Simulation Modelling Practice and Theory 33 (2013) 144–157

Contents lists available at SciVerse ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

A fuzzy Petri net model to estimate train delays

Sanjin Milinković *, Milan Marković [1], Slavko Vesković [1], Miloš Ivić [1], Norbert Pavlović [1]

University of Belgrade, Faculty of Transport and Traffic Engineering, 11000 Belgrade, Vojvode Stepe 305, Serbia

ARTICLE INFO

Article history:
Available online 16 January 2013

Keywords:
Fuzzy Petri nets
Train delays
Railway simulation
ANFIS

ABSTRACT

Even with the most accurate timetable, trains often operate with delays. The running and waiting times for trains can increase unexpectedly, creating primary delays that cause knock-on delays and delays for other trains. The accurate estimation of train delays is important for creating timetables, dispatching trains, and planning infrastructures. In this work, we proposed a fuzzy Petri net (FPN) model for estimating train delays. The FPN model with characteristics of hierarchy, colour, time, and fuzzy reasoning was used to simulate traffic processes and train movements in a railway system. The trains were coloured *tokens*, the track sections were termed *places*, and discrete events of train movement were termed *transitions*. The train primary delays were simulated by a fuzzy Petri net module in the model. The fuzzy logic system was incorporated in the FPN module in two ways. First, when there were no historical data on train delays, expert knowledge was used to define fuzzy sets and rules, transforming the expertise into a model to calculate train delays. Second, a model based on the Adaptive Network Fuzzy Inference System (ANFIS) was used for systems where the historical data on train delays were available (from detection systems or from the train dispatcher's logs). The delay data were used to train the neuro-fuzzy ANFIS model. After the results of the fuzzy logic system were verified, the ANFIS model was replicated by a fuzzy Petri net. The simulation was validated by animating the train movement and plotting the time-distance graph of the trains. Results of the simulation were exported to a database for additional data mining and comparative analysis. The FPN model was tested on a part of the Belgrade railway node.

1. Introduction

Train delays are one of the most commonly used parameters for determining timetables and solving infrastructure problems. The delay data are also important for train dispatching and railway traffic operations. Day-to-day or even hour-to-hour train delays can occur within the same day, and the unpredictability of the delays makes efficient planning of railway operations very difficult even for a short period of time.

Primary delays are train delays caused by external stochastic disturbances. When primary delays develop inside the observed network, they are called original delays. If the buffer times between trains are less than the length of the primary disturbance, delay is propagated to other trains. The primary delay of one train can cause delays on other trains and create knock-on or secondary delays. It is very difficult to calculate and predict secondary delays because they depend on the length of primary delays, the timetable of the trains, and the infrastructure (such as single or double track, station layouts, and interlocking) [10]. Primary delays are often caused by technical failures, lower-than-scheduled running speeds, prolonged

* Corresponding author. Tel.: +381 (11) 3091306; fax: +381 (11) 3096704.
E-mail addresses: s.milinkovic@sf.bg.ac.rs (S. Milinković), milan@sf.bg.ac.rs (M. Marković), veskos@sf.bg.ac.rs (S. Vesković), m.ivic@sf.bg.ac.rs (M. Ivić), norbert@sf.bg.ac.rs (N. Pavlović).
[1] Tel.: +381 (11) 3091306; fax: +381 (11) 3096704.

1569-190X/$ - see front matter
http://dx.doi.org/10.1016/j.simpat.2012.12.005

alighting and boarding times of passengers, and bad weather conditions [8]. Distribution of primary delays can be obtained by a statistical analysis of existing empirical data. Yuan and Hansen [24] proposed an analytical stochastic model of propagation delays in the stations to calculate secondary delays.

Three common approaches to determine train delays are typically used in the literature [14]: analytical methods, micro-simulation methods and statistical analyses based on empirical data.

The analytical model uses the queuing theory; it does not require a large amount of input data and usually applies simplifying assumptions on the system. Queuing models estimate the total and average waiting time of trains at platform tracks or junctions. These models are applied during strategic planning to evaluate the impact of increasing train frequencies and the impact of modifying infrastructure and train characteristics on the waiting time. Simulation models are detailed representation of a railway system, where different trains interact with each other and with the infrastructure. They require data regarding the infrastructure, the timetable and the performance of the trains [9]. If one of these pieces of data is unknown, assumptions would be required; therefore, the simulation results would depend on the quality of the input data. Microscopic simulation tools can be used to model the propagation of train delays in large railway networks, but their use require extensive work to model the infrastructure topology, signalling and timetables using simulation software such as RailSys and OpenTrack [3,9].

Statistical analysis is primarily used for modelling the occurrence of primary delays. The observed delays can be used to establish empirical relationships between how well the capacity is utilised and the secondary delays, given the prevailing level of primary delays. This can be applied in railway systems that are well regulated and operate in stable conditions. In systems where there are many possible sources of disruptions and a relatively high probability of external influences that could induce primary delays, it is difficult to find a relationship that can be used to calculate train delays.

In the railway system of Serbian Railways, statistical analysis of arrival delay data suggests that many factors influence train delays; additionally, large disturbances are present in the system. Train delays are very difficult to predict because of the many factors involved.

Fuzzy logic is a mathematical tool used to model traffic processes that are distinguished by subjectivity, uncertainty, ambiguity and imprecision [22,23]. Many authors use predictive modelling systems with fuzzy logic. Fay [7] used fuzzy logic to model a dispatcher support system for railway operation and control. This model was defined as a fuzzy Petri net model that combined expert knowledge of fuzzy systems and the graphical power of Petri nets, making the model easy to design, test, improve and maintain. Cheng and Yang [5] proposed a fuzzy Petri net model that used the professional knowledge of dispatchers to create database rules for testing a system in case of disorder.

In this paper we present two approaches to modelling the primary delays of arriving trains in Serbian Railways network. The first model was created for systems without any historical data on train delays that require additional information to determine the occurring frequency and length of train delays. For most technical systems, the solution to a problem is determined using the experience of experts who have dealt with these problems for an extended period of time. The fuzzy logic model used to calculate train delays in this work considered the expertise, experience and knowledge of railway personnel who directly participated in regulating the traffic in the system. Data from personnel interviews and timetable information were used to define the parameters of a fuzzy system in a fuzzy Petri net for forecasting trains delays. In this manner, the parameters of the fuzzy logic system were different between specific cases. Knock-on delays were calculated by the model using simulation results. A second model based on the Adaptive Network Fuzzy Inference System (ANFIS) [12] was used for systems where the historical data on train delays were available (from detection systems or from the train dispatcher's logs). The delay data were used to train the neuro-fuzzy ANFIS model. After the results were verified, the ANFIS model was replicated by a fuzzy Petri net (FPN).

Simulation of train movement in railway systems is very complex because of the many parameters and relations required to describe such a complex system. Complex system can be divided into specific interrelated modules (subsystems). These modules can be easily modelled. In modelling complex systems, we can use abstraction, polymorphism, hierarchy, and monitoring [2]. The principle of monitoring provides the information about the model behaviour and observation data of the changes in parameter values of model's objects, and evaluation of the current state and markings in the model. The simulation tool must be able to construct a model that incorporates all interlocking principles, operating rules and data. Petri nets are used for graphical and mathematical modelling of various systems. High level Petri nets (HLPNs), which consider the time, colour, stochastic and hierarchical characteristics, are used to model complex system, and they can present the model graphically. Many simulation models found in the literature presented analyses of various railway systems with focus on train delays. Basten et al. [4] created a simulation model for the analysis of interlocking specification using collared Petri nets in the software Expect [11]. Aalst and Odijk [1] proposed the interval-timed coloured Petri nets to model and analyse railway stations; the train delays were specified by an upper and lower bound, i.e., an interval. Daamen et al. [6] developed coloured Petri nets to identify route conflicts and estimate knock-on delays.

