Bulletin of Electrical Engineering and Informatics
Vol. 12, No. 1, February 2023, pp. 502~513
ISSN: 2302-9285, DOI: 10.11591/eei.v12i1.4055

Neuro-fuzzy-based mathematical model of dispatching of an industrial railway junction

Leyla M. Bogdanova[1], Sergey Ya Nagibin[2], Dmitry I. Loskutov[2], Natalia A. Goncharova[2]
[1]Department of Research Association, Plekhanov Russian University of Economics, Moscow, Russia Federation
[2]Moscow Aviation Institute (National Research University), Moscow, Russia Federation

Article Info

Article history:

Received Jul 6, 2022
Revised Aug 24, 2022
Accepted Oct 14, 2022

Keywords:

Fuzzy systems design
Identification of nonlinear dependencies
Neuro-fuzzy network
Priority queue
Sugeno algorithm
Task of finding the optimal sequence of operations

ABSTRACT

In any transport system, especially at industrial railway junctions, it is fundamentally important to build an effective timetable (traffic schedule) to regulate traffic flows. The task is complicated by the high dimensionality of the railway network of the node, the large number of variable parameters associated with scheduling the use of a traction resource (locomotives) during operation for sorting wagons and transporting payloads (ore, fuel, finished products and empty wagons). The problem is that most plotting problems are NP-hard, i.e. the algorithms for solving them, used to automate the process, may require an unacceptably long execution time by traditional methods of solving this problem (sequential, using reference information; method of thread laying). The article deals with the issues of building a mathematical model for dispatching an industrial railway junction to minimize the time of using locomotives in order to increase the efficiency of its operation. The mathematical model uses the technique of neuro-fuzzy computing to determine the parameters for identifying fuzzy systems and calculating the priorities of operations in the framework of creating a flexible schedule for the decision support system of the dispatching service. The results of modeling and recommendations on the use of the developed methodology are presented.

Corresponding Author:

Sergey Ya Nagibin
Moscow Aviation Institute (National Research University)
Volokolamskoye Shosse, Moscow, Russian Federation
Email: nagibinmai@mail.ru

1. INTRODUCTION

The digital transformation of the economy is also transforming the logistics industry. The main reason for this transformation is the process of transferring functions and business processes previously performed by people to the digital environment, as well as the requirements imposed by new technologies and equipment that traditional management technologies are not able to provide. Modern requirements for the quality of logistics processes are becoming more complex. One of the current trends is the attraction of new and more efficient equipment and technologies. Internet of things (IoT), radio-frequency identification tags (RFID tags), artificial intelligence (AI), neural networks and gadgets make it possible to automate logistics procedures and develop "seamless" transportation procedure, during which goods are not delayed at stages, which speeds up delivery and reduces its cost.

The introduction of IT technologies in logistics processes is not a new trend [1]. The development and implementation of IT technologies in the organization of logistics automation at industrial railway junctions, which, as a rule, have several stations serving industrial enterprises, is very actual. Schemes of such nodes depend on the location of enterprises in relation to the main railway line, size of the freight turnover, nature of the enterprises and local conditions. Railway junctions in the areas of metallurgical, mining and processing and a number of other industrial enterprises are complex multidimensional systems: complex topology of tracks, significant number of entrances and exits, dead ends, depots, traffic lights, arrows, intersections, unloading and loading stations, empty and loaded cars, mass arrival and departure of goods. So, for example, the industrial railway junction of the magnitogorsk iron and steel works contains more than 30 stations and one day of its downtime costs about $170 000. Therefore, the organization of logistics to ensure business continuity at such nodes is relevant and is scientificly and practically interesting.

One of the most costly components in the process of organizing the operation of such a unit is the use of the traction resource located on it (locomotives) for sorting wagons and transporting payloads (ore, fuel, finished products and empty wagons). The analysis of the methods of scheduling theory, as applied to an industrial railway junction, showed that the problem under consideration belongs to the type of machine scheduling problems of scheduling theory, which considers the assignment of requirements to executors. In this class of problems, there are various objective functions, the approaches to solving which are very different, but the problem of minimizing the number of performers is not considered in the classical scheduling theory [2], [3].

General scheduling problems are demonstrational, that is, simple in terms of optimization. In this article, a fairly large number of criteria are taken into account, which significantly increases the dimension of the problem. A separate computational load is dynamism of the graph (passing transit trains, blocking) and change in the state (temporary unavailability of edge and node complexes) of the graph due to the parallel operation of other systems [4], [5].

Another problem is related to the dimension of the problem. It was shown in [6] that discrete optimization problems for scheduling theory problems with more than 60 jobs are NP-hard and extremly NP-hard problems in terms of scheduling theory. When constructing the requirements graph, the initial positions of the requirements fulfillment and their duration are set as constants, but the duration of the locomotive's movement from the end of the fulfillment of one requirement to the beginning of the fulfillment of another must be calculated based on the current state of the graph. In addition, it is necessary to take into account the parallel movement of other locomotives and trains (current and planned), temporary parking of trains in settling tanks, and possible blocking of tracks for maintenance, that is, it is necessary to take into account the dynamism of the graph (classical problem of finding the shortest path and its known solution algorithms, for example, Dijkstra’s algorithm [7], work only with static graphs) [8], [9]. It is also necessary to take into account the fact that the classical problem of finding the shortest path considers the movement of a material point along a graph, while trains can have a length that is sometimes many times greater than the length of individual railway sections, thus being simultaneously on several tracks.

In addition, motion planning for several pieces of equipment simultaneously is not considered in classical graph theory. The methods of Levenberg-Marquardt, gradient descent, as well as the methods of nonlinear regression analysis used to calculate the schedule, make it possible to bring the calculated data closer to the experimental ones, but at the same time lead to a number of problems: firstly, they give significant errors when trying to predict the behavior of the "priority function "; secondly, such methods for the most part work with functions of a given type, while in this case the function is generally unknown [9], [10]. The purpose of this article is to increase the availability of resources and capacity of a transport railway graph with a complex topology to ensure business continuity of an industrial enterprise by developing a mathematical dispatching model that includes a mathematical model for optimizing the schedule for using transport resources and monitoring their execution. To achieve this goal, it is necessary to solve the following tasks: i) to conduct a study of the methods of resource availability theory schedule and methods of formation of optimal train routes; ii) to develop the optimal sequence of technological operations execution (queue of dispatching operations); and iii) to develop a method for calculating dynamic priorities for each task.

