and triplets) are presented (distinguished and unique above 95% threshold skills of i-th group, comparing with j-th group of occupations). Eighthly, such skills were extracted in the following manner: the presence of core skills that determines each occupational group was already set, thus, with the use of by-row intersection (for each of given matrices), determinant ones (core and different for the given occupational group) are extracted. The following thresholds are used: for pairs the threshold of at least 2/3 different from the other groups (4 and more out of 6 the rest groups repeated); for triplets 100% different skills (6 out of 6). Using the logic given above, the lists of such skills were obtained for each particular occupational group of vacancies that represents at the same time core (highly-demanded) skills from companies and skills inherent in the particular occupation.

3. Results

On the stage of key skills and skillsets determination for different occupational groups in the IT sector, the most popular skills are extracted. In accordance with the different occupations, such skills are presented in the form of the Word Clouds by Top-50 skills for each occupation (by the number of occurrences in vacancies' description) in *Figure 4.*

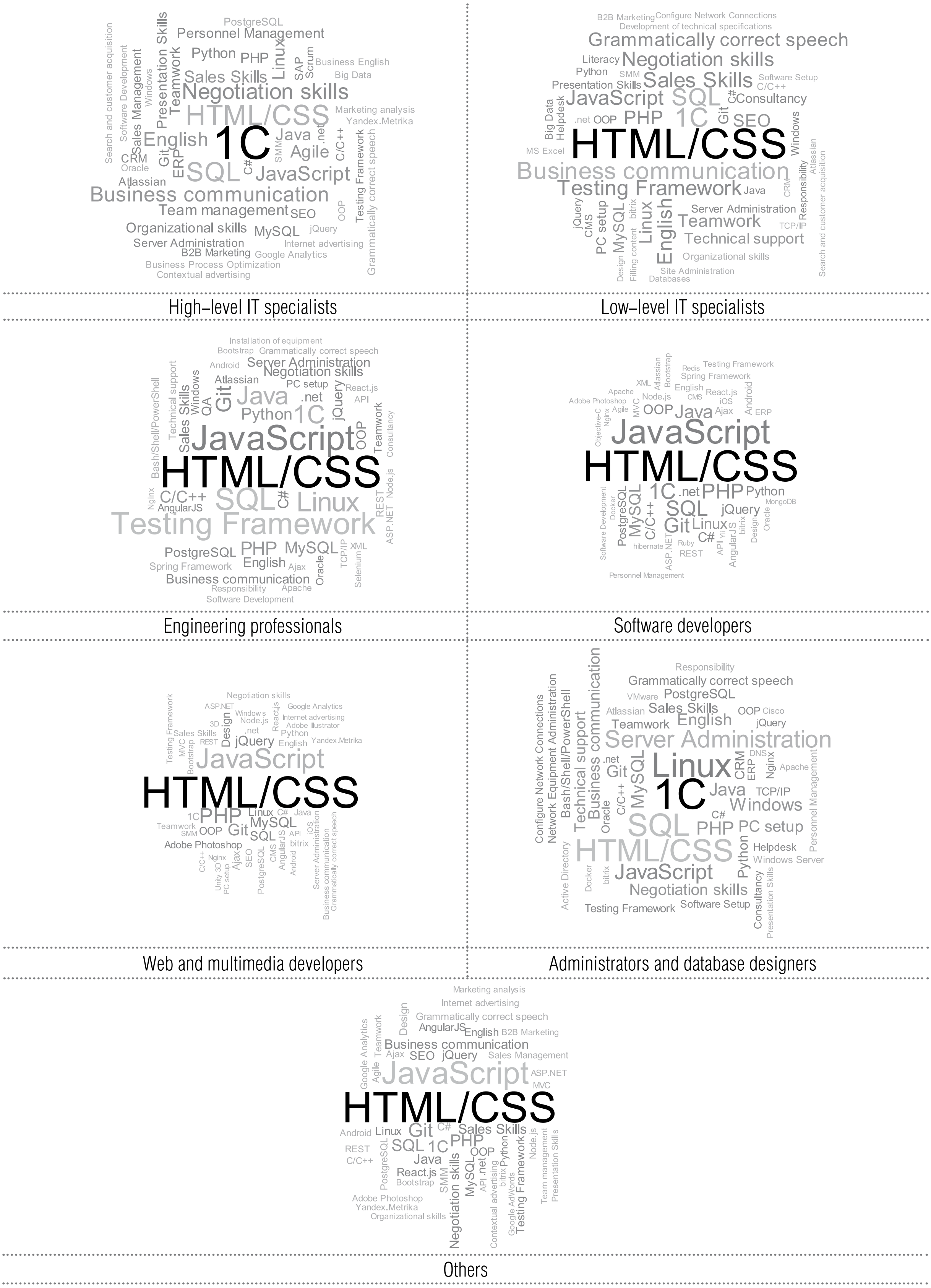
However, within the presence of vacancies that are related to several occupations, several skills are duplicated among different groups of occupations. Thus, at the stage of extraction of pairs and triplets of skills such duplication is reduced using the cross-intersection of determinant skills. The most in demand and at the same time occupational specific pairs of skills are presented in *Table 7*, triplets – in *Table 8*[9].

As a result, from the qualitative point of view, the sets of skills in high demand are extracted for different occupational groups. Moreover, using pairs and triplets of skills, the specific combinations of skills are obtained. Thus, the proposed methods of skills preparation and extraction could be useful for a broader understanding of the demand side of the labor market and provide more evidence for the educational system in order to maintain and renew educational standards to follow the trends (in skills) on the labor market.