In our previous work [15], we have shown that a fuzzy logic model that uses expert knowledge can be used to calculate primary train delay. In further research, as presented in this paper, we use the data detected from the real system to train and test the new neuro-fuzzy model for estimating the train delays and to establish the connection between the train parameters and the corresponding delay. We tested the Petri net (PN) model on a part of the Belgrade railway node of Serbian Railways. The FPN module in the PN model generated the primary delays that were combined with timetable data to produce the arrival times of the trains. Analysis of the traffic conditions and data collected during the research suggested that, for the FPN module, the following parameters should be used as inputs for fuzzy logic train delay models: the train category, the time of arrival at the station, the distance travelled, and the infrastructure influence. The time of arrival, the train data and the infrastructure data were inputs to the Petri net simulation model. Results of the simulation model were verified by animating track sections occupied by trains and using a train time-distance graph. The FPN model presented in this paper was used as a new method in Serbian Railways research studies for estimating train delays.

This paper is structured as follows. Section 2 starts with a definition of fuzzy Petri nets. The FPN model to simulate train delays, and basic rules for train movements are described in Section 3. Section 4 presents the hierarchy and modules of the FPN. The steps for constructing a model of the railway system are provided in Section 5. Section 6 demonstrates the application and the results of the FPN model of the part of Belgrade railway node. Conclusions are provided in the final section.

2. Fuzzy Petri nets

A fuzzy Petri net is a Petri net that uses fuzzy logic rather than Boolean logic [13]. The fuzziness concept can be incorporated in Petri nets by applying a fuzzy reasoning mechanism over the Petri nets structure.

2.1. Petri nets

Petri nets are mathematical modelling tools used to analyse and simulate concurrent systems [17]. The theory of Petri nets (PNs) is based on a mathematical theory of bipartite graphs. A Petri net is one of several mathematical descriptions of discrete distributed systems. The distributed system is modelled as a bipartite directed graph with two sets of nodes: the set of places that represent state or system objects and the set of events or transitions that determine the dynamics of the system.

A Petri net is a 5-tuple $PN = (P, T, I, O, M)$, where:

- $P = \{p_1, p_2, \ldots, p_m\}$ is a finite set of *Places*.
- $T = \{t_1, t_2, \ldots, t_n\}$ is a finite set of *Transitions*.
- I: an input function, $(P \times T) \rightarrow N$, where N is the set of non-negative integer numbers. The value $I\ (p,t)$ is the number of (directed) *arcs* from the place p to the transition t.
- O: an output function, $(T \times P) \rightarrow N$, where the value $O\ (t,p)$ is the number of arcs from the transition t to the place p.
- M: $P \rightarrow \{0, 1, 2, 3, \ldots\}$ is the initial marking assigning to place p a non-negative integer k, i.e., marking place p with k tokens.

Places are represented with circles, and transitions are represented with rectangles on PN graphs. The transition of a system from one state to another occurs when an event transpires; an event can also be the moment when the time period in a certain state expires. The arcs (input and output functions) are represented with directed lines.

So that performance analyses are feasible and PN models are more transparent and understandable, additional values, parameters, and variables are used within the Petri nets. Such extensions are denoted as high-level Petri nets (HLPNs).

A HLPN is a structure HLPN = $(P, T, D; Type, Pre, Post, M_0)$ where:

- D is a non-empty finite set of non-empty domains where each element of D is called a type.
- *Type*: $P \cup T \rightarrow D$ is a function used to assign types to places and to determine transition modes.
- *Pre*; *Post*: $TRANS \rightarrow \mu PLACE$ are the pre and post mappings with
  $TRANS = \{(t,m) | t \in T, m \in Type(t)\}$.
  $PLACE = \{(p,g) | p \in P, g \in Type(p)\}$.
- $M_0 \in \mu PLACE$ is a multiset known as the initial marking of the net ($\mu PLACE$ is the set of multisets over the set, *PLACE*).

In theory, PNs and HLPNs have the same computational power, but HLPNs have much more modelling power because they have better structuring facilities. With HLPNs, it is possible to use data types and complex data manipulation because each token is assigned its own colour as a certain value of a certain complex data type. Logic expressions and functions can be built using the token colours and can be associated with places, transitions and arcs. Additionally, with HLPNs, it is possible to make hierarchical descriptions, where a large model can be obtained by combining a set of submodels.

The characteristics that render an HLPN a sufficient tool for developing, describing and analysing complex railway systems include the following:

- An HLPN can effectively analyse concurrent systems by verifying the safety rules and standards for train operations (such as operating rules and signalling rules) [9,19] and timetable analysis.
- An HLPN uses a graphical presentation that is easy to understand, even for those that are not familiar with HLPNs.
- The HLPN model is easy to modify because of its modularity.
- The initial marking of HLPNs allows for different train operating scenarios to be experimented with.

2.2. Fuzzy logic in fuzzy Petri nets

Fuzzy Petri nets can be used to model systems under incomplete information about state and time. As mentioned in Section 2.1, PNs are constructed using several types of objects: transitions, places, tokens and arcs. All of these objects may be fuzzified (fuzzy token, fuzzy place, fuzzy transition and fuzzy arc) [25]. A fuzzy token is a generalisation of the token that assigns a truth value for belonging to a place. The token has a linguistic value that is defined as a membership function for a linguistic variable. This function also determines the degree of membership in a particular place or the truth value of that proposition.

Modelling fuzzy systems with classical PNs or HLPNs presumes that elements are redefined such that fuzzy information can be presented; in addition, structural and functional elements are defined for specific features of the fuzzy system [13,20].

The most commonly used fuzzy inference techniques are the Mamdani and Sugeno methods [18]. The Mamdani-type fuzzy inference system (FIS) assumes that the output of the process is a fuzzy set. The set of IF-THEN rules, which forms the linguistic description in a fuzzy control system, is:

- *Rule* 1 (c_1): IF I_1 is A_{11} AND...AND I_n is A_{1n} THEN O is B_1.
- *Rule i* (c_i): IF I_1 is A_{i1} AND...AND I_n is A_{in} THEN O is B_i.
- *Rule m* (c_m): IF I_1 is A_{m1} AND...AND I_n is A_{mn} THEN O is B_m.

where I_i are input variables, O is the output variable, A_{ij} and B_i are certain predicates characterising the input and output variables, *Rule i* is a rule of linguistic description, and c_i denotes the confidence degree of applying *Rule i*.

These rules can be graphically represented using an FPN with *places, transitions* and *arcs* (Fig. 1), where each transition corresponds to one rule of the linguistic description.

The fuzzy inference process consists of five steps: fuzzification of the input variables; application of the fuzzy operator (AND or OR) in the antecedent; implication from the antecedent to the consequent; aggregation of the consequents across the rules; and defuzzification [21].

The fuzzification module enables the transition of each input variable and maps the crisp input signal to a corresponding fuzzy value. The input is fuzzified by evaluating a function. The input value x_0 is then evaluated using the input variables defined with fuzzy set A, and $\mu_A(x)$ is calculated. Each input is fuzzified over all of the qualifying membership functions required by the rules. After the inputs are fuzzified, the antecedent is examined. If the antecedent of a given rule has more than one part, the fuzzy operator is applied to obtain one number that represents the result of the antecedent for that rule. This number is then applied to the output function. The fuzzy AND operator selects the minimum of the input values. The weight associated with every rule is applied to the number given by the antecedent. Then, the implication method is used to produce a fuzzy set represented by a membership function. The consequent is reshaped using a function associated with the antecedent (a single number). Then, the implication method is applied for each rule as MIN (minimum). On the other hand, the aggregation function combines the results from all of the rules into a single fuzzy set with the aggregation method of MAX (maximum). In the final defuzzification step, the aggregated fuzzy output set is converted into a crisp, single number. The Mamdani-type FIS can be easily modelled using Petri nets. Furthermore, Petri net graphs can be created according to specific properties of the fuzzy system, such as the number of inputs, number of outputs, number of rules, and types of operators. The created fuzzy Petri net model could be easily modified using the values of the new fuzzy sets and other properties of FIS. The Mamdani method is widely used because it is intuitive and suitable for human input.