2. THEORETICAL BASIS

To determine the optimal sequence of technological operations, the developed model (and its implementation, software module) forms a queue of dispatching operations, taking into account the calculated priorities for their implementation. That is, when an element is included, a search is made for the place where it will be included, and then the inclusion procedure occurs. With an exception, the element at the beginning will be removed [11].

The priority queue is directly used in the presented mathematical model. When using constant values as priorities, the following problem arises: with the constant arrival of more significant tasks, the time of tasks with a lower priority will constantly increase, which can lead to non-execution of the latter. The concepts of "priority of the dispatching task" and "waiting time in the queue" are fuzzy from the point of view of logic. Therefore, this article developed a method for finding dynamic priorities that reduces downtime, which is implemented in a neuro-fuzzy module. The process of finding the dependence of the final priority (hereinafter referred to as the "priority function") for placing the operation in the dispatcher's queue for execution is complicated by the fact that, in addition to the parameters, it is also necessary to find the general form of the function [12]–[14].

Until now, the task of constructing an adaptive and efficient schedule has not been implemented even in areas of less complexity than an industrial railway junction. For specific cases of scheduling on a given section, the algorithms are reduced to enumeration of all possible cases with rejection of the branch of incorrect decisions [15]. The problem is that most of the graphing problems are NP-hard, i.e. algorithms for solving them, used to automate the process, may require an unacceptably long execution time [2]–[4], [16].

At the same time, the ability of artificial neural networks to self-learn and solve weakly formalized problems is known, which can be used to apply the theory of neural networks to work with problems of scheduling theory. Currently, a large number of artificial neural networks are known: perceptrons, recurrent networks, convolutional neural networks (CNN), deconvolutional neural networks (DNN), ResNet, and long short term memory network (LSTM). However, they work well in the field of image recognition, face recognition, speech recognition, and translation [17]–[23].

The problems of scheduling using artificial neural networks for static processes with equal duration of any process were studied in [17], [21], [22], [24]. The main problematic issues of these solutions are the inability to establish a correspondence between the meaningful aspects of the scheduling problem, the terms of the neural network energy function and the issue of calculating the coefficients of the equation of this function. These solutions are multistage: at the first stage, one of the heuristic algorithms is used to obtain a basic solution, then it is converted to a form suitable for inputting an artificial hopfield neural network (provided that there are no restrictions on the order of the tasks to be executed), the result of which, later is processed by local optimization algorithms.

Hopfield networks [21], [22] in various modifications are used in scheduling problems, but only for those cases where only the final distribution of resources is important (available/not available). This method cannot be used when it is important how the state of these resources changed, if the state of the resource at the selected point in time should be equivalent to the state of the previous resource at the past point in time, and if the sequence data is required not only to remember, but also to shift in two-dimensional space (for example, in space "time-distance"). It is known that solutions based on the use of artificial neural networks often turn out to be suboptimal, but they are satisfactory in terms of the ratio of time spent and the required computing power, and in terms of the quality of the solutions obtained, they outperform strict classical formulations. In this paper, since the method of finding dynamic priorities is implemented in a neuro-fuzzy module, this is what led to the choice for further work of an adaptive network based on the Takagi-Sugeno fuzzy inference system adaptive neuro-fuzzy inference system (ANFIS).

2.1. Formal statement of the problem

At the beginning of the execution of planning operations, M requirements are given with given execution durations and positions on the station graph and N locomotives. We represent the set of requirements in the form of graph (Figure 1), the vertices of which are the requirements, the weight of the vertices is the duration of the requirement, and the weight of the edges is the duration of the move from the end of one requirement to the beginning of another. Moreover, each two requirements are connected by a pair of edges, since the path from the starting point to the end point and vice versa is not always the same (due to the dynamics of the graph).

To compile a sequence of requirements fulfilled by locomotives, we divide the original graph of requirements into directed subgraphs consisting of sequentially connected vertices, and the length of the path in the subgraph will be considered the duration of the locomotive (Figure 2). To minimize the number of units of traction stock involved in the schedule, it is necessary to divide the requirements graph into subgraphs in such a way that: i) duration of operation of each locomotive did not exceed the specified schedule length (parameter that can be associated with the length of the work shift, length of the day); ii) number of selected subgraphs was minimal; and iii) there are no vertices left that are not included in any subgraph.

The priority (of execution) of the dispatching task is the value determined on the interval (0, 1) and characterizing the degree of urgency of the execution of this task. A priority queue is a sequence of elements in which an element is included or excluded according to their priorities. A priority-exclusion queue is a queue in which elements are included at the end, and when excluded, the element with the highest priority is searched for. A priority inclusion queue is a queue in which a sequence of elements is kept ordered at all times in descending order of priority. The developed software package generates a graph of the structure of the railway junction and, on its basis, builds a model for multi-stage optimization of the functioning of the locomotive and wagon fleets: i) at the first stage, minimization of the maximum time for the train to pass through the railway network of the junction is carried out and ii) at the second stage, the neuro-fuzzy module for calculating the priorities for the execution of dispatcher tasks forms the optimal sequence for the execution of technological operations.

Logistics dispatching consists in planning the sequence of work performed, monitoring their execution and developing control actions in case of their deviation from the plan [1], [25], [26]. The main goals of this work are to queue operations in order to optimize costs, execution time or minimize servicing devices. Dispatching is especially relevant when the service intensity is high.

For railway junctions, tracking the location of all elements of the rolling stock (locomotives, empty and loaded cars), as well as state of arrows, semaphores and occupied tracks becomes additional complexity. Since the railway tracks are located close to each other and traffic on them is possible in both directions. To link the coordinates of the rolling stock to the topology of the station's tracks, RFID technologies, GPS, and Glonass are used.

We give sets of locomotives, lists of operations, station graph and duration of the work plan. It is necessary to distribute the locomotives in such a way as to use the minimum number of them. We reduce the problem to a linear programming problem (Figure 3). Input parameters are the following: i) information about all tasks (currently available from operations); ii) information about all available or active locomotives; iii) station graph; iv) plan duration; v) output parameters; and vi) task/locomotive pairs. Output parameters: task/locomotive pairs. Limitations are the following: tasks in operations are performed strictly sequentially.