Conclusion

Along with the results obtained in this work, it is worth mentioning that the market is slightly diverse in terms of certain occupational groups segregation. In other words, this work provides an opportunity to run the set of classification and clusterization algorithms in order to provide the other occupational separation. In addition, the results are limited by the presence of posted vacancies in the specific online source of data but

[9] Full lists of pairs and triplets may be presented by the authors upon request



High–level IT specialists

Low–level IT specialists

Engineering professionals

Software developers

Web and multimedia developers

Administrators and database designers

Others

Fig. 4. Top–50 skills by occupational groups

Table 7.

Key pairs of skills by occupational groups

Skill 1	Skill 2	Jaccard Similarity	Group
Dart	Flutter	0.167	high
Billing	Solaris	0.136	high
Arduino	Raspberry Pi	0.125	high
Technical means of information protection	Assembly	0.073	high
Means of cryptographic information protection	Assembly	0.065	high
Production automation	CAD	0.125	low
CCNA	OSPF	0.125	low
A/B tests	Mobile Marketing	0.100	low
Arduino	ARM	0.083	low
Business Process Optimization	Citrix	0.038	low
Network monitoring systems	Google Cloud Platform	0.071	engineers
Cordova	Xamarin	0.065	engineers
Personnel Management	Yandex.Metrika	0.014	engineers
Elasticsearch	Node.js	0.011	engineers
MS SharePoint	Windows	0.010	engineers
Firebase	Google Cloud Platform	0.083	soft
Grammatically correct speech	Drawing up contracts	0.015	soft
Elasticsearch	Yii	0.011	soft
Scrum	TFS	0.010	soft
Contextual advertising	Search and customer acquisition	0.006	soft
Business Intelligence Systems	Olap	0.063	web
3D	Altium Designer	0.022	web
SPA	Unit Testing	0.018	web
Writing Articles	Google AdWords	0.017	web
API	Mercurial	0.016	web
Website technical audit	Technical translation	0.074	admin
Analytical research	System analysis	0.033	admin
Apache	Windows Server	0.029	admin
REST	Xsd	0.022	admin
API	Xsd	0.016	admin
Proofreading Texts	Adobe Lightroom	0.111	others
Mobility	Billing	0.111	others
Pandas	Wifi networks	0.100	others
Website technical audit	SMO	0.091	others
A/B tests	Business Analysis	0.080	others

Table 8.

Key triplets of skills by occupational groups

Skill 1	Skill 2	Skill 3	Jaccard Similarity	Group
ARM	GCC	Raspberry Pi	0.019	high
Media planning	Marketing campaign planning	facebook	0.018	high
CentOS	EJB	NetBeans	0.017	high
Video processing	Adobe Premier Pro	SketchUp	0.013	high
Business planning	Mobile Marketing	Product Marketing	0.012	high
Production automation	Instrumentation	CAD	0.050	low
Process Automation	Instrumentation	CAD	0.048	low
Production automation	Process Automation	CAD	0.043	low
Debian	OSPF	VLAN	0.043	low
Analytical research	Business Analysis	Product Marketing	0.038	low
Video processing	Image processing	Adobe Lightroom	0.020	engineers
Conducting correspondence in a foreign language	Writing Press Releases	Technical translation	0.018	engineers
Writing Press Releases	Written translation	Technical translation	0.015	engineers
Image processing	Adobe After Effects	Adobe Lightroom	0.014	engineers
FreeBSD	OSPF	VLAN	0.014	engineers
Search engine optimization sites	Work with exchanges	Website technical audit	0.021	soft
Mathematical analysis	MATLAB	R	0.017	soft
Mathematical statistics	MATLAB	R	0.016	soft
Proofreading Texts	Writing Press Releases	Presentation Preparation	0.013	soft
Work with exchanges	Website technical audit	SMO	0.013	soft
Microsoft Azure	TensorFlow	Torch/PyTorch	0.020	web
Banner advertising	Video processing	Adobe Premier Pro	0.016	web
Reporting	Tax reporting	Billing	0.016	web
Proofreading Texts	Writing Press Releases	Rewriting	0.015	web
Microsoft Azure	Spark	TensorFlow	0.014	web
Mathematical analysis	Olap	VBA	0.019	admin
Mathematical analysis	A/B tests	R	0.018	admin
Chef	LDAP	Wifi networks	0.015	admin
BGP	Chef	LDAP	0.013	admin
Chef	LDAP	OSPF	0.013	admin
Internal website optimization	Website technical audit	SMO	0.063	others
Internal website optimization	Russian search engines	SMO	0.048	others
Flask	Pandas	Wifi networks	0.037	others
Mobility	Electronic document management	Billing	0.031	others
Internal website optimization	Lidogeneration	SMO	0.027	others

could be aggregated on the level of the population, using the official statistics (if the objectives of the work will be directed to economic issues: salary, changes in time perspective, etc.).

Points for discussion are as follows. Firstly, the proposed database for the analysis has a highly diverse set of already defined occupations. In other words, introducing classification or clusterization for detecting occupational groups could improve the overall results. Nevertheless, the provided list of skills' combinations is constructed in terms of occupation-specific skillsets extraction maintenance.

Secondly, following the logic of mixed occupations that could be declared by the employer in one specific vacancy, the skills' grouping (e.g. "soft" and "hard" skills) may be used by the feature for classification purposes. Moreover, there are skills that are related to the technology itself and the framework for its implementation that cannot be separated in one-way or both directions.

Thirdly, provision of the larger sequence of words in -grams (4 and more) may provide more evidence for extraction skills from unstructured databases. However, the computing power for calculating such algorithms could be extremely high and may demand the simplification of similarity metrics calculation (e.g. using hash-functions and approximate formulas).

Fourthly, the extended implementation of the algorithm could be aimed at detection of key skillsets in the other sectors of the labor market, capturing changes in a time perspective and the organization of cross-regional comparison.