The Sugeno method, which is another commonly used fuzzy inference technique, uses a single spike (a singleton) as the membership function of the rule consequent. The similarity between the Mamdani and Sugeno methods is in the approach to

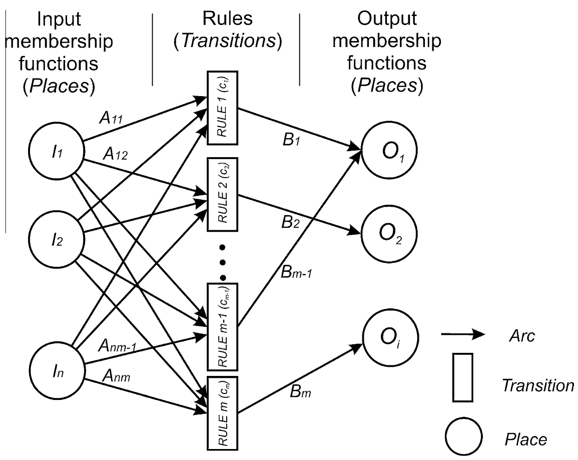

Fig. 1. Graphical presentation of the rule base in a fuzzy Petri net.

fuzzifying inputs and implementing the operators. The main difference is in the type of output membership functions, which could only be linear function or a constant in the Sugeno method.

Because the Sugeno system is more compact and computationally efficient than the Mamdani system, the system enables the construction of fuzzy models using adaptive techniques. These adaptive techniques can be used to customise the membership functions so that the fuzzy system can best model the data [20,21]. In comparison to the Mamdani method, the Sugeno method works better with optimisation and adaptive techniques; it is guaranteed to have continuity of the output surface and is well suited for mathematical analysis.

The Sugeno method is used in Adaptive Network Fuzzy Inference System (ANFIS). An adaptive network is a network of nodes and directional links. The ANFIS hybrid model combines a neural network and fuzzy logic [12]. It uses the abilities of neural networks to learn from the data and the capabilities of fuzzy logic systems to interpret the imprecise data. Using a given input/output data set, the ANFIS system forms a fuzzy inference system where the parameters are adjusted by using the backpropagation algorithm. This approach allows the system to learn from the data it models. During the training process of the ANFIS, if-then rules are determined and membership functions are defined. As the ANFIS model is trained, the model creates and fine-tunes the relationships between inputs and outputs. Once trained, the ANFIS model can be used to calculate the output values for given inputs.

3. The fuzzy Petri net model for simulating train delays

The simulation model of the railway system was created using the ExSpect version 6.41 software [11], in which the HLPN extends the PN to *data*, *time* and *hierarchy* [17]. Tokens have *types* and *values* (colours). In addition, each place in the net also has a type, restricting the type of tokens allowed in that place. ExSpect specification provides a special kind of place, a *store*. A store always contains exactly one token, and it might be considered a variable. Another extension of ExSpect is the *processor*, defined as a transition whose exact behaviour is specified in a functional programming language. A processor has functions to test and modify the value of tokens. In ExSpect, extension of time is defined in tokens, where every token is assigned a time stamp indicating the time that the token becomes available for consumption. The property of hierarchy is defined in elements called *systems.* A system is specified in terms of lower-level systems, places, processors, and an interface to the environment using different types of *pins*.

3.1. Modelling trains movements using fuzzy Petri nets

Modelling the propagation of train delays focuses on specific track layouts, the signalling and train protection system and timetable design. Based on primary delays at the system boundaries, knock-on delays can be estimated using an FPN simulation model. The model of a part of railway network can simulate the effects of primary disturbances on train movements. The propagation of primary delays in such models causes route conflicts and extended waiting times for connections. The overall performance of the system and the secondary delay analysis can be obtained from an FPN simulation model. We proposed an FPN simulation model where primary delays were calculated in the FPN subsystem (module) as inputs to the simulation model and secondary delays were delays caused inside the model due to the propagation of primary delays. Primary or arrival delays were generated by the fuzzy logic module defined for each entry of the model.

In an FPN model, *places* represent track sections, *transitions* represent conditions for train movement, and *tokens* represent trains. The hierarchy of the model defines an insulated track section, which can be a block section, a switch section or a station track section, as a subsystem or a module. However, more detailed descriptions are required regarding the position of a section relative to the signals, the stations and the junction for the track section to be properly defined. A module was defined for each distinctive section. The model was then created by positioning and connecting modules according to the railway line section plan. The simulation model must incorporate complex dispatching and interlocking principles in addition to rules of train movement. Despite the additional time required for the initial programming in this approach, using the simulation model allows predefined modules to be used for modelling systems with similar processes (such as modelling traffic processes in the station or on a railway line).

Train movement and spacing is determined by Automatic Block System rules and procedures [9,19]. There are three train categories in the model: freight, regional and passenger. The three train categories were defined to model the train movement by sections according to their characteristics. Section occupancy time depends on the train length, train acceleration and deceleration as well as on its maximum speed. A section occupation time is calculated based on equations of train movement and maximum speed for the section (and speed limits). The section occupation time depends the section length and train speed $V = min\ (max\ V_{tr};\ max\ V_{sec})$, where V_{tr} is the train maximum speed and V_{sec} is maximum section speed. The station sections have additional dwell time in the stations. The physical occupancy time of the section is the time from the moment when the first shaft enters the section to the moment when the last shaft leaves the same section. Additional train delays can occur when the previous train occupies the next block section. The total occupancy time is the period in which the section is occupied with trains in motion, and it accounts for the time in which the section is reserved for train routes in addition to the physical occupancy. Model parameters are defined for the section and for the token/train. Train parameters are defined in coloured tokens that carry information on the time of departure, train category, occupation time from the last section, and time of entering and departing from the last section. Time data change dynamically in the token as the train moves between sections.

The principles of traffic organisation and train movement used in the model include the following:

- trains can be dispatched from the station if the exit signal allows train movement and the protective path signals are set to forbid movement,
- the train can occupy the next section if next two block sections are free,
- when entering the station, the condition needed to create the route is that all sections on the route must be free.

4. Fuzzy Petri nets modules and the model hierarchy

The basic subsystem or module used for model creation is the track section module. There are similar modules that use the same concepts to model interlocking principles, safety and signalling systems. However, these modules differ in the number of connecting input and output pins used (for modules of the switch) and in the additional rules for train movement within a station area. Modules for generating trains use the timetable data imported from an external database used to generate tokens (trains). In these timetable data, each token is loaded with information regarding the train it represents (such as the train number, category, time of entering into the system and the train route). Tokens leave the module when the simulated clock time matches the time of train departure.

4.1. The block section on the open track module

The module of a block section (Fig. 2) was used to represent a block section on an open track. The module contained *places*, *transitions, storages, and pins (for connecting with other modules).* Storage is a special kind of place for storing parameters. Transitions in the module were enabled for entering and leaving the section depending on the data in the storages. The storages contained information regarding the state of the adjacent sections that were connected, the state of the signals and the state of the simulation clock. When the transition *TrainIN* enabled firing, tokens were placed in *sectionbusy.* Instantaneously, the information regarding the occupancy of the section was sent to the previous two sections and to the signal. The token remained in place until the conditions defined in the transition *TrainOUT* were met. The conditions that needed to be met simultaneously were as follows: the advancement of the simulation clock for the amount of time required for the train to cross the section, the section occupancy time (the travelling time of the train on the section), an unoccupied subsequent section and a signal that allowed further movement.