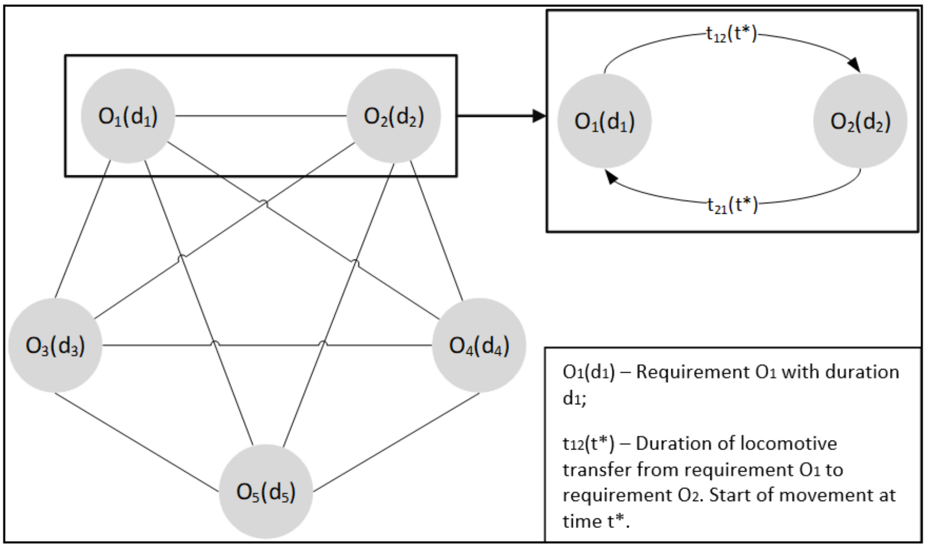

Figure 1. Representation of a set of requirements in the form of graph

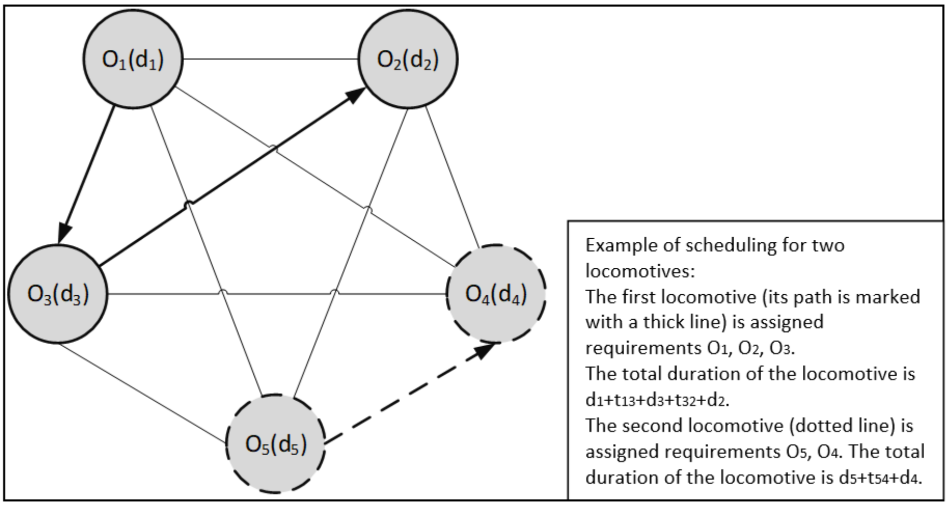

Figure 2. Example of generating schedules for locomotives

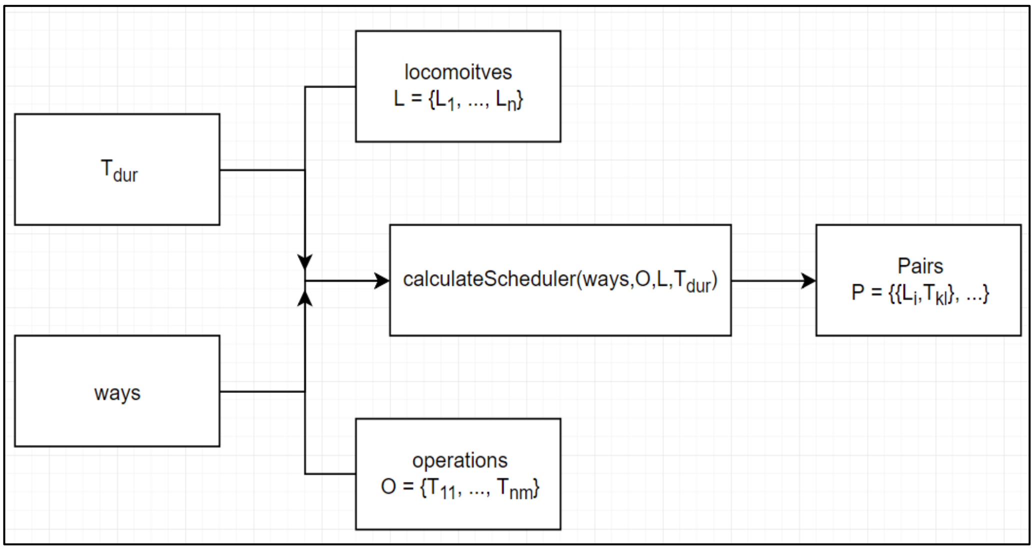

Figure 3. Functional scheme of the problem solution

Where:
T_{dur} is the plan duration;
$ways$ is the station count;
$locomotives$ are all available locomotives;
$operations$ are the operations for which we need to calculate the schedule;
$Pairs$ is the vector in which all operations are distributed and locomotives are assigned;
t_{ij}^{k} is the task priority;
N is the number of used locomotives;
k is the iteration number of the algorithm;
Op_{dur} is the duration of all operations;
X is the vector of input signals for configuring Sugeno knowledge base;
y is the corresponding output value to the vector X;
M is the number of pairs of experimental data for configuring the knowledge base is increased;
$\mu_j(x_i)$ is the membership function of some fuzzy set;
c_{ij} and σ_{ij} are the parameters of the membership function;
β_{rj} is the relative degree of fulfillment of the *j*-th rule for the input vectorX_r.

