Delay Propagation in a Real Life Railway Network Controlled by a Fuzzy Logic Rule Base

Mark Farkas

Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Budapest, Hungary, farkas@tmit.bme.hu

Tibor Héray, Gábor Rózsa

Department of Automation, Faculty of Engineering Sciences, Széchenyi István University, Győr, Hungary, heray@sze.hu, rozsa@sze.hu

László T. Kóczy

Institute of Informatics, Electrical and Mechanical Engineering, Faculty of Engineering Sciences, Széchenyi István University, Győr, Hungary, koczy@sze.hu;
Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Budapest, Hungary, koczy@tmit.bme.hu

Abstract: This paper considers a real-life railway timetable related problem, where a set of interconnected railway junction points form a railway network, which is essentially a directed graph with corresponding vertex and edge capacities representing railway tracks and railway platforms.The dynamic behavior of this model is driven by a timetable. Unforeseen weather and other external effects may contribute to delays in the timetable, which alters the behavior of the whole system. In this scenario a hierarchical fuzzy system is proposed that can suggest a possible outgoing delay for each train by evaluating a set of fuzzy rule bases using two input data. The first proposal does not take into account the possible propagation of delay in the whole railway network. In this article a negative feedback is applied on the hierarchical fuzzy system.The fuzzy sets are optimized by an evolution based global search metaheuristics.

Keywords: railway timetable, connection conflicts, fuzzy hierarchical rule base, automated simulation, evolution, global search metaheuristics, bacterial memetic algorithms

1 Introduction

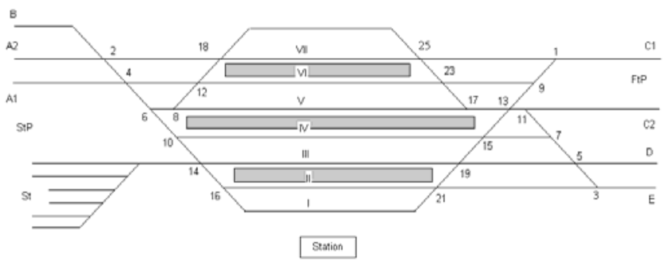
1.1 Real-Life Railway Networks

In a real life scenario, a set of railway junction points or railway stations are connected together with various types of railway lines. Railway junction points are basically vertices in a directed graph, where each vertex has an associated vertex capacity. This vertex capacity determines the number of platforms in a railway station (e.g. how many trains can wait at that station simultaneously). This is the basic model of a railway network. Trains are driven by timetables and these timetables implicitly define relations amongst the affected trains. Passengers and cargo traffic can be regarded as a complex network flow. The nature of this flow is determined by the timetable and the structure of the railway network [9,10,11]. Timetables may suffer from inaccuracies and delay due to the unforeseen change in the weather or other external effects, such as the strike of the union of railway workers in a neighbouring country. This delay has an effect on every connecting train along the railway network. The delay is propagated along the railway network and small delays are gradually aggregated into a much bigger one, which might render the timetable useless.

2 A Simple Approach

2.1 The Simplified Railway Network

In our previous articles [1,2,5] a simple hierarchical fuzzy rule base was proposed to give a solution to the relaxed form of the real-life problem mentioned above. In this approach, only single railway junction points were considered:



1. Figure
A typical real-life railway junction point

A railway station with eight tracks was examined, which is to be found in the connection point of a central-structured railway network. In the above model each line can either be a single track or a double track line and two types of trains are operated, namely slow and fast trains. Each train has a unique identifier associated to the train itself, which marks the type of the train and a unique number. From the aspect of the passengers and the cargo shipments a relation is defined by the timetable amongst the pairs of trains, which describes a connection relation: An incoming and an outgoing train is in a connected relation if the respective time of arrival and departure lie within one hour. In every other case trains are considered to be unrelated [2]. The primary objective of the proposed hierarchical fuzzy rule base is to determine the optimal outgoing delay for each outbound train by evaluating the timetable based connection time and the estimated incoming delay of every inbound train, which is in a connected relation with the outbound train.

$$G=(V,E);c:V\rightarrow\mathbb{N}^{+}$$
$$TR:=\{train_1,\dots,train_n\}$$
$$RW:=(G,c,TR) \tag{1}$$

2.1.1 Crisp Input Parameters

The crisp input parameters of the hierarchical fuzzy system are CT and ID.

a) CT: Connection Time

Connection time is basically the time difference between a pair of connected trains. E.g. the difference between the departure of the outbound train and the arrival of the respective inbound train.

$$CT(i,j):=T_{d_i}-T_{a_j} \tag{2}$$

b) ID: Incoming Delay

Incoming delay is the estimated amount of delay of an inbound train. This information must be available a priori to the decision support system in order to make appropriate suggestions to the outgoing delay. Incoming delay may change at any time and therefore it has to be updated periodically as the estimation becomes more and more precise.