Transition was allowed when the conditions were met. The signal was set to allow train movement to the next section. Then, the token left the section module. Additional tokens were fired to the place *sectionfree,* and the information regarding leaving the section was sent to the connected modules. The purpose of storage in the modules was to maintain data regarding the section state. The data were used in the transition processor to impose logical conditions and to calculate the time of journey of the train on the section. The other type of storage (*sectionstate*) served to gather the data generated when transitions were fired. These storages contained information regarding the state of the signals and the section occupancy.

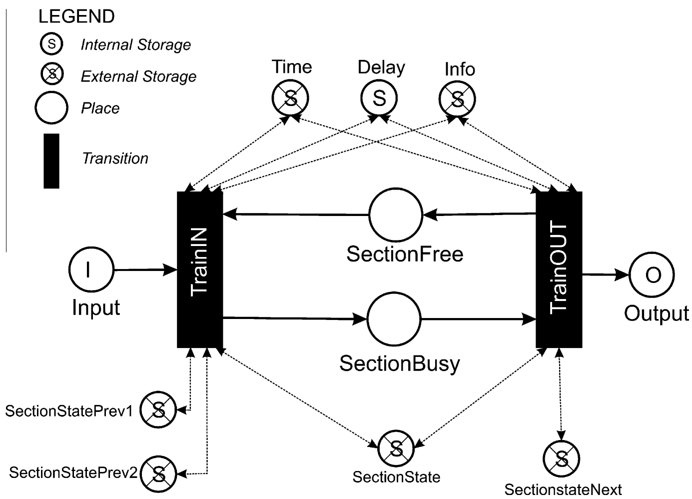

Fig. 2. Module of the block section.

4.2. The station track section module

The processors (predefined transitions) that are used to model tokens entering and leaving the module were connected to a large number of storage nodes. Some of the nodes stored data that managed logical conditions and calculated the section occupation time (also called the train travel time on section). Two processors controlled the process of entering into a section, and another two controlled the process of leaving for the next section. Four processors were used because of the various possible routes in the station (Fig. 3).

4.3. The module to generate trains

The timetable data were imported from an external database used to generate tokens (trains). Additionally, each token contained information regarding the train it represented. The information included the train number, the category, the time of entering into the system and the train route. Defined tokens left the module when the time on the simulation clock matched the time of train departure.

4.4. The module to calculate the primary delay of trains

The module used to calculate primary train delays based on fuzzy logic was the first subsystem that the token entered in the model. After the delay was calculated, the train entered the first section (a section of a station track) in the model when the simulation clock time equalled the time of train departure plus the delay time. The structure, relations, rules and weights of the FPN module were defined by experts' knowledge or historical data for each specific station (Fig. 4). The FPN module was created according to the fuzzy Petri net defined in Section 2.2 and (Fig. 1). The module could be used to represent either the Mamdani fuzzy systems or the ANFIS systems for train delays. This module was developed to calculate train delays on a section of a double-track network between three stations. Depending on the type of system that was modelled, different combinations of input parameters could be applied.

4.4.1. The module to calculate the delay of trains based on the mamdani fuzzy system

Expert knowledge was used to setup the fuzzy logic system when there were no data on train delays in the system. Parameters of a fuzzy logic system were defined in collaboration with traffic dispatchers, operators and experts familiar with the functioning of the system. Their knowledge and experience were used to define input variables, the rules base and output variables. The Delphi method was used to gather the data on the causes of delays and average delays. The experts answered questionnaires, and their statistically processed results were used to create a fuzzy logic system to predict train delays.

We tested various defuzzification methods (including methods based on the centroid or centre of gravity, the middle of maximum and the bisector) in previous research and found that results from these defuzzification methods deviated from the real system data. The centre of gravity (COG) method produced crisp and precise numbers for relatively long train delays, but deviations were observed for smaller delays. In addition, the COG method did not produce values suitable for the lowest output value (a very small delay); furthermore, this method could not be used to produce zero train delays. Therefore, we

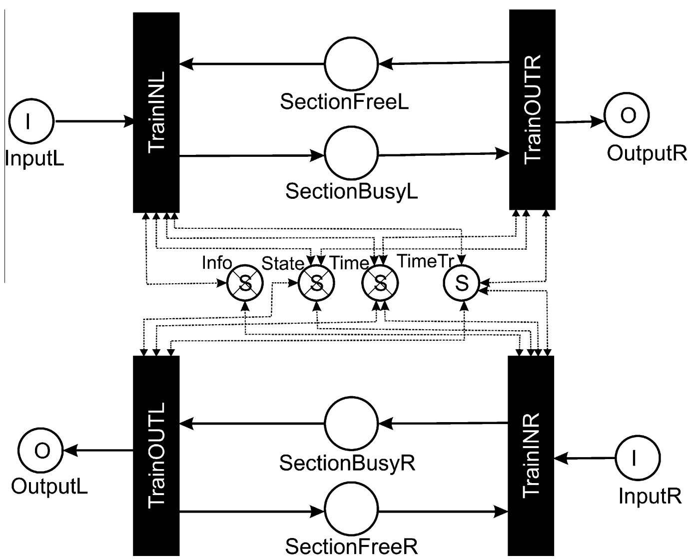

Fig. 3. Station track section module.

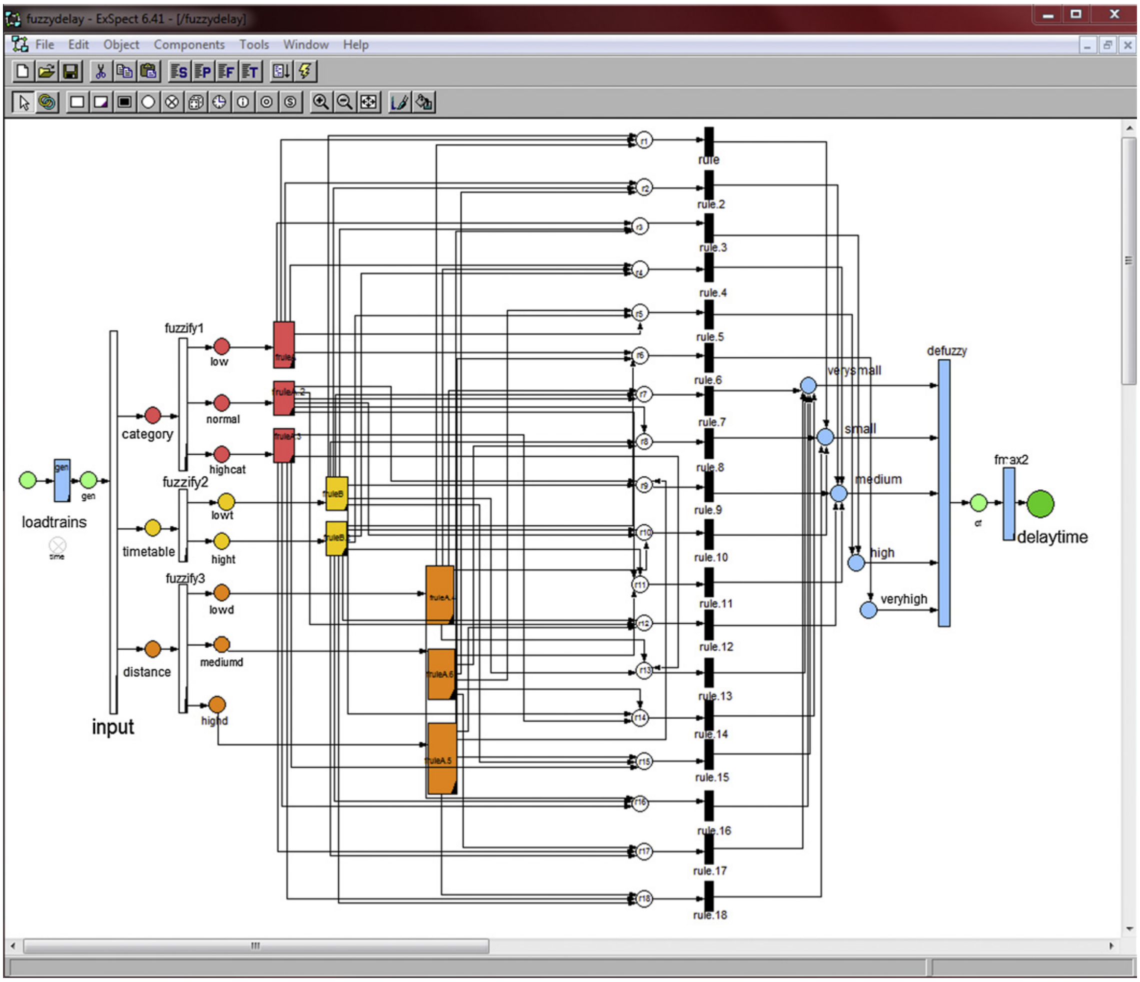

Fig. 4. Example of the FPN module for train delay calculation with three input parameters.