It is necessary to minimize the target function (1):

$$F = \sum_k \sum_{i,j} t_{ij}^{k} x_{ij}^{k} \rightarrow min, \quad (1)$$

Where:
$x_{ij}^{k} \in \{0,1\}$,
$x_{ij}^{k} \in \{0,1\}$,
$\sum_j x_{ij}^{k} \leq 1$,
$\sum_{i,j} x_{ij}^{k} = N$

In order to understand the required number of locomotives to perform all operations, it is necessary to evaluate the number of required locomotives (2):

$$N = \left\lceil \frac{Op_{dur}}{T_{dur}} \right\rceil + 1, \quad (2)$$

The weak point in the above scheme is the static nature of the parameter t_{ij}^{k}. When using constant values as priorities ones, we have the problem of constantly postponing less urgent, but already added tasks for a long time, which can lead to their non-fulfillment with the constant receipt of more significant operations. To find the "golden mean" in the work, a method for calculating dynamic priorities is proposed, which reduces downtime in work.

3. METHOD

3.1. Method for calculating dynamic priorities

The neuro-fuzzy module calculates the final priority of the task based on two values: waiting time of the task in the queue and task's own priority (calculated using the proposed mathematical model). In the process of adding a new technological operation to the queue, its tasks are assigned an initial priority, based on the urgency of performing this particular operation. After placing this operation in the queue, the waiting time starts counting, which affects the subsequent recalculation of the priority. This dynamic process of calculating the priority value allows optimizing the order of execution of technological tasks.

The process of finding the dependence of the final priority on the waiting time in the queue of this operation and the task's own priority is complicated by the fact that in addition to the parameters of the dependency functions, it is also necessary to find the general form of this function itself. The most well-known algorithms, such as, for example, methods of nonlinear regression analysis, allow bringing the calculated data closer to the experimental ones, but at the same time lead to a number of problems. Firstly, they give large errors when trying to predict the behavior of the system, and secondly, such methods mostly work with functions of a given type, while the type of function in our case is unknown. In addition, the concepts of "priority of a technological operation" and "waiting time in a queue" are fuzzy, from the point of view of logic, therefore, it is proposed to use the method of two-stage identification of nonlinear dependencies using fuzzy knowledge bases, based on the works of Rothstein [27], [28], the main provisions of which can be formulated as follows:

a. "The principle of linguistics of input and output variables". According to this principle, the inputs of an object and its output are considered as linguistic variables that are evaluated by qualitative terms.
b. "The principle of forming the structure of the input-output dependence in the form of a fuzzy knowledge base". A fuzzy knowledge base is a set of rules, IF <inputs>, THEN <output>, which reflect the qualification of an expert specialist and his competence in analyzing cause-and-effect relations in the analyzed decision-making task (management, diagnostics, and forecasting).
c. "The principle of two-stage configuration of fuzzy knowledge bases". In accordance with this principle, the construction of a model of a nonlinear object is carried out in two stages, which are called "structural and parametric identification". The configurable parameters are the weights of the fuzzy if-then rules and the forms of the membership functions.

Structural identification is the formation of a fuzzy Sugeno knowledge base [29], [30], the parameters of which are adjusted according to experimental data: for the dependence model $y = f(X)$, where X is vector of input signals, y is corresponding to the output value, there is a training sample of M pairs of experimental data connecting the inputs $X_r = (x_{r,1}, x_{r,2}, \dots, x_{r,n})$ with the output of the studied dependence $(X_r, y_r), r = \overline{1, M}$ (Table 1).

Table 1. Dependence of experimental data for the formation of Sugeno knowledge base

Input vector				Output value
$x_{1,1}$	$x_{1,2}$	…	$x_{1,n}$	y_1
…	…	…	…	…
$x_{r,1}$	$x_{r,2}$	…	$x_{r,n}$	y_r
…	…	…	…	…
$x_{M,1}$	$x_{M,2}$	…	$x_{M,n}$	y_M

The following fuzzy Sugeno knowledge base is known: «If $x_1 = \tilde{a}_{1j}$ and $x_2 = \tilde{a}_{2j}$, … , $x_n = \tilde{a}_{nj}$, then $y = b_j$, $j = \overline{1, m}$, where b_j are some real numbers». For the initial filling of the knowledge base, we suggest using each term of the input variable in only one rule. To obtain differentiable input-output relations, we will use Gaussian membership functions (3):

$$\mu_j(x_i) = exp\left(-\frac{1}{2}\left(\frac{x_i - c_{ij}}{\sigma_{ij}}\right)^2\right), \tag{3}$$

where $\mu_j(x_i)$ is the membership function of a fuzzy set $\tilde{a}_{ij}$; c_{ij} and σ_{ij} are the parameters of the membership function. Input vector $X_r = (x_{r1}, \dots, x_{r3})$ corresponds to the following result of the fuzzy output (4):

$$y_r^f = \beta_{rj} \sum_{j=1}^{m} (b_0 + b_{j1} x_{r1} + \dots + b_{jn} x_{rn}), \tag{4}$$

where β_{rj} is the relative degree of fulfillment of the j-th rule for the input vector X_r. When using multiplication as the *t*-norm, the relative degree of rule fulfillment is calculated as (5):

$$\beta_{rj} = \frac{exp\left(-\frac{1}{2}\sum_{i=1}^{n}\left(\frac{x_{ri}-c_{ij}}{\sigma_{ij}}\right)^2\right)}{\sum_{k=1}^{m}\left(-\frac{1}{2}\sum_{i=1}^{n}\left(\frac{x_{ri}-c_{ik}}{\sigma_{ik}}\right)^2\right)}. \tag{5}$$