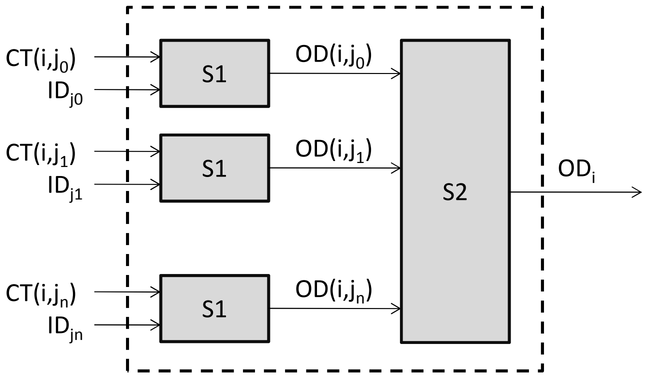
Each outbound train has a set of connected inbound trains, which all have an associated connection time and estimated incoming delay. These parameters form a crisp incoming delay and connection time vector:

$$Input=\{\langle CT_i\rangle,\langle ID_i\rangle\} \tag{3}$$

2.1.3 Crisp Output Parameters

The sole output parameter of the system is the outgoing delay time (OD), which is the amount of time the outbound train has to be delayed in order to wait for each inbound connected train.

2.2 The Hierarchical Fuzzy Model for Conflict Resolution



2. Figure
A hierarchical fuzzy model of the decision support system

In [1] a hierarchical fuzzy system was proposed, which is depicted in Figure 2 above. The hierarchical fuzzy model is composed of two separate subsystem, where only the S1 subsystem can be regarded as a real fuzzy system and S2 is solely a simple rule based machine.

Figure 2 shows the way of resolving a railway timetable related conflict of the set of pairs of interconnected trains, where each pair has a common outbound train, from the aspect of the outbound train. Each input symbol is a crisp time value and the only output symbol is the defuzzified time value which represents the amount of outgoing delay.

2.2.1 The S1 Fuzzy Subsystem

The S1 fuzzy subsystem [1,2] takes one pair of interconnected trains and evaluates the optimal amount of outgoing delay when only this pair is considered. First and foremost the input variables are fuzzified and the output is determined by a Mamdami-type inference engine [13]. This simple system is based on 6x5 rules of the following format:

IF ID={*ID class*} **AND** CT={*CT class*} **THEN** OD={*OD class*}

2.2.2 The S2 Fuzzy Subsystem

The S2 subsystem [1,2] cannot be considered a real fuzzy system. First, the fuzzy set is taken from the <OD_i> input vector which has the biggest associated time interval on the grounds of S1. Then the number of occurences of the selected maximal fuzzy set is evaluated. The rules are then constructed in the following format:

IF max#{OD_i = *OD class*} **THEN** OD={*OD class*}

Where #max denotes the number of occurences of the maximal fuzzy set as an outgoing delay.

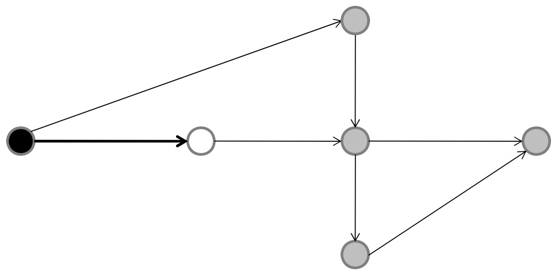
2.2.3 Problems with this Approach

The fuzzy system introduced above is only capable to resolve the most simple form of timetable conflicts using a partial timetable data extracted from a real-life railway traffic scenario and estimated incoming delays. Therefore this approach has several shortcomings:

a) The delay is not propagated along the series of interconnected trains. E.g. only the effect of a single delay is considered, but in real life scenarios waiting for an incoming train, which has suffered a certain amount of delay, has an additional cost: the delay will be propagated along the series of consecutive interconnected trains and the amount of delay is aggregated, which eventually leads to huge delays in the whole network.
b) Fuzzy sets and linguistic variables are provided on an ad-hoc basis, a suggested objective function, total aggregated cost of waiting [4] may help to improve the fuzzy system by determining the optimal membership functions and extending the rules. A method was suggested to this problem in [2].
c) The associated capacity of each vertex must be taken into account when a delay decision is made. Each railway station has a predefined number of platforms that determines the number of trains that can stay at a given station simultaneously. This vertex capacity is not considered in this article but can be a subject of further investigation.

3 A More Complex Model

3.1 Delay Propagation



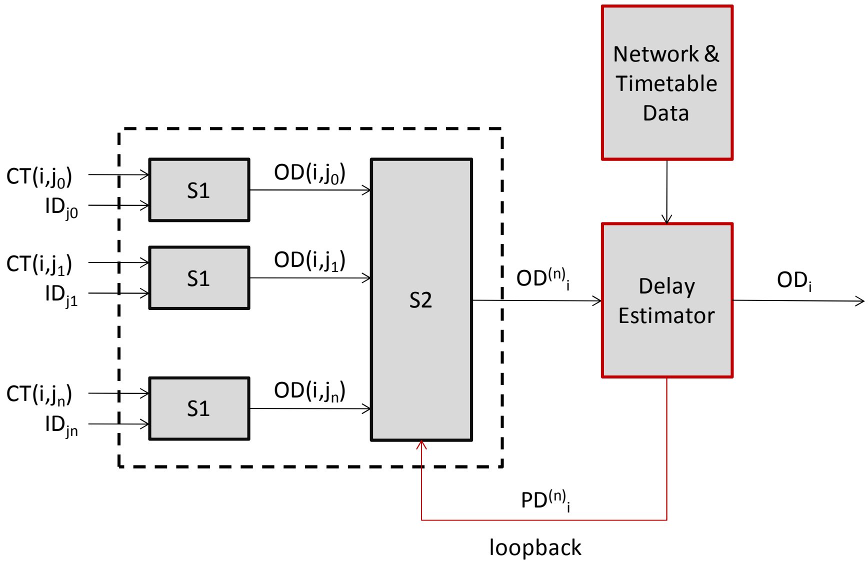
3. Figure
An extracted subnet of the railway network

Delays are propagated along the series of interconnected trains in the railway network. Figure 3 shows an extracted subnetwork of a railway network graph. In this example, when a train suffers an outgoing delay at the station maked with black color, due to incoming delays in the case of the related inbound trains, then the related trains waiting for the delayed train at the black station also have to wait. Therefore the delay is propagated along the lines marked with black-gray and black-white. These trains also suffer a certain amount of delay and their own delay is added to the amount of incoming delay. As the delay is propagated along the series of consecutive connected trains in the railway network, the aggregated delay becomes more and more enormous and after a while it breaks the one hour limit. Trains that are otherwise connected will miss their connection. The propagation of delay has a distinct distruptive effect on the whole railway network.

This effect is due to the fact that only the inbound side of the delay was considered in the previous hierarchical fuzzy model based decision support system and the outbound delay propagation was not taken into account at all.

To subdue this problem, the concept of Propagated Delay (PD) is introduced, which is basically the estimated aggregated delay of the trains at the outgoing side of the railway station. The estimation is done in a similar fashion to the Incoming Delay (ID) and only adjacent railway stations are considered as the proper estimation of the propagation delay would require the evaluation of the whole connection tree. Propagation Delay is a random variable with an exponential distribution and a constant, which is basically the amount of the associated Outgoing Delay.

3.2 Hierarchical Fuzzy System with Negative Feedback



4. Figure
The modified fuzzy system with a negative feedback

The system described in 2.2 is altered by introducing a delay estimator and a negative feedback loop to the S2 subsystem. The output delay is calculated in an iterative way. The iteration is continued until either no improvement can be obtained by the subsequent iteration or the propagation delay time is considered to be sufficiently low and maintains the stability of the network.