proposed a modified COG method where marginal values for small delays were calculated by a linear membership function of the leftmost fuzzy set. For example, the crisp value was the right limit of the membership function when the output fuzzy set B1 had the probability $\mu B(x) = 1$, and the crisp value was zero when $\mu B(x) = 0$.

Based on the results obtained by the experts' survey, the model was defined by three input parameters: the train category, the timetable influence and the distance travelled by the train. The train category was highly dependent on the probability of train delays. The influence of infrastructure parameters and timetable characteristics in this model were included in the parameter timetable influence. The timetable influence also included the influence of operation time, the types of locomotives, local conditions, technological solutions, principles for safety and signalling and weather conditions. Long travelling distances increased the probability of train delays.

An example of a fuzzy logic membership function with three input variables (the train category, the timetable influence on a train delay and the distance travelled by the train) for a characteristic station in the Belgrade railway node is shown in Fig. 5.

The membership function for the output variable train delay was defined with five fuzzy sets: very small (μ_{VS}), small (μ_S), medium (μ_M), high (μ_H) and very high delay (μ_{VH}) (Fig. 5). A more detailed definition of the input and output variables membership functions can be found in Milinković et al. [16].

The fuzzy logic system was governed by 18 rules, and the weight of every rule was one. The logical AND operator was used as MIN (the rule of minimum for AND relationships). In creating the consequent fuzzy set, the MAX–MIN inference was used. Defuzzification of the output fuzzy variable was performed by modifying the COG method to emphasise the results of defuzzified marginal aggregated output.

An example calculation of train delays using the fuzzy inference system is presented in Fig. 6. The test case used the following input parameters:

Table 1
Comparison of actual train delays with ANFIS results modelled for the Rakovica station.

	Actual delay (min)	Delay by ANFIS model (min)
Mean delay	43.04	44.0
Standard deviation of delay	47.44	41.85

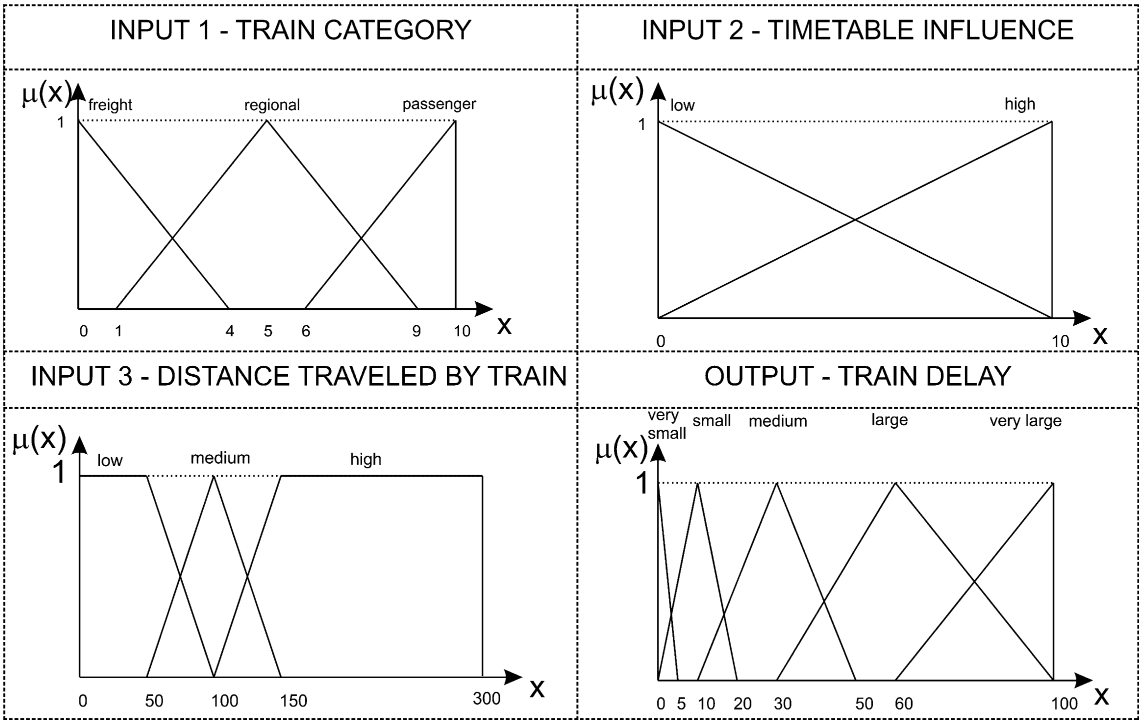

Fig. 5. Membership functions of the input and output parameters.

- train category – 5 (regional train),
- timetable influence – score 4,
- the distance travelled by the train – 40 km.

Rules 7 and 20 were imposed on this set of input data. The calculated train delay using defuzzified output variable was 5.2 min.

4.4.2. The module to calculate the delay of trains based on the adaptive network fuzzy inference system

The previous example addressed the problem where train delays were calculated by the fuzzy logic model defined by expert knowledge of the occurrence of train delays and the performance of the trains in specific systems. For real systems where train delays were monitored by train detection systems or by train dispatcher logs, the train delay data could be used to create a model. A neuro-fuzzy ANFIS model uses the train delay data to determine the membership functions and rules for the FPN module for train delays. We tested various combinations of input parameter sets against output values of train delays for selected stations in a part of the Belgrade railway node. We collected the data on train delays for 31 days in July 2010 for several stations, and data were collected for 3710 trains in all categories: international passenger, domestic passenger, suburban and regional passenger, international freight, direct freight, pick-up freight and other trains. In 31 days, 826 freight trains and 427 other trains were not registered in the predefined timetable. Passenger trains operated according to the predefined timetable, and freight trains operated based on the demand for transport. Therefore, the FPN module for train delays could only be used for trains that operated according to the timetable. Moments of freight trains arrival were generated by a random generator with a statistical distribution predetermined by the Kolmogorov–Smirnov goodness-of-fit test. For the delay frequency of the passenger trains, we constructed statistical tests for each station in the system. After analysing the traffic conditions and the data collected during the research, the best ANFIS model had input data that were defined by the following parameters: the train category, the arrival time at station, the travelled distance and the infrastructure influence.