Finally, several contributions of the current work could be highlighted. Firstly, the proposed algorithm allows us to identify and standardize key skills which might be applicable for creation of the system of Russian classification for occupations and skills. Secondly, the algorithm allows us to provide lists of the most popular (key) combinations of skills that are in high demand by companies and employers inside each particular vacancy. Finally, the flexibility of the algorithm allows us to combine it with classification and clusterization techniques of data analysis that could be useful for research into the labor market.■

References

1. Autor D.H., Levy F., Murnane R.J. (2003) The skill content of recent technological change: An empirical exploration. *The Quarterly Journal of Economics*, vol. 118, no 4, pp. 1279–1333. DOI: 10.1162/003355303322552801.
2. Bensberg F., Buscher G., Czarnecki C. (2019) Digital transformation and IT topics in the consulting industry: A labor market perspective. *Advances in consulting research: Recent findings and practical cases* (ed. V. Nissen). Cham, Switzerland: Springer, pp. 341–357.
3. Christoforaki M., Ipeirotis P.G. (2015) A system for scalable and reliable technical-skill testing in online labor markets. *Computer Networks*, vol. 90, pp. 110–120. DOI: 10.1016/j.comnet.2015.05.020.
4. Florea R., Stray V. (2018) Software tester, we want to hire you! An analysis of the demand for soft skills. Proceedings of the *19th International Conference on Agile Processes in Software Engineering and Extreme Programming (XP 2018), Porto, Portugal, 21–25 May 2018* (eds. J. Garbajosa, X. Wang, A. Aguiar), pp. 54–67.
5. Goles T., Hawk S., Kaiser K.M. (2008) Information technology workforce skills: The software and IT services provider perspective. *Information Systems Frontiers*, vol. 10, no 2, pp. 179–194.
6. Johnson K.M. (2016) Non-technical skills for IT professionals in the landscape of social media. *American Journal of Business and Management*, vol. 4, no 3, pp. 102–122. DOI: 10.11634/216796061504668.
7. Kappelman L., Jones M.C., Johnson V., McLean E.R., Boonme K. (2016) Skills for success at different stages of an IT professional's career. *Communications of the ACM*, vol. 59, no 8, pp. 64–70. DOI: 10.1145/2888391.

8. Litecky C.R., Arnett K.P., Prabhakar B. (2004) The paradox of soft skills versus technical skills in is hiring. *Journal of Computer Information Systems*, vol. 45, no 1, pp. 69–76.
9. Havelka D., Merhout J.W. (2009) Toward a theory of information technology professional competence. *Journal of Computer Information Systems*, vol. 50, no 2, pp. 106–116.
10. Hussain W., Clear T., MacDonell S. (2017) Emerging trends for global DevOps: A New Zealand perspective. Proceedings of the *IEEE 12th International Conference on Global Software Engineering, Buenos Aires, Argentina, 22–23 May 2017* (ed. R. Bilof), vol. 1, pp. 21–30. . DOI: 10.1109/ICGSE.2017.16.
11. Wowczko I. (2015) Skills and vacancy analysis with data mining techniques. *Informatics*, vol. 2, no 4, pp. 31–49. DOI: 10.3390/informatics2040031.
12. Bailey J., Mitchell R.B. (2006) Industry perceptions of the competencies needed by computer programmers: Technical, business, and soft skills. *Journal of Computer Information Systems*, vol. 47, no 2, pp. 28–33.
13. Brooks N.G., Greer T.H., Morris S.A. (2018) Information systems security job advertisement analysis: Skills review and implications for information systems curriculum. *Journal of Education for Business*, vol. 93, no 5, pp. 213–221.
14. Casado-Lumbreras C., Colomo-Palacios R., Soto-Acosta P. (2015) A vision on the evolution of perceptions of professional practice. *International Journal of Human Capital and Information Technology Professionals*, vol. 6, no 2, pp. 65–78. DOI: 10.4018/IJHCITP.2015040105.
15. Föll P., Thiesse F. (2017) Aligning is curriculum with industry skill expectations: A text mining approach. Proceedings of the *25th European Conference on Information Systems, ECIS 2017, Guimarães, Portugal, 5–10 June 2017* (eds. I. Ramos, V. Tuunainen, H. Krcmar), pp. 2949–2959.
16. Stal J., Paliwoda-Pękosz G. (2019) Fostering development of soft skills in ICT curricula: A case of a transition economy. *Information Technology for Development*, vol. 25, no 2, pp. 250–274. DOI: 10.1080/02681102.2018.1454879.
17. Boselli R., Cesarini M., Mercorio F., Mezzanzanica M. (2018) Classifying online job advertisements through machine learning. *Future Generation Computer Systems*, vol. 86, pp. 319–328.
18. Colombo E., Mercorio F., Mezzanzanica M. (2019) AI meets labor market: Exploring the link between automation and skills. *Information Economics and Policy*, no 47, pp. 27–37. DOI: 10.1016/j.infoecopol.2019.05.003.
19. Karakatsanis I., AlKhader W., MacCrory F., Alibasic A., Omar M.A., Aung Z., Woon W.L. (2017) Data mining approach to monitoring the requirements of the job market: A case study. *Information Systems*, no 65, pp. 1–6. DOI: 10.1016/j.is.2016.10.009.
20. Lovaglio P.G., Cesarini M., Mercorio F., Mezzanzanica M. (2018) Skills in demand for ICT and statistical occupations: Evidence from web-based job vacancies. *Statistical Analysis and Data Mining*, vol. 11, no 2, pp. 78–91. DOI: doi.org/10.1002/sam.11372
21. Amato F., Boselli R., Cesarini M., Mercorio F., Mezzanzanica M., Moscato V., Picariello A. (2015) Challenge: Processing web texts for classifying job offers. Proceedings of the *2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015), Anaheim, California, USA, 7–9 February 2015* (eds. M.S. Kankanhalli, T. Li, W. Wang), pp. 460–463.
22. De Mauro A., Greco M., Grimaldi M., Ritala P. (2018) Human resources for Big Data professions: A systematic classification of job roles and required skill sets. *Information Processing & Management*, vol. 54, no 5, pp. 807–817. DOI: 10.1016/j.ipm.2017.05.004.
23. Gurcan F., Cagiltay N.E. (2019) Big data software engineering: Analysis of knowledge domains and skill sets using LDA-based topic modeling. *IEEE Access*, no 7, pp. 82541–82552.
24. Pejic-Bach M., Bertoncel T., Meško M., Krstić Ž. (2020) Text mining of industry 4.0 job advertisements. *International Journal of Information Management*, no 50, pp. 416–431.
25. Radovilsky Z., Hegde V., Acharya A., Uma U. (2018) Skills requirements of business data analytics and data science jobs: A comparative analysis. *Journal of Supply Chain and Operations Management*, vol. 16, no 1, pp. 82–101.