3.2.1 Delay Estimator

The role of the delay estimator, as its name might suggest, to estimate the delay propagation (PD) by evaluating the maximum of the estimated incoming delays of each adjacent connected trains and the predicted additional delay based on a random variable with exponential distribution. This value is fed back to the S2 subsystem in the next iteration cycle. The delay estimator aquires required datas from the network connection graph and the timetable itself.

3.2.2 Modified S2 Subsystem

A new linguistic variable is defined for the propagation delay. The associated fuzzy membership functions can be the same as in the case of outgoing delay (OD) because semantically both linguistic variable express a delay.

The S2 subsystem need to be extended with 6x6 rules, where each rule has the following format:

IF OD(n) = {OD class} **AND** PD(n) = {PD class} **THEN** OD(n+1) = {OD class}

These rules basically express the following: the greater the outgoing delay is the greater the propagation delay will be, thus one has to decrease the outgoing delay in the next iteration in order to cope with the distruptive effect of the delay propagation.

One of the key differences between the feedback of propagation delay and the optimization of fuzzy rules and membership functions described in [2] is that in the case of the optimization statistical traffic datas were utilized and the fuzzy system was modified after (a posteriori) the evaluation of outgoing delay. However in the case of the former method up to date delay information is fed back into the fuzzy system, which alters the outgoing delay directly. These two methods can be combined to achieve the robostusness of the hierarchical fuzzy system with a feedback and yet keep the learning capabilities and adaptivity of the fuzzy system combined with a bacterial memetic algorithm.

4 Implementation and Results

4.1 XFL3 and C# Language Implementation

RailwayConflict v.1.1, which is an improved version of the cross-platform decision support system described in [1], can automatically calculate outgoing delay (OD) values based on the incoming delay vector and connection time vector.

The program is developed in .NET Framework and adopts a hybrid approach. The iteration cycle is the following: S1 subsystem generates outgoing delays for each incoming line from their corresponding income delay and connection time. Then S2 subsystem determines the next iteration of outgoing delay based on the outgoing delays from S1 subsystem and the loopback propagation delay value (PD). S2 is basically a mixture of a crisp decision table and a fuzzy rule base with 2 input variables (OD and PD) and one output variable (OD).

The S1, S2 subsystem and the Delay Estimator together form a hierarchical fuzzy control / reasoning system with a negative feedback.

The inference engine is described in a opensource fuzzy system description language XFL3, which is a declarative language. In the next step, a compiler translates the XFL3 declaration into an adequate ISO/IEC 23270:2003 compliant C# language code. This transcoder is based on the Java compiler of the Xfuzzy 3 toolkit.

4.2 Results

a. Railway station connection matrix:

The railway station connection matrix contains each line, its name and the respective connection times and incoming delays in an RLW file, which has the following format:

'#x stationname'.RLW:

LINE	'#1 outgoing line name'			
CTS	*CT1*	*CT2*	…	*CTn*
IDS	*ID1*	*ID2*	…	*IDn*
…				
LINE	'#m outgoing line name'			
CTS	…			
IDS	…			
…				

Where CT and ID values are numeric values between 0 and 60 minutes.

b. Railway network:

The railway network file (RLN) defines list of stations and the type of interconnection, namely the type and length of railway lines between the adjacent stations:

MODE	'geographic', 'cartesian'
STATION	'#1 station name'
LONGLAT	'longitude', 'latitude' *or*
XYZ	'x', 'y', 'z'
...	...
STATION	'#n station name'
LONGLAT	'longitude', 'latitude'
...	
LINE	'#1 line name'
TYPE	'single', 'double'
CONNECTS	'start station name', 'end station name'
...	
LINE	'#o line name'
TYPE	'single', 'double'
CONNECTS	'start station name', 'end station name'
...	

The STATION tag defines the stations by their name, and their respective coordinates in a geographic coordinate system (DMS, DM or DD mode) or a cartesian coordinate system, this behaviour is altered by the MODE tag. The LINE tag defines the name of the lines connecting the pairs of stations, these names must be the same as the names of the connected stations' corresponding incoming and outgoing lines. The type of the railway line is defined by the TYPE tag.

The RLN (railway network file) is the same across the whole network. The decision support is always made respect to a railway station and not the network, thus the program is executed in the following way:

RailwayConflict.exe [NetworkFile].RLN [StationName]

One possible output may be the following:

```
Line: S22
---------
# 0   CT:   00:41   ID:   00:10   OD:   00,0 [min]
# 1   CT:   00:36   ID:   00:