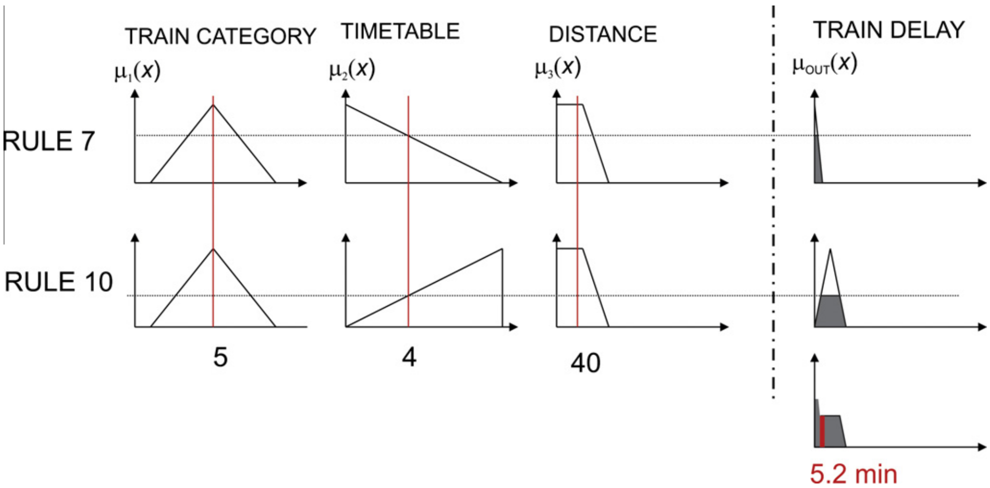

Fig. 6. An example of the fuzzy inference system.

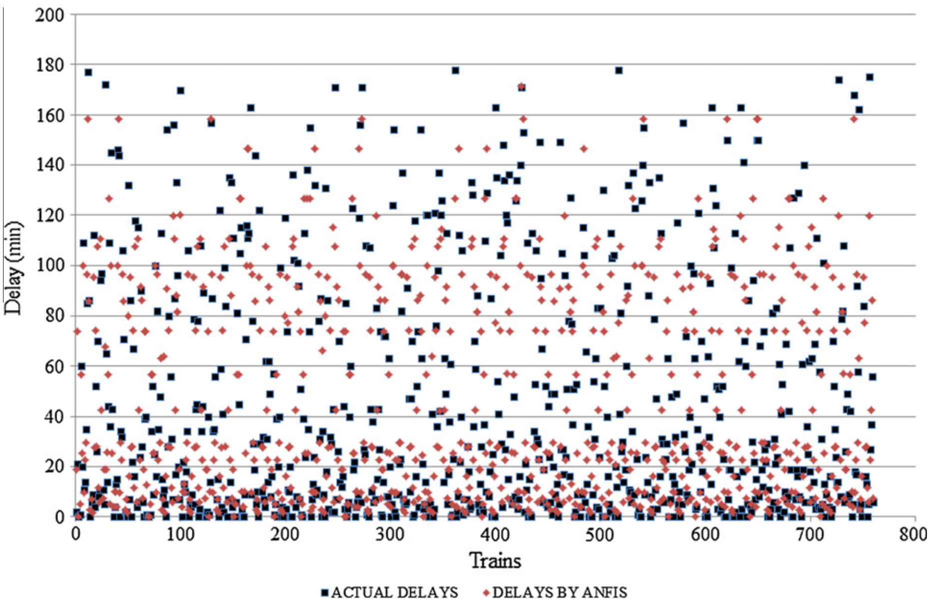

Fig. 7. Actual train delays compared to delays generated by the ANFIS model for 762 trains in the Rakovica station.

The ANFIS model could learn from the given data set. Data sets collected by train delay analysis were transformed into pairs of input–output data. The inputs were the data collected and normalised for each train. These data include the train category determined by its number, the arrival time at the system, the distance travelled by the train, and the infrastructure influence on train delays represented by a normalised value (which depended on factors such as the percentage of restricted speed sections, the number of junctions and the number of stations). The output data were the delays of each train; they were used as the target data in ANFIS. Pairs of input–target data were used to calculate the membership functions of the Takagi–Sugeno type fuzzy logic system. This approach allowed the ANFIS system to learn from the data that it is modelled after. The data were randomly dived into 3 sets for training, validation and verification. The ANFIS model was developed using the Matlab Neural Networks Toolbox. The fuzzy inference system was trained by the backpropagation optimisation method, and the output was generated by a linear membership function. Various structures of the ANFIS model were tested to find the best results, and the results generated by the ANFIS model were compared with actual train delays and verified by statistical methods (Table 1).

For the Rakovica station, the results were tested by two-sample Kolmogorov–Smirnov (K–S) tests. The K–S tests showed that the two data samples may reasonably be assumed to come from the same distribution. Additional statistical tests also verified that the results from the ANFIS model correspond to actual delays. The actual delays and the results of the ANFIS

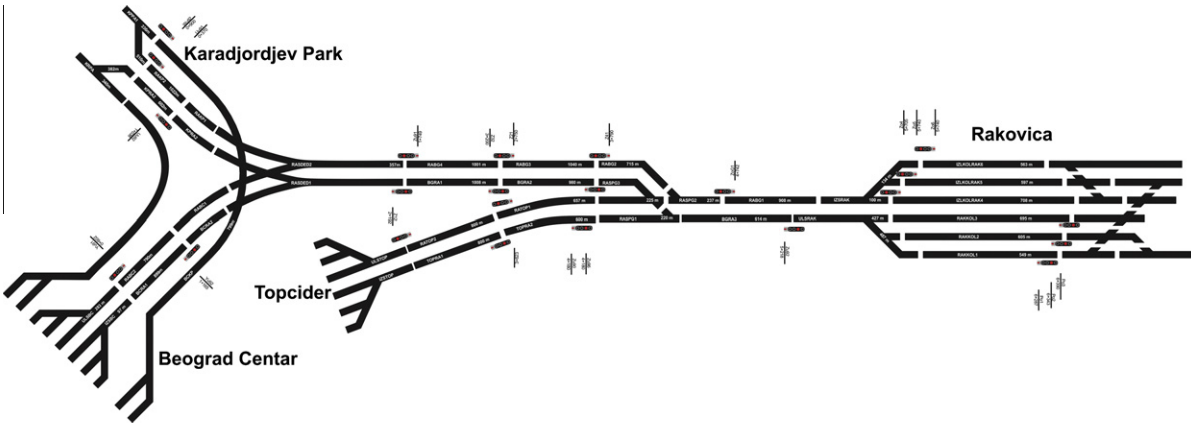

Fig. 8. The block section plan for an FPN model.

model correlated well, and the correlation coefficient (R value) was 0.866. When comparing actual train delays to delays generated by the ANFIS model (Fig. 7), we can see that the results are good for small delays. Delays of trains greater than 60 min were not predicted as precisely because the occurrence of these delays is smaller, and these delays were more difficult to forecast.

After the ANFIS model was verified, its structure, relations, rules and weights were used to model train delays using a module in the FPN model (Fig. 4).

5. Creating the FPN model of the railway system

Previously defined modules can be used to create an FPN simulation model of various railway systems. Creating the simulation model required connecting and arranging the modules according to the layout plan of the track sections [16].

When creating a model, one must take into account all of the rules and operating conditions of a railway system. The following algorithm is used to create the model.

- The railway system is analysed, and definitions are prepared for the fuzzy logic model and the data on timetable and infrastructure.
- An ANFIS model is created and trained with collected data on train delays. In case there are no data on train delays, it is necessary to conduct a survey on experts to determine the parameters of the Mamdani type fuzzy logic model.
- An FPN module for train delays is constructed according to the structure, relations, rules and weights of the fuzzy logic model for train delays defined in previous step.
- Modules that represent distinct types of sections in the model are created and defined.
- An HLPN graph is created by connecting the modules.
- Storage nodes that represent signals and states of the section are defined.
- Storage nodes are connected with modules, the initial values of the storage nodes are entered (such as the section lengths and maximum speed), and the time storage (the simulation clock) and random storage numbers are connected.
- The database that contains data on train timetables and train parameters in the model (input data) is defined.
- The database that stores data from the simulation program is defined.
- The animation window for the section state is created according to the section plan.

The external database stored input and output data. The simulation model sent the data regarding the movement of each train through the model along with the data regarding the section state (or the occupancy of each section). The database was customised to create quick reports based on queries and to filter data by train, section, signals or train delay time in the model. With programmed macros, data from the database were used to create train diagrams (or the time versus distance plots) with the vertical axis representing section lengths and the horizontal axis representing time.