The second stage is the parametric identification of the studied dependence by finding such parameters that minimize the deviation of the results of fuzzy modeling from the experimental data. The coefficients in the conclusions of the rules will be the configurable parameters for identification $B = (b_1, \dots, b_m)$ and the parameters of the membership functions of fuzzy terms: coordinates of the maxima $C = (c_{11}, \dots, c_{n1}, c_{22}, \dots, c_{n2}, \dots, c_{1m}, \dots, c_{nm})$ and concentration coefficients $S = (\sigma_{11}, \dots, \sigma_{n1}, \sigma_{22}, \dots, \sigma_{n2}, \dots, \sigma_{1m}, \dots, \sigma_{nm})$. The task of tuning is to find the vector *(B, C, S)*, such that the target functionality reaches a minimum (6):

$$J(B,C,S) = \sqrt{\frac{1}{M}\sum_{r=\overline{1,M}}(y_r - F(B,C,S,X_r))^2} \to min. \tag{6}$$

Finding partial derivatives of errors $E^{<r>}$ by managed variables b_j, c_{ij}, σ_{ij}, we get the following training rules (7)-(9):

$$b_j^{<r+1>} = b_j^{<r>} - \alpha\frac{\partial E^{<r>}}{\partial b_j} = b_j^{<r>} - 2\alpha e\beta_{rj}, \tag{7}$$

$$c_{ij}^{<r+1>} = c_{ij}^{<r>} - \alpha\frac{\partial E^{<r>}}{\partial c_{ij}} = c_{ij}^{<r>} - \alpha e\beta_{rj}\left(b_j^{<r>} - y_r\right)\frac{x_{ri}-c_i^{<r>}}{\left(\sigma_{ij}^{<r>}\right)^2}, \tag{8}$$

$$\sigma_{ij}^{<r+1>} = \sigma_{ij}^{<r>} - \alpha\frac{\partial E^{<r>}}{\partial \sigma_{ij}} = \sigma_{ij}^{<r>} - \alpha e\beta_{rj}\left(b_j^{<r>} - y_r\right)\frac{x_{ri}-c_i^{<r>}}{\left(\sigma_{ij}^{<r>}\right)^2}, \tag{9}$$

where $i = \overline{1,n}, j = \overline{1,m}, r = \overline{1,M}$.

For each input-output, the training takes place in two stages. At the first stage, fuzzy output is performed for the current input vector [30], [31]. At the second stage, the error is calculated (difference between received and expected output values), and the parameters of the fuzzy knowledge base are modified [32], [33].

ANFIS is an adaptive network based on the Takagi-Sugeno neuro-fuzzy inference system. ANFIS is one of the first variants of hybrid neuro-fuzzy networks: neural network of direct signal propagation of a special type [34], [35]. The architecture of a neuro-fuzzy network is isomorphic to a fuzzy knowledge base. Neuro-fuzzy networks use differentiable implementations of triangular norms (multiplication and probabilistic OR), as well as smooth membership functions [36]–[38]. This makes it possible to use fast learning algorithms based on the error back propagation method to configure neuro-fuzzy networks [18], [34].

A distinctive feature of Takagi-Sugeno's neuro-fuzzy inference is as follows: the rules of the algorithm are fuzzy only in IF part, while THEN part is a functional dependence, i.e. the rules in the knowledge base are a kind of switches from one linear law to another. In this case, the boundaries of the subdomains are blurred, which means that several laws can be fulfilled at once, but with different degrees. The degree of ownership of the input vector $X^* = (x_1^*, \dots, x_n^*)$ to values $d_j = b_{j0} + \sum_{i=\overline{1,n}} b_{ij}x_i$ is calculated as follows (10):

$$\mu_{d_j}(X^*) = \mu_j(x_1^*) \dots \mu_j(x_n^*), j = \overline{1,m}. \tag{10}$$

Then the resulting value of the output *y* is found as a superposition of linear laws executed at a given point X^**n*-th of a dimensional factor space (11):

$$y = \frac{\sum_{j=1}^{m}\mu_{d_j}(X^*)d_j}{\sum_{j=1}^{m}\mu_{d_j}(X^*)}. \tag{11}$$

The described algorithm is a combination of methods of the steepest descent and back propagation of the error, which optimizes linear and nonlinear parameters of the fuzzy knowledge base. This method allows simultaneously configuring the conclusions of the rules and the membership functions of the terms of the input variables [3]. Also, using this method, we can get fairly simple rules for configuring the parameters of rule parcels.

4. RESULTS AND DISCUSSION

The constructed fuzzy system simulates the dependence of the final priority on the priority of the task and the time the task is in the queue, which was the result of a two-stage procedure for designing fuzzy Sugeno system. To form fuzzy rules from the experimental sample, we used a fast one-pass algorithm without iterative optimization procedures, the so-called “subtractive clustering method” [4]. For cluster analysis, a radius from 0 to 1 is specified, which determines how far from the center of the cluster its elements can be. It has been experimentally found that setting the radius to a scalar value of 0.5 gives the best approximation. As a result, a fuzzy Sugeno model with eight rules was obtained. The value of the difference of squares was 0.0055 (Figure 4, top graph).

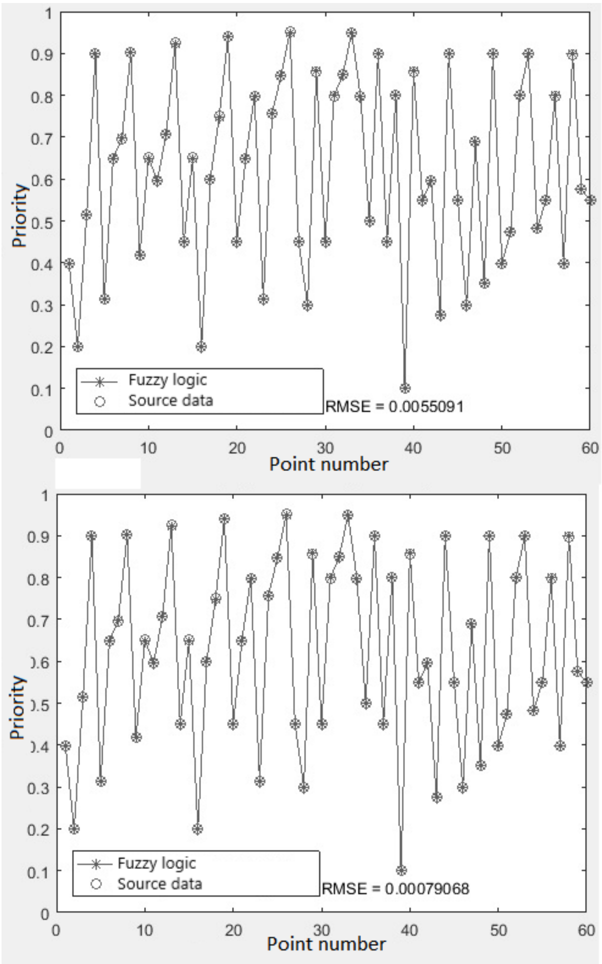

Figure 4. Comparison of the obtained results with experimental data (from above: after structural identification, from below: after parametric identification)