About the authors

Andrei A. Ternikov

Doctoral Student, Doctoral School on Economics;

Lecturer, Department of Management, St. Petersburg School of Economics and Management,
National Research University Higher School of Economics,
3, Kantemirovskaya Street, Saint-Petersburg 194100, Russia;

E-mail: aternikov@hse.ru

ORCID: 0000-0003-2354-0109

Ekaterina A. Aleksandrova

Cand. Sci. (Econ.);

Director, International Centre for Health Economics, Management, and Policy;
Associate Professor, Department of Economics, St.Petersburg School of Economics and Management;
Associate Professor, Department of Finance, St.Petersburg School of Economics and Management,
National Research University Higher School of Economics,
3, Kantemirovskaya Street, Saint-Petersburg 194100, Russia;

E-mail: ea.aleksandrova@hse.ru

ORCID: 0000-0001-7067-5087

DOI: 10.17323/2587-814X.2020.2.84.92

About the possibility of determining the prefix and suffix of a word by subwords of fixed length

Galina N. Zhukova[a]
E-mail: galinanzhukova@gmail.com

Yuri G. Smetanin[b]
E-mail: smetanin.iury2011@yandex.ru

Mikhail V. Ulyanov[c]
E-mail: muljanov@mail.ru

[a] National Research University Higher School of Economics
Address: 20, Myasnitskaya Street, Moscow 101000, Russia

[b] Federal Research Center "Computer Science and Control", Russian Academy of Sciences
Address: 40, Vavilova Street, Moscow 119333, Russia

[c] Trapeznikov Institute of Control Sciences, Russian Academy of Sciences
Address: 65, Profsoyuznaya Street, Moscow 117997, Russia

Abstract

In applied problems of business informatics related to data analysis (in particular, in the analysis and forecasting of time series, in the study of log files of business processes, etc.), problems of qualitative analysis arise. Qualitative analysis methods often use symbolic coding as a way of presenting information about the processes under study. In a number of situations, due to the fragmentation of such descriptions, the problem arises of reconstructing a complete symbolic description of a process (word) from its successive fragments (subwords). From the multiset of all subwords of a sufficiently large length, the original word is uniquely restored. In the case of insufficiently long subwords, several different reconstructions of the original word are possible. The number of feasible reconstructions can be reduced by determining the suffix and prefix of the reconstructed word. A method is proposed for determining the prefix and suffix of a word consisting of $k-1$ symbols each on the basis of multiset V of subwords of a fixed length equal to k. We accept the hypothesis that this multiset is generated by a window of a fixed length k of one symbol shift in an unknown word. The method for determining the prefix and suffix is based on the construction and analysis of the matrix formed by subwords from V written in rows in arbitrary order and the use of the operator acting on multisets of characters of the alphabet formed by neighboring columns of this matrix. The method is capable of determining the prefix $a_1a_2 \dots a_{k-1}$ and suffix $b_1b_2 \dots b_{k-1}$, if $a_i \neq b_i$ for any i from 1 to $k - 1$. If in the prefix and suffix $a_i \neq b_i$ only for some values of i, the characters in the corresponding positions are determined, and $a_j = b_j$ for the remaining characters. In the worst case, the method concludes that $a_i = b_i$ for any i from 1 to $k - 1$, but does not determine the characters themselves. This is a situation in which the prefix and suffix coincide but cannot be determined.

Key words: word reconstruction; prefix; suffix; multiset of subwords; subwords of fixed length; shift operator.

Citation: Zhukova G.N., Smetanin Yu.G., Ulyanov M.Yu. (2020) About the possibility of determining the prefix and suffix of a word by subwords of fixed length. *Business Informatics*, vol. 14, no 2, pp. 84–92. DOI: 10.17323/2587-814X.2020.2.84.92

Introduction

In the applied areas of business informatics related to data analysis, such as time series analysis and forecasting [1–6], research of business process log files [7], etc., problems of qualitative analysis arise. In this case, one of the commonly used methods for presenting information about processes is symbolic encoding [8]. Furthermore, a description of the behavior of a time series or a business process is encoded with a word over a finite alphabet which is the object of further research. However, in a number of cases, including the analysis of business processes and time series, researchers do not know the whole word, but a multiset of subwords that are consecutive fragments of a certain word. Since in this case the positions of the subwords in the original word are unknown, the problem of reconstruction arises, i.e. the restoration of the unknown word from the original set of subwords [9–17]. This problem relates to a special section of discrete mathematics, namely the combinatorics on words [18]. The objects of research in the combinatorics on words are words over arbitrary alphabets, and the subject of research is the study of the combinatorial properties of various sets of words, both finite and infinite. In real-life applied problems, information about words is often incomplete; for example, such a situation is inevitable in the analysis of infinite time series measured over finite time intervals.

We note that one of the important areas of practical application of the methods of combinatorics on words is the field of bio-molecular models and processes. At the same time, work with fragmentary information is characteristic of a number of bio-informatics and genomics problems. For example, the problem of sequencing genomes [19, 20] is essentially the problem of reconstructing words under conditions of strong restrictions, implying unique reconstruction.