The section states of the simulated runs were animated in the program. During the simulation, the stored parameters of each section changed when a train entered the section. These numerical data were used to animate the sections.

6. The application of a model

The model was tested on a part of the Belgrade railway node and was defined using the infrastructure data of the sections, the railway signal layout plan, and the timetable data from the year 2010. The boundaries of the model consisted of the Beograd Centar station, the Topcider station, the Rakovica station and the Karadjordjev Park station (Fig. 8). Three train categories were present in the model: freight, regional (including regional and suburban) and passenger (including long distance passenger trains).

6.1. Results and testing of the example model

For the Rakovica and Topcider stations, the primary delays were calculated using the FPN modules defined by the ANFIS system. Data on train delays from the train dispatchers' logs were collected and paired with data on the train category, the arrival time at the station, the distance travelled and the infrastructure influence. The results generated by ANFIS were highly correlated with the actual delays in stations Rakovica and Topcider, with a difference between actual and calculated delays of less than 5%. For the other two stations (Karadjordjev Park and Beograd Centar), the ANFIS model did not produce good results. The correlation coefficient (R value) was below 0.5 for both stations, and the difference between average delays was over 10%. The main cause of this result is that the number of trains in each of these stations was low (five times lower than in Rakovica and Topcider), and many delay outliers caused train delays calculated using the ANFIS model to be inaccurate. Additionally, stations Karadjordjev Park and Beograd Centar are located on the central part of Belgrade railway node and are mainly used for suburban and regional trains. Thus, the nature and causes of train delays in these stations are different. For the Karadjordjev Park and Beograd Centar stations, the Mamdani fuzzy system was created according to expert knowledge on train delays. The input data to the FPN simulation model were as follows: the timetable for the year 2010, the infrastructure data (including section lengths, section plans, restricted speeds and track routes), and the parameters for the FPN module used to determine train delays for each station and the initial marking of the PN.

During simulation, the data were stored in an external database. The results were categorised and filtered for analysis after the simulation. The data could be sorted by section, train number, train relation and train category. Multiple queries were used by the database to generate reports comparing the experimental results with actual data.

The FPN model was verified by analysing the simulation results. Results from the database were used to generate the train time versus distance diagram, where any irregularities in the model were easily identified (Fig. 10).

The animation window was modelled to visually resemble the interlocking control panel [19] of the dispatcher. Sections of the model were animated using the section state data (Fig. 9).

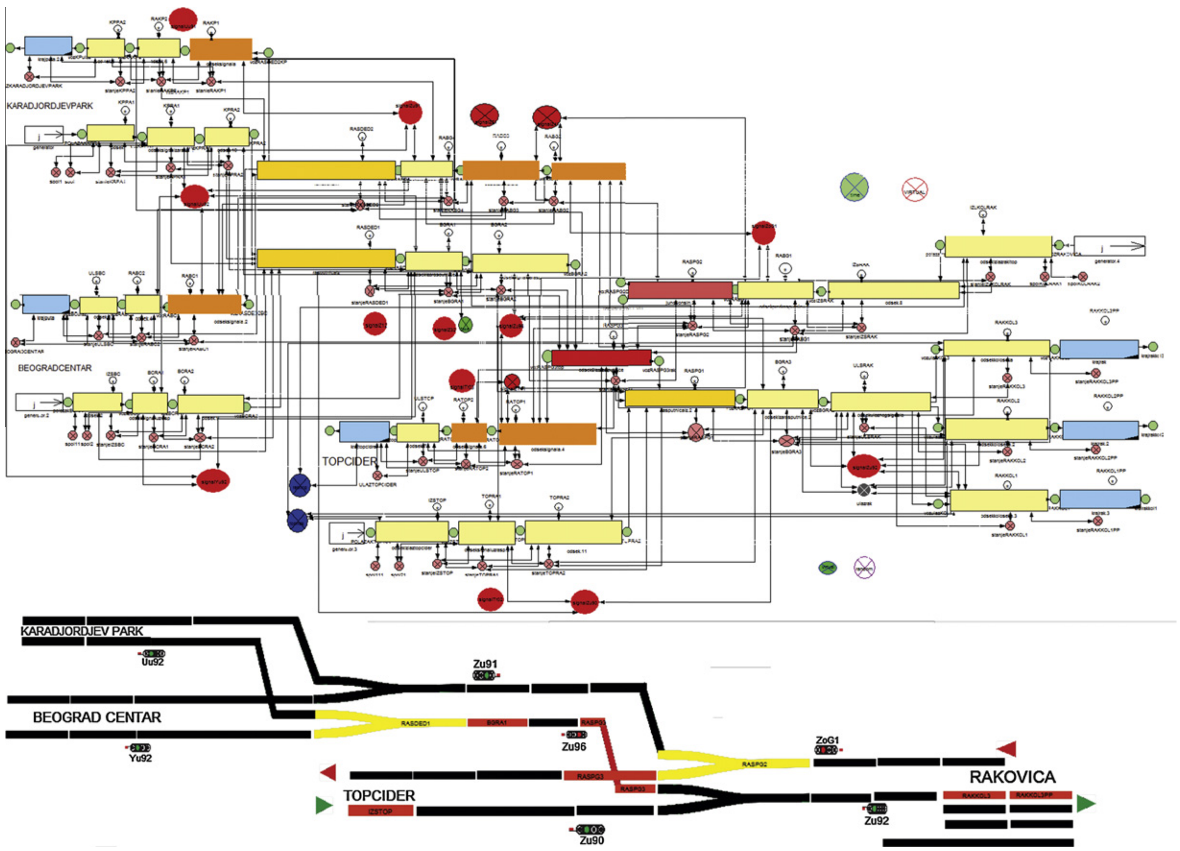

Fig. 9. FPN graph with the animation window.

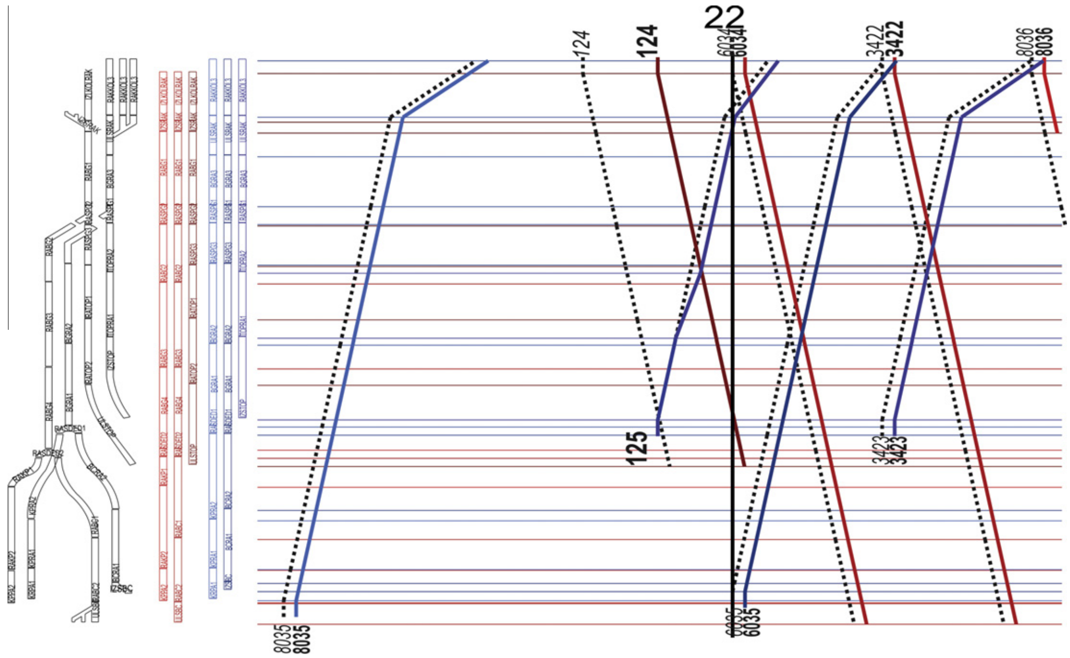

Fig. 10. Time versus distance graph generated using simulation results.