Further tuning of the parameters of the fuzzy model in order to improve the accuracy was carried out using ANFIS training. The value of the square difference at this stage was 7.9*10-4, which is 1.43 times better than at the previous stage. Thus, an increase in the quality of modeling is achieved. A comparison of the experimental input data with the results of the ANFIS system calculations can be seen in the Figure 4 (lower graph), where the experimental data are indicated by dots, and the data obtained by the neuro-fuzzy module are indicated by asterisks. It is clearly seen that these two values for each element of the sample coincide.

As a result, a neural network with two input variables and one output variable was obtained (Figure 5), which has the following structure:

a. The first layer. Each node of the layer is a term with the corresponding membership function μ. The network inputs are connected only to their own terminals. The number of nodes of the first layer is equal to the sum of the powers of the term sets of input variables. As a result of the program, 8 nodes of the first layer were obtained for each input variable: a total number is 16 nodes. The output of the node is fed the degree of belonging of the value of the input variable to the corresponding term;
b. The second layer. Each node of this layer corresponds to one fuzzy rule. The node of the second layer is connected only to the other nodes of the first, which form the antecedents (premises) of the corresponding rule. The output of the node is the degree of rule fulfillment, which is calculated as the product of the input signals;
c. The third layer. On this layer, the degrees of rule execution are normalized: each node calculates the relative degree of execution of a fuzzy rule;
d. The fourth layer. Each node calculates the contribution of one fuzzy rule to the network output;
e. The fifth layer. A single node of this layer summarizes the contributions of all the rules. Aggregation of the result obtained by different rules.

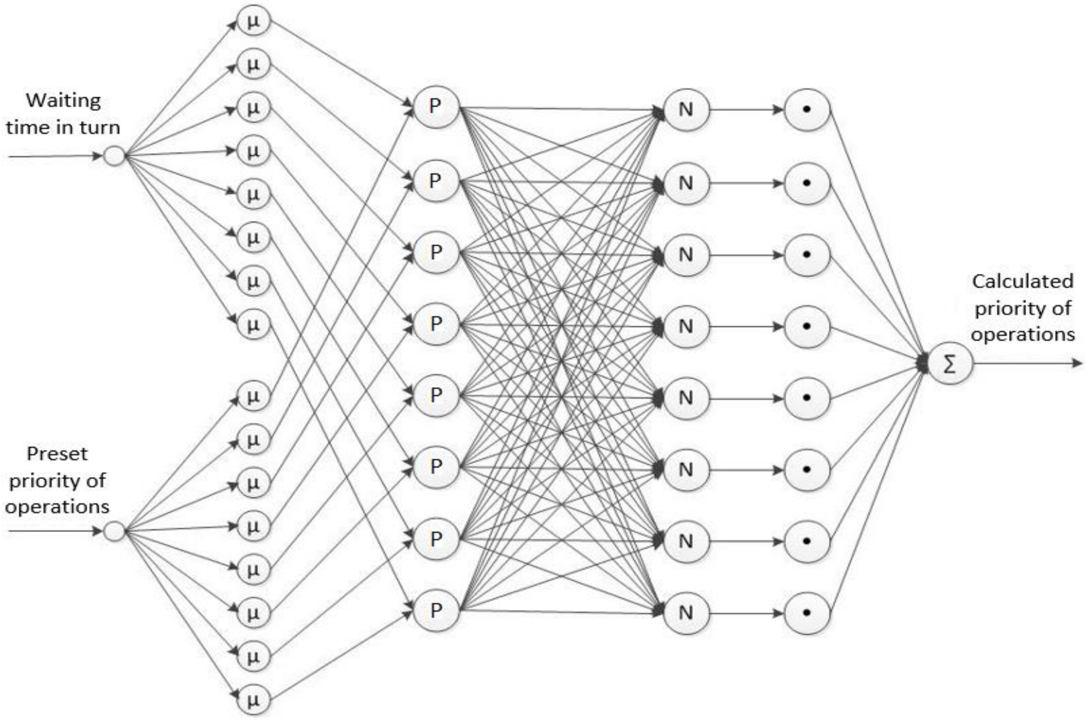

Figure 5. Resulting neural network

After building and training the neural network, ANFIS system builds a decision-making surface, which is shown in the Figure 6. Criteria R^2 for the resulting model, it was 1 with accuracy 10^{-4}. The developed neuro-fuzzy module is the main stage for the construction of a linear programming problem, which is solved by the well-known simplex method under given constraints. The resulting surface (Figure 6) describes the dependence of the new priority on the initial one set by the dispatcher and the time when the task is in the queue. The obtained values serve as known coefficients in the optimization problem of linear programming for determining the optimal locomotive-operation pairs. The conducted experiments on the formation of a plan for the execution of technological operations showed that the developed optimization module allowed saving about 19% of the operating time of the locomotive fleet, reducing it from 13.57 hours to 10.97 for the situation of receiving 5 tasks during the day.