The problems of reconstructing words over a finite alphabet have different statements, differing both in the amount of information available and in the restrictions on feasible solutions [21–23]. Usually, these problems, as problems with incomplete information, are complex, and obtaining any additional information obviously allows us to reduce the set of solutions under consideration.

In a qualitative analysis of time series [24, 25], the coding of the observed variable can be carried out in a certain alphabet, for example, (A, B, C, D, E, F), which symbols can be used to name half-segments of the observed values of the variable in the ascending order. For example, A is the name of the half segment of the smallest value, F is the largest. Since observations are recorded in discrete time, the description of the values of the time series by the names of half-segments is a word over the alphabet of names. If the observed process is characterized by sharp outliers of the observed value (up to level F) relative to the basal level (A, B) in one moment, as well as sharp drops (from F to B), then the resulting codewords of time series will not contain subwords CDE and EDC. In this case, if the initial data are subwords (scattered fragments of observations), then the problem of reconstructing a word from subwords is the problem of restoring the entire description of a time series under the assumption of the peculiarities of its behavior.

A similar situation arises when reconstructing business process log files in the presence of fragmented information. When describing business processes by the graph theory apparatus [7], a model (business process graph) can be represented as follows: process states are encoded by named vertices, and state transitions are encoded by edges identified with stages of the business process. Then the record of a particular implementation of a business process is a word over the alphabet of vertex names that reflects the state transition order. If the process is physically distributed between various organizations and executors, then most likely we will receive information about its complete performance in the form of a set of subwords. In addition, prohibited subwords can be interpreted as violations of the model (the regulation of the business process). The arising reconstruction problem, without forbidden subwords, means the possibility of a complete reconstruction of the entire process corresponding to the theoretical model.

Thus, it is of interest to study in detail the various versions of the word reconstruction problem with a certain set of subwords of shorter length, interpreted as a set of consecutive fragments of an unknown word. Moreover, of interest are both the case when the reconstructed word does not contain a predetermined forbidden subword and the case with the presence of forbidden subwords. One of the possible solutions to this problem, based on subwords of fixed length, in the shift by one symbol hypothesis, was proposed by the authors in [26, 27]. However, the set of possible reconstructions can be large and the problem arises of a possible reduction in the number of feasible solutions for the "correct" reconstructed word. We want to obtain additional information from the initial set of subwords, which will be useful in reducing the resulting set of reconstructions. We are talking about the possibility of restoring and / or determining the pattern of the prefix and suffix of an unknown word, which, as part of the reduction procedure, will lead to the consideration of only those words that have the obtained patterns of prefix and suffix. It is the problem that is the subject of this article.

1. Terminology and notation

Further in the text of the article, the following notation will be used:

$\Sigma = \{s_1, s_2, \dots s_l\}$ – alphabet where s_i is i-th symbol of the alphabet;

Σ^k – the k-fold Cartesian product (Cartesian product of set Σ, i.e. the set of k-element tuples);

$\Sigma^* = \bigcup_{k=0}^{\infty} \Sigma^k$ – transitive closure of Σ (the set of all possible tuples);

w – a word (above the alphabet), which is a sequence of characters of the alphabet, while the actual characters of the alphabet are words by definition;

$L(\cdot) : L(C\ ) = W$ where $C \subseteq \Sigma^*$ is a set of tuples, W is a set of words. Operator$L(\cdot)$ is an operator acting on a set of tuples; $L(\cdot)$ creates a set of words consisting of characters from Σ ;

a_i is i-th character of word w, $a_i \in \Sigma$;

$w = a_1 a_2 \dots a_n \in L(\Sigma^n)$ is an arbitrary word consisting of *n* characters of alphabet Σ;

$|w| = n$ – word length, defined as the number of its elements;

$L_k = L(\Sigma^k) = \{w \mid |w| = k\}$ – the set of all words of length *k* over alphabet Σ.

Let$w = a_1 a_2 \dots a_n \in L(\Sigma^n)$, and $k < n$, then

$v = a_{i_1} a_{i_2} \dots a_{i_k}, 1 \le i_1,\ i_2 = i_1 + 1,\ i_k = i_{k-1} + 1 \le n$ – a subword of a word *w* of length *k*;

$Q(w, i, k)$ is an operator that gives the subword of length *k* of word *w*, starting with a character in position *i*.

Let $|w| = n$, then the operator is defined for $i + k - 1 \le n$ so that

$$Q(a_1 a_2 \dots a_n, i, k) = a_i a_{i+1} \dots a_{i+k-1},$$

$$Q(w, i, k) \in L_k;$$

For the following two operators, we assume that$|w| = n \ge 2$ and $1 \le k < n$:

$P(w, k) = Q(w, 1, k) = a_1 a_2 \dots a_k \in L_k$ is the prefix of length *k* of word *w*;

$S(w, k) = Q(w, n - k + 1, k) = a_{n-k+1} \dots a_n \in L_k$ is the suffix of length *k* of word *w*;

$SH1(w, k)$ is a shift by one operator. The operator, defined when $|w| > k$, generates a set of subwords of length *k* (the cardinality of this set is $|w| > k + 1$), performing a shift of a window of length *k* along word *w*, starting from the leftmost position of word *w*:

$$SH1(w, k) = \{v_j | j = 1, |w| - k + 1;\ v_j = \\ = Q(w, i, k)\}.$$

2. Statement of the problem

Afterwards, we consider as a given: the length of the subword is *k*, the number of subwords is *m*, and the original multiset *V* of subwords over alphabet Σ, considered as the basis for the reconstruction of some unknown word *w*:

$$V = \{v_i \mid i = \overline{1, m};\ v_i = a_{i1} a_{i2} \dots a_{ik} \in L_k\}.$$

The hypothesis of shift one accepted by the authors states that *V* is a multiset of subwords of shift by one symbol alongside some unknown word *w*, where $|w| = n = m + k - 1$ and

$$V = SH1(w, k) = \\ = \{v_j \mid j = 1, n - k + 1;\ v_j = Q(w, j, k)\}.$$

Informal statement: Under the hypothesis of shift one, is it possible to determine the prefix and suffix of length $k - 1$ of the unknown word *w*, or to obtain any meaningful information about its prefix and suffix using only multiset *V*?