The FPN model calculated the train delay when the train entered the model. The propagation of train delays could cause knock-on delays for other trains. The simulated train traffic determined the conflicts between the train routes and identified sections where secondary delays occurred and the trains were involved in the delay. Data on secondary delays could be obtained by analysing the train time versus distance plot generated by the PN simulation results. Fig. 10 presents a part of the train time versus distance plot for sections of a model. Dashed lines represent timetable train routes, and continuous lines represent disturbed train routes in which the delay was calculated using the FPN. As presented in Fig. 10, the FPN calculated a delay for train 124 that disturbed other trains. Subsequently, trains could not depart because of the conflicting route with train 124. Train 125 had a secondary delay on the TOPRA1 section while waiting for train 124 to leave the junction section.

The average section occupancy data can indicate possible bottlenecks of the system. To determine sections with the highest occupancy, we have calculated the average occupancy for each section. From the 10 independent replications, we obtained the estimate of the average section occupancy with a relative error less than 0.05 and a 95% confidence level. The sections with the highest percentages of total occupancy are the following: RABG1 (4.6%), RASPG3 (4.5%), RASPG2 (3.2%), TOPRA1 (3.3%) and RASPG1 (3.0%). These sections are junction sections and are adjacent to the junction (Fig. 8). Results from the replications show that, with a relative error of 0.007 and a 95% confidence level, section RABG1 has a total occupancy of one train in the interval [85.6, 86.8] seconds. Most of the knock-on delays are generated on the section IZLKOLRAK for the trains that depart from Rakovica station. The average knock-on delay for trains departing Rakovica is 27 s, and the average knock-on delay for a train in the system is 18 s. Analysis of the data from the 10 independent replications shows that although trains have large primary delays, there are no significant bottlenecks or secondary delays in the system due to the low number of trains.

7. Conclusion

An FPN can be used to model systems characterised by ambiguity and uncertainty. Train delays can be modelled intuitively or using the ANFIS model trained by historical data. Systems with unknown behaviour and without known data can be modelled as fuzzy systems defined by the experience of experts (in this case, the train dispatchers and operating staff). Such a model depends on expert knowledge along with the timetable and infrastructure data to create suitable FPN parameters to determine train delays. In systems where the train delay data could be collected, the FPN module for train delay calculation can use parameters from the trained hybrid neuro-fuzzy ANFIS system. The advantage of this approach is that the delay is calculated for each train; in the statistical approach, the results for individual trains were less precise. The soft computing approach increases the time to implement the model but enables more precise results for each train. Testing FPN models and adjusting their input variables for the membership functions, the rule base and the output variables for the membership function can enable the creation of fuzzy PN models to determine train delays that produce quality results comparable to real system delays. Future research will be focused on improving the FPN simulation model by adding a module for train route conflict management that uses fuzzy logic to model the strategy of train dispatchers in traffic control. With a train conflict module, an FPN simulation model will be able to resolve eventual train route conflicts on a junction similar to a real system train conflict management.

Acknowledgments

The authors acknowledge the support of research Project TR 36012, funded by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

References

[1] W.M.P.v.d. Aalst, M.A. Odijk, Analysis of railway stations by means of interval timed coloured Petri nets, Real-Time System 9 (1995) 241–263.
[2] A. Bagdasaryan, Discrete dynamic simulation models and technique for complex control systems, Simulation Modelling Practice and Theory 19 (2011) 1061–1087.
[3] F. Barber, M. Abril, M.A. Salido, L. Ingolotti, P. Tormos, A. Lova, Survey of automated systems for railway management, in: Technical Report DSIC-II/01/07. (Technical University of Valencia, 2007).
[4] T. Basten, R. Bol, M. Voorhoeve, Simulating and analyzing railway interlockings in expect, IEEE Parallel & Distributed Technology 3 (1995) 50–62.
[5] Y.-H. Cheng, L.-A. Yang, A fuzzy petri nets approach for railway traffic control in case of abnormality: evidence from Taiwan railway system, Expert Systems with Applications 36 (2009) 8040–8048.
[6] W. Daamen, R.M.P. Goverde, I. Hansen, Non-discriminatory automatic registration of knock-on train delays, Networks and Spatial Economics 9 (2009) 47–61.
[7] A. Fay, A fuzzy knowledge-based system for railway traffic control, Engineering Applications of Artificial Intelligence 13 (2000) 719–729.
[8] R.M.P. Goverde, A delay propagation algorithm for large-scale railway traffic networks, Transportation Research Part C: Emerging Technologies 18 (2010) 269–287.
[9] I.A. Hansen, J. Pachl, Railway timetable & traffic: analysis – modelling – simulation, (Eurailpress, 2008).
[10] I.A. Hansen, R.M.P. van der Meer, D.J., Online train delay recognition and running time prediction, in: 13th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2010, pp. 1783–1788.
[11] K. van Hee, P. Rambags, P. Verkoulen, Specification and simulation with ExSpect, in: P. Lauer (Ed.), Functional Programming, Concurrency, Simulation and Automated Reasoning, Springer, Berlin/Heidelberg, 1993, pp. 296–328.
[12] J.-S.R. Jang, ANFIS: adaptive-network-based fuzzy inference systems, IEEE Transactions on Systems, Man, and Cybernetics 23 (1993) 665–685.
[13] C.G. Looney, Fuzzy petri nets for rule-based decisionmaking, IEEE Transactions on Systems, Man, and Cybernetics 18 (1988) 178–183.
[14] L.-G. Mattsson, Railway capacity and train delay relationships, in: A.T. Murray, T.H. Grubesic (Eds.), Critical Infrastructure, Springer, Berlin Heidelberg, 2007, pp. 129–150.
[15] S. Milinković, M. Marković, S. Vesković, M. Ivić, N. Pavlović, A Fuzzy petri net model for estimation of train delays, in: Proceedings of the 7th EUROSIM Congress on Modelling and Simulation. Prague, Czech Republic, 2010.
[16] S. Milinković, S. Vesković, M. Marković, M. Ivić, N. Pavlović, Simulation model of a railway junction based on petri nets and fuzzy logic, in: Selected Proceedings of the 12th World Conference on Transport Research Society, Lisbon, Portugal, 2010.
[17] T. Murata, Petri nets: properties, analysis and applications, Proceedings of the IEEE 77 (1989) 541–580.
[18] H. Nakanishi, I.B. Turksen, M. Sugeno, A review and comparison of six reasoning methods, Fuzzy Sets and Systems 57 (1993) 257–294.
[19] J. Pachl, Railway operation and control, VTD Rail Pub., 2009.
[20] A. Palit, D. Popovic, Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications (Advances in Industrial Control), Springer, 2005.
[21] S.N. Sivanandam, S. Sumathi, S.N. Deepa, Introduction to Fuzzy Logic using MATLAB, Springer, 2007.
[22] D. Teodorovic, Fuzzy logic systems for transportation engineering: the state of the art, Transportation Research Part A: Policy and Practice 33 (1999) 337–364.
[23] D. Teodorović, K. Vukadinović, Traffic Control and Transport Planning: A Fuzzy Sets and Neural Networks Approach, Kluwer Academic Publishers, Boston/Dordrecht/London, 1998.
[24] J. Yuan, I.A. Hansen, Optimizing capacity utilization of stations by estimating knock-on train delays, Transportation Research Part B: Methodological 41 (2007) 202–217.
[25] H. E. Virtanen, A study in fuzzy petri nets and the relationship to fuzzy logic programming, Reports on Computer Science and Mathematics, Åbo Akademi A, 162, 1995.