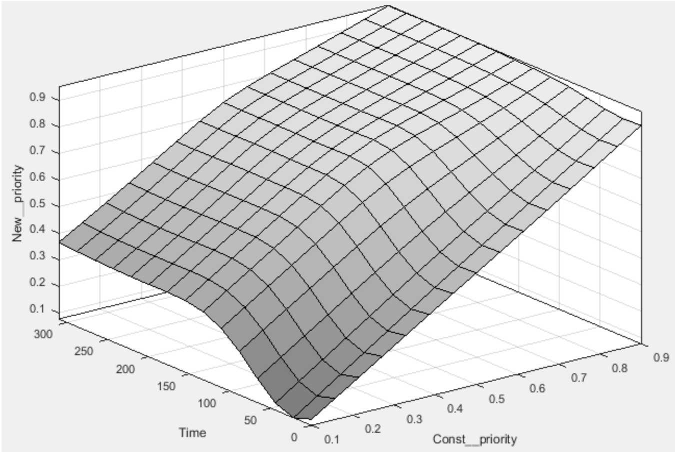

Figure 6. Decision-making surface

5. CONCLUSION

The developed mathematical model uses the technique of neuro-fuzzy computing to identify the parameters for identifying fuzzy systems and calculating the priorities of operations within the decision support system of a logistics service with a complex delivery topology. The simulation results and recommendations for using the developed model are presented. The model is implemented practically in the form of a software system: i) at the first stage, the calculation of the optimal train movement path along the railway station is carried out, which allows reducing the operating time of the locomotive fleet and ii) at the second stage, the software complex begins to function in real time. At this stage of work, the time of operations execution and the optimal sequence of operations are calculated by calculating the dynamic priority of dispatching tasks.

The novelty of the developed mathematical model, in contrast to the known ones, is in the development of scheduling theory methods in terms of problems of minimizing the number of performers, graph theory in terms of the dynamic employment of individual nodes and edges of the graph, as well as planning the movement of several pieces of equipment in the graph simultaneously. Also new is the developed method for calculating dynamic priorities, which, unlike the known ones, is a combination of steepest descent and backpropagation methods, which optimizes linear and nonlinear parameters of the fuzzy knowledge base. This method allows simultaneously adjusting the conclusions of the rules and the membership functions of the terms of the input variables. Studies conducted on real data have shown that the implementation of the method for calculating dynamic priorities in the form of a constructed fuzzy model is adequate, according to the criterion R2; the developed method is integrated with the optimization algorithm and put into operation in the control room of a large metallurgical plant, as a result of which it was possible: i) to minimize the time budget for the use of locomotives at the railway station and ii) to minimize the average daily mileage of cars.

REFERENCES

[1] I. N. Lavrikov, *Transport logistics*. Tambov, Russia: Publishing house “TSTU,” 2016.
[2] V. S. Tanaev and V. V. Shkurba, *Introduction to scheduling theory*. Moscow, USSR: Nauka, 1975.
[3] V. S. Tanaev, Y. N. Sotskov, and V. A. Strusevich, *Scheduling Theory: Multi-Stage Systems*. Springer Netherlands, 1994, doi: 10.1007/978-94-011-1192-8.
[4] X. Xie, D. Yue, and C. Peng, “Relaxed Real-Time Scheduling Stabilization of Discrete-Time Takagi–Sugeno Fuzzy Systems via An Alterable-Weights-Based Ranking Switching Mechanism,” *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 6, pp. 3808–3819, Dec. 2018, doi: 10.1109/tfuzz.2018.2849701.
[5] N. Popescu, D. Popescu, V. Ciobanu, and M. Ivanescu, “A note on the Takagi–Sugeno control algorithm for a class of distributed parameter system,” *Soft Computing*, vol. 23, no. 16, pp. 7207–7214, Jul. 2018, doi: 10.1007/s00500-018-3366-8.
[6] D. I. Batishchev and D. A. Kogan, *Computational complexity of extremal enumeration problems*. Russia: Lobachevsky State University of Nizhni Novgorod, National Research University, 1994.
[7] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, Dec. 1959, doi: 10.1007/bf01386390.
[8] X. Liu, X. Su, P. Shi, and C. Shen, “Observer-Based Sliding Mode Control for Uncertain Fuzzy Systems via Event-Triggered Strategy,” *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 11, pp. 2190–2201, Nov. 2019, doi: 10.1109/tfuzz.2019.2895804.
[9] H. T. Rauf, S. Malik, U. Shoaib, M. N. Irfan, and M. I. Lali, “Adaptive inertia weight Bat algorithm with Sugeno-Function fuzzy search,” *Applied Soft Computing*, vol. 90, p. 106159, May 2020, doi: 10.1016/j.asoc.2020.106159.
[10] A. A. Artamonov, L. M. Bogdanova, S. Y. Nagibin, D. I. Loskutov, and A. S. Chemakin, “Mathematical model of chemical process prediction for industrial safety risk assessment,” *Procedia Computer Science*, vol. 190, pp. 107–114, 2021, doi: 10.1016/j.procs.2021.06.013.

[11] J. M. A. Moral, C. Castiello, L. Magdalena, and C. Mencar, “An Overview of Fuzzy Systems,” in *Explainable Fuzzy Systems*, Springer International Publishing, 2021, pp. 25–47, doi: 10.1007/978-3-030-71098-9_2.

[12] L. Wang and H.-K. Lam, “New Stability Criterion for Continuous-Time Takagi–Sugeno Fuzzy Systems With Time-Varying Delay,” *IEEE Transactions on Cybernetics*, vol. 49, no. 4, pp. 1551–1556, Apr. 2019, doi: 10.1109/tcyb.2018.2801795.

[13] A.-T. Nguyen, T. Taniguchi, L. Eciolaza, V. Campos, R. Palhares, and M. Sugeno, “Fuzzy Control Systems: Past, Present and Future,” *IEEE Computational Intelligence Magazine*, vol. 14, no. 1, pp. 56–68, Feb. 2019, doi: 10.1109/mci.2018.2881644.

[14] C. Zhang, Y. Yu, Y. Wang, and M. Zhou, “Takagi–Sugeno Fuzzy Neural Network Hysteresis Modeling for Magnetic Shape Memory Alloy Actuator Based on Modified Bacteria Foraging Algorithm,” *International Journal of Fuzzy Systems*, vol. 22, no. 4, pp. 1314–1329, Apr. 2020, doi: 10.1007/s40815-020-00826-9.

[15] T. H. Cormen, C. I. Lazerson, R. L. Rivest, and C. Stein, “Algorithms: Construction and Analysis,” in *Max-Midi*, Moscow, Russia: Dialectics Williams, 2016. [Online]. Available: https://www.maxmidi.com/algorithms-construction-and-analysis-thomas-h-cormen-charles-i-lazerson-ronald-l-rivest-clifford-stein/

[16] D. I. Batishchev, *Genetic algorithms for solving extremal problems*. Nizhny Novgorod, Russia: Lobachevsky State University of Nizhni Novgorod - National Research University, 1995.