Mathematical statement: For a given multiset *V* with the length *k* of the subwords and the number of subwords equal to *m*, determine prefix $P(w, k - 1)$ and suffix $S(w, k - 1)$ of length $k - 1$ of the original word $w = a_1 a_2 \dots a_n$, and indicate the conditions under which a solution is possible.

3. Method for determining the prefix and suffix

First, we note that the main problem, both in the aspect of the reconstruction problem and in the aspect of the problem of determining the suffix and prefix, is that we were initially given a multiset of subwords *V*, but not a tuple of subwords. The main difficulty is connected with the loss of order in the original subwords obtained by the shift operator.

We begin the solution of this problem by constructing matrix *A* consisting of *m* rows and *k* columns whose rows are words v_i from set *V*. Words from set *V* can be represented in form $v_i = a_{i1}, a_{i2}, \dots a_{ik}$, and the elements of matrix *A* are the symbols of alphabet Σ, i.e. $A = (a_{ij})$, where a_{ij} is a symbol of the alphabet at the *j*-th position in the *i*-th word of multiset *V* in the order in which they are listed.

We explicitly write matrix A in the direct sequence of the window of shift by one symbol. Obviously, in reality, in the order of enumeration in multiset V, we will observe some permutation of words of the direct sequence, and, consequently, the corresponding permutation of the rows of matrix A:

$$A = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_{m-1} \\ v_m \end{pmatrix} = \begin{pmatrix} a_1, a_2, \ldots, a_k \\ a_2, a_3, \ldots, a_{k+1} \\ a_3, a_4, \ldots, a_{k+2} \\ \vdots \\ a_{n-k}, a_{n-k+1}, \ldots, a_{n-1} \\ a_{n-k+1}, a_{n-k+2}, \ldots, a_n \end{pmatrix}.$$

The solution to the problem of determining the prefix and suffix is based on the analysis of neighboring columns of this matrix. Let us consider the first and second columns. In each of them, at any permutation of rows, there will be symbol a_2 that is the second symbol of the unknown word w, and symbol a_3 that is the third symbol of w, etc. If the matching pairs of characters are deleted from these two columns, only a_1 and a_{n-k+2} remain if they are not equal. If they are different, we get their exact values. If a_1 and a_{n-k+2} coincide, then all characters in these columns will be crossed out, and we will get information that unknown, but coincident characters are in the corresponding positions of the prefix and suffix. We continue such an analysis for all $k - 1$ pairs of neighboring columns of matrix A. Provided that after crossing out pairs of matching characters we always have a mismatched pair, we will restore the prefix and suffix of length $k - 1$ of the unknown word w.

We describe the method formally.

We introduce a tuple of all symbols of the alphabet for which multiplicities of elements are allowed

$$C = \left(s_1^{(\alpha 1)}, s_2^{(\alpha 2)}, \ldots, s_l^{(\alpha l)}\right),$$

where multiplicity 0 yields an empty set $s_i^{(0)} = \varnothing$ in this position. We define operator G acting on the i-th column of matrix A and creating tuple C_i containing, for all characters of the alphabet, their multiplicity in accordance with the number of characters in this column

$$GC(A, i) = C_i = \left(s_1^{(\alpha 1)}, s_2^{(\alpha 2)}, \ldots, s_l^{(\alpha l)}\right).$$

We apply operator G to two columns of matrix A, and denote:

$$GC(A, i) = C_i = \left(s_1^{(\alpha 1)}, s_2^{(\alpha 2)}, \ldots, s_l^{(\alpha l)}\right),$$

$$GC\,(A, k) = C_k = \left(s_1^{(\beta 1)}, s_2^{(\beta 2)}, \ldots, s_l^{(\beta l)}\right).$$

We introduce operator GS of obtaining a character that acts on two tuples of columns of matrix A according to the following rule:

$$GS(A,i,k) = \begin{cases} \bigcup\limits_{j=1}^{l} s_j^{(\alpha_j - \beta_j)}, \left(s_j^{(\alpha_j)} \in GC(A,i)\right), \\ \qquad\qquad s_j^{(\beta_j)} \in GC(a,k), \\ s_j^{(\alpha j - \beta_j)} = \varnothing, \text{если}\; a_j - \beta_j \le 0. \end{cases}$$

Now apply operator GS to two consecutive columns of matrix A. Due to the structure of successive columns of matrix A described above, the result of action of operator GS will be either a symbol or an empty set. Note that if $GS(A, i, i + 1) \ne \varnothing$, then $GS(A, i + 1, i) \ne \varnothing$ too. In this case, we define the $n - k + i$-th prefix character $a_i = GS(A, i, i + 1)$ and the $n - k + i$-th character $a_{n-k+i} = GS(A, i+1, i)$ of the unknown word, which is the i-th character of suffix of length $k - 1$.

For example, if $GS(A, 1, 2) = s_i$, then we know the first character $a_1 = s_i$ of the unknown word w (the first character of the prefix) and, in this situation, the value $GS(A, 2, 1)$ is not necessarily an empty set. Let $GS(A, 2, 1) = s_j$, and we get the first character $a_{n-k+2} = s_j$ of the suffix. If $GS(A, 1, 2) \ne \varnothing$, then, it is obvious that $GS(A, 2, 1) \ne \varnothing$ too and we get information that $a_1 = a_{n-k+2}$, but at the same time the symbol of the alphabet itself at these positions remains unknown to us.