[17] V. A. Kostenko and A. V. Vinokurov, “Locally Optimal Algorithms for Designing Schedules Based on Hopfield Networks”, *Programming and Computer Software*, vol. 29, no. 4, pp. 199–209, Jul. 2003, doi: 10.1023/A:1024918625291.

[18] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Second edition. Delhi, India: Pearson Education Prentice Hall, 1999.

[19] Z. T. Omer and A. H. Abbas, “Image anomalies detection using transfer learning of ResNet-50 convolutional neural network,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 27, no. 1, p. 198, Jul. 2022, doi: 10.11591/ijeecs.v27.i1.pp198-205.

[20] R.-M. Chen and Y.-M. Huang, “Competitive neural network to solve scheduling problems,” *Neurocomputing*, vol. 37, no. 1–4, pp. 177–196, Apr. 2001, doi: 10.1016/s0925-2312(00)00344-1.

[21] J. J. Hopfield and D. W. Tank, “‘Neural’ computation of decisions in optimization problems,” *Biological Cybernetics*, vol. 52, no. 3, pp. 141–152, Jul. 1985, doi: 10.1007/bf00339943.

[22] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities.,” *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, Apr. 1982, doi: 10.1073/pnas.79.8.2554.

[23] S. Y. Nagibin and D. I. Loskutov, “Analysis of Neural Networks for Predicting Time Series When Assessing Industrial Safety Risks,” *TEM Journal*, pp. 477–483, May 2020, doi: 10.18421/tem92-08.

[24] A. V. Ignatenkov, *Neural network modeling and ordering of traffic flows on railway lines*. Penza, Russia: Penza State University, 2018.

[25] A. V. Nazarov and A. I. Loskutov, *Neural network algorithms for forecasting and optimizing systems*. Saint Petersburg, Russia: Science and Technology, 2003.

[26] L. M. Kadyeva, S. E. Levin, and Ya. N. Okrent, “Kombinirovannyy Algoritm Lineynoy Optimizatsii s Poiskom Maksimal'nogo Potoka na Grafe”, *Molodoy Uchenyy (in Russ.),* vol. 19, no. 123, pp. 8–13, 2016. https://elibrary.ru/item.asp?id=26740290

[27] A. P. Rothstein, *Intelligent identification technologies: fuzzy sets, genetic algorithms, neural networks*. Vinnytsa, Ukraine: UNIVERSUM-Vinnytsa, 1999.

[28] A. P. Rotshtein and D. I. Katel’nikov, “Identification of nonlinear objects by fuzzy knowledge bases”, *Cybernetics and Systems Analysis*, vol. 5, pp. 53–61, Sep. 1998, doi: 10.1007/bf02667040.

[29] T. Takagi and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, pp. 116–132, Jan. 1985, doi: 10.1109/tsmc.1985.6313399.

[30] A. I. Orlov, *Applied statistics*. Moscow, Russia: Exam, 2004.

[31] S. D. Shtovba, *Designing fuzzy systems using MATLAB*. Moscow: Hot line-Telecom, 2007.

[32] Ya. Z. Tsypkin, *Foundations of Information Identity Theory*. Moscow: Science, 1984.

[33] J.-S. R. Jang, “ANFIS: adaptive-network-based fuzzy inference system,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993, doi: 10.1109/21.256541.

[34] D. Nauck, F. Klawonn, and R. Kruse, *Foundations of Neuro-Fuzzy Systems*. New York: John Wiley & Sons, 1997.

[35] V. V. Kruglov, *Fuzzy logic and artificial neural networks*. Moscow, Russia: Fizmatlit, 2001.

[36] Y. Wang, Y. Xia, C. K. Ahn, and Y. Zhu, “Exponential Stabilization of Takagi–Sugeno Fuzzy Systems With Aperiodic Sampling: An Aperiodic Adaptive Event-Triggered Method,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 2, pp. 444–454, Feb. 2019, doi: 10.1109/tsmc.2018.2834967.

[37] H.-B. Zeng, K. L. Teo, Y. He, and W. Wang, “Sampled-data-based dissipative control of T-S fuzzy systems,” *Applied Mathematical Modelling*, vol. 65, pp. 415–427, Jan. 2019, doi: 10.1016/j.apm.2018.08.012.

[38] S. Dong, M. Fang, P. Shi, Z.-G. Wu, and D. Zhang, “Dissipativity-Based Control for Fuzzy Systems With Asynchronous Modes and Intermittent Measurements,” *IEEE Transactions on Cybernetics*, vol. 50, no. 6, pp. 2389–2399, Jun. 2020, doi: 10.1109/tcyb.2018.2887060.

BIOGRAPHIES OF AUTHORS

Leyla M. Bogdanova graduated from the Master's degree program of National Research Nuclear University MEPhI. Currently, she is a researcher in Plekhanov Russian University of Economics. She has 5 articles, 4 of which are indexed in Scopus. She can be contacted at email: bogdanova_leyla_rea@mail.ru.

Sergey Ya Nagibin is a Doctor of Technical Sciences, Professor, Head of Department "Control systems, informatics and electric power industry", Moscow Aviation Institute (National Research University). He is a laureate of the Prize of the Government of the Russian Federation in the field of science and technology, academician of Russian Academy of Cosmonautics named after K. E. Tsiolkovsky, Corresponding Member of the International Academy of Informatization, Honored Tester of Space Technology, Honorary Radio Operator of Russia, author of more than 250 scientific papers (including 9 monographs and 10 inventions). He was awarded state awards (among them there is Order of Military Merit and several medals, as well as departmental medals and certificates of Bank of Russia, Federation of Cosmonautics and the Academy of Cosmonautics). He can be contacted at email: nagibinmai@mail.ru.

Dmitry I. Loskutov he is a postgraduate student in Moscow Aviation Institute (National Research University). He is the author of more than 10 scientific papers in the field of neural networks, including indexed in Scopus and WoS. He can be contacted at email: loskutov.d.i@mail.ru.

Natalia A. Goncharova is a researcher of Moscow Aviation Institute (National Research University). She has 24 publications, including in journals indexed in Scopus. She can be contacted at email: natalia.a.goncharova@mail.ru.