Since we have $k-1$ consecutive pairs of columns, then if for each consecutive pair of columns operator *GS* gives a non-empty set, then using the "+" operation to indicate the concatenation of characters, we get the solution:

$$P(w,k-1)=a_1a_{2.}\dots a_{k-1}=\sum_{i=1}^{k-1}GS(A,i,i+1),$$

$$S(w,k-1)=a_{n-k+2}\dots a_n=\sum_{i=1}^{k-1}GS(A,i+1,i).$$

If for each pair operator *GS* yields an empty set, then the prefix and suffix characters remain unknown, but at the same time, we get information about their equality as subwords

$$P(w,k-1)=S(w,k-1).$$

In a general case, we get information about prefix and suffix characters in the form of some pattern, and if these are specific characters, then they are located at the same positions of the prefix and suffix, and if the characters cannot be determined, then we have information about that at these positions the prefix and suffix characters match.

Let us give an example for word $w = abbaaabb$ in alphabet $\Sigma = \{a, b\}$ and the set of subwords obtained by the shift to one symbol operator with a window of width three. In this case, $k=3$, $m=6$, $n=8$, and matrix A has the form:

$$A=\begin{pmatrix} abb \\ bba \\ baa \\ aaa \\ aab \\ abb \end{pmatrix}.$$

Applying operator G to the three columns of matrix A gives the following tuples:

$$GC(A,1)=C_1=(a^{(4)},b^{(2)}),$$

$$GC(A,2)=C_2=(a^{(3)},b^{(3)}),$$

$$GC(A,3)=C_3=(a^{(3)},b^{(3)}).$$

and we get$GC(A,1,2)=a$, $GC(A,2,1)=b$, and $GC(A,2,3)=GC(A,3,2)=\varnothing$. Thereby, we get the prefix pattern $P(w,2)=a^*$ of length two of word $w = abbaaabb$, and the suffix pattern $S(w,2)=b^*$, where symbol $*$ denotes an unknown but matching symbol in the corresponding positions of the prefix and suffix (in fact, this is the symbol "*b*").

4. Application to the reconstruction problem

In one of the previous articles [26], the authors proposed a solution to the problem of complete reconstruction, under the conditions of a given multiset of subwords and one shift hypothesis. In some cases, the number of reconstructions determined by the number of Euler paths or cycles in the corresponding de Brain multi-graph can be significant [26].

Let us introduce the set of possible word reconstructions by the initial set V:

$$W=\{(w|\ |w|=m,\ k-1=n,\ V=SH1(w,k)\},$$

In this case, if $|W|\geq 2$, then reconstruction is possible and there can be many of them. Let w^* be the word under consideration, that is unknown to us, based on which the set V is obtained, where $V = SH1(w^*, k)$. Then when choosing a possible reconstruction in set W, we select only those words that possess the prefix and suffix obtained by the operator *GS*, taking into account patterns with possibly unknown characters. As a result we obtain

$$\tilde{W}=\left\{\begin{matrix}(w|\ P(w,k-1)=\sum_{i=1}^{k-1}GS(A,i,i+1), \\ S(w,k-1)=\sum_{i=1}^{k-1}GS(A,i+1,i)\end{matrix}\right\},$$

where $w^*\in\tilde{W}$ is guaranteed.

This leads to a reduction in the resulting set of reconstructions, since we consider only those words that have the given prefix and suffix patterns. Moreover, this approach can be applied not only to reduce a finite set of reconstructions, but to consider the prefix as a pattern for choosing the initial arcs for the Euler paths in the de Brain multi-graph when constructing the reconstruction [26].

Conclusion

In this article, in the aspect of solving the problem of reconstruction of symbolic descriptions of time series and logs of business processes, a solution to the problem of determining the prefix and suffix of an unknown word is proposed. The solution is based on the assumption that the full set of subwords of fixed length k, originally generated by the window of length $k$going alongside an unknown word with a shift to one symbol, is initially given. A solution has been obtained that allows us to acquire information about the prefix and suffix of an unknown word or some patterns for the prefix and suffix. The proposed solution allows us to obtain additional information about possible reconstructions, and thereby reduce the number of possible word reconstructions for a given set of subwords. In the best case, the proposed method allows us to determine the prefix and suffix of length k of an unknown word, or, in the worst case, to state that the prefix and suffix coincide.

The results can be used in conjunction with solving the reconstruction problem [26, 27] to reduce the set of possible reconstructions during qualitative analysis in such problems of business informatics as analysis of time series and logs of business processes. ■

Acknowledgments

The study reported here was funded by RFBR, project number 19-07-00150.

References

1. Ding H., Trajcevski G., Scheuermann P., Wang X., Keogh E. (2008) Querying and mining of time series data: Experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, vol. 1, no 2, pp. 1542–1552. DOI: 10.14778/1454159.1454226.

2. Kurbalija V., Radovanović M., Geler Z., Ivanović M. (2011) The influence of global constraints on DTW and LCS similarity measures for time-series databases. *Advances in Intelligent and Soft Computing*, vol. 101, pp. 67–74. DOI: 10.1007/978-3-642-23163-6_10.

3. Wu Y.-L., Agrawal D., el Abbadi A. (2000) A comparison of DFT and DWT based similarity search in time-series databases. Proceedings of the *Ninth International Conference on Information and Knowledge Management (CIKM '00), McLean, VA, 6–11 November 2000*, pp. 488–495.

4. Bemdt D.J., Clifford J. (1994) Using dynamic time warping to find patterns in time series. *AAAI-94 Workshop on Knowledge Discovery in Databases*, pp. 359–370. Available at: https://www.aaai.org/Papers/Workshops/1994/WS-94-03/WS94-03-031.pdf (accessed 15 March 2020).

5. Dreyer W., Dittrich A.K., Schmidt D. (1994) Research perspectives for time series management systems. *SIGMOD Record*, vol. 23, no 1, pp. 10–15.

6. Keogh E.J., Pazzani M.J. (1998) An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. Proceedings of the *Fourth International Conference on Knowledge Discovery and Data Mining (KDD'98), New York, 27–31 August 1998*, pp. 239–241.

7. Andersen B. (1999) *Business processes improvement toolbox*. New York: ASQ Quality Press.

8. Lin J., Keogh E., Wei L., Lonardi S. (2007) Experiencing SAX: A novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, vol. 15, no 2, pp. 107–144. DOI: 10.1007/s10618-007-0064-z.

9. Acharya J., Das H., Milenkovic O., Orlitsky A., Pan S. (2014) String reconstruction from substring compositions. *SIAM Journal on Discrete Mathematics*, vol. 29, no 3, pp. 1340–1371.

10. Manvel B., Meyerowitz A., Schwenk A., Smith K., Stockmeyer P. (1991) Reconstruction of sequences. *Discrete Mathematics*, vol. 94, no 3, pp. 209–219. DOI: 10.1016/0012-365X(91)90026-X.

11. Carpi A., de Luca A. (2001) Words and special factors. *Theoretical Computer Science*, vol. 259, no 1–2, pp. 145–182.

12. de Luca A. (1999) On the combinatorics of finite words. *Theoretical Computer Science*, vol. 218, no 1, pp. 13–39.

13. Dudík M., Schulman L.J. (2003) Reconstruction from subsequences. *Journal of Combinatorial Theory. Series A*, vol. 103, no 2, pp. 337–348. DOI: 10.1016/S0097-3165(03)00103-1.

14. Erdős P.L., Ligeti P., Sziklai P., Torney D.C. (2006) Subwords in reverse-complement order. *Annals of Combinatorics*, vol. 10, no 4, pp. 415–430. DOI: 10.1007/s00026-006-0297-3.

15. Fici G., Mignosi F., Restivo A., Sciortino M. (2006) Word assembly through minimal forbidden words. *Theoretical Computer Science*, vol. 359, no 1–3, pp. 214–230. DOI: 10.1016/j.tcs.2006.

16. Levenshtein V.I. (2001) Efficient reconstruction of sequences from their subsequences or supersequences. *Journal of Combinatorial Theory, Series* A, Vol. 93, pp. 310–332.

17. Piña C., Uzcátegui C. (2008) Reconstruction of a word from a multiset of its factors. *Theoretical Computer Science*, vol. 400, no 1–3, pp. 70–83. DOI: 10.1016/j.tcs.2008.01.052.

18. Lothaire M. (2002) *Algebraic combinatorics on words*. Cambridge, UK: Cambridge University Press.

19. Gusfield D. (1997) *Algorithms on strings, trees, and sequences: Computer science and computational biology*. Cambridge, UK: Cambridge University Press.

20. Skiena S.S., Sundaram G. (1995) Reconstructing strings from substrings. *Journal of Computational Biology*, vol. 2, no 2, pp. 333–353.

21. Leont'ev V.K., Smetanin Y.G. (2002) Problems of Information on the set of words. *Journal of Mathematical Sciences*, vol. 108, no 1, pp. 49–70. DOI: 10.1023/A:1012705332306.

22. Levenshtein V.I. (1997) Restoring objects based on the minimum number of distorted samples. *Doklady Akademii Nauk*, vol. 354, no 5, pp. 593–596 (in Russian).

23. Krasikov I., Roditty Y. (1997) Note: On a reconstruction problem for sequences. *Journal of Combinatorial Theory, Series A*, no 77, pp. 344–348.

24. Ulyanov M.V., Smetanin Yu.G. (2013) Determining the characteristics of Kolmogorov complexity of time series: An approach based on symbolic descriptions. *Business Informatics*, no 2, pp. 49–54 (in Russian).

25. Smetanin Yu.G., Ulyanov M.V. (2014) Measure of symbolical diversity: Combinatorics on words as an approach to identify generalized characteristics of time series. *Business Informatics*, no 3, pp. 40–46 (in Russian).

26. Smetanin Yu.G., Ulyanov M.V. (2014) Reconstruction of a word from a finite set of its subwords under the unit Shift hypothesis. I. Reconstruction without for bidden words. *Cybernetics and Systems Analysis*, vol. 50, no 1, pp. 148–156.

27. Smetanin Yu.G., Ulyanov M.V. (2015) Reconstruction of a word from a finite set of its subwords under the unit Shift hypothesis. II. Reconstruction with forbidden words. *Cybernetics and Systems Analysis*, vol. 51, no 1, pp. 157–164. DOI: 10.1007/s10559-015-9708-y.

About the authors

Galina N. Zhukova

Cand. Sci. (Phys.-Math.);

Associate Professor, School of Software Engineering, Faculty of Computer Science,
National Research University Higher School of Economics,
20, Myasnitskaya Street, Moscow 101000, Russia;

E-mail: galinanzhukova@gmail.com

ORCID: 0000-0003-1835-7422

Yuri G. Smetanin

Dr. Sci. (Phys.-Math.);

Chief Researcher, Federal Research Center “Computer Science and Control”,
Russian Academy of Sciences,
40, Vavilova Street, Moscow 119333, Russia;

E-mail: smetanin.iury2011@yandex.ru

ORCID: 0000-0003-0242-6972

Mikhail V. Ulyanov

Dr. Sci. (Tech.);

Leading Researcher, Laboratory of Scheduling Theory and Discrete Optimization,
V.A. Trapeznikov Institute of Control Sciences, Russian Academy of Sciences,
65, Profsoyuznaya Street, Moscow 117997, Russia;

E-mail: muljanov@mail.ru

ORCID: 0000-0002-5784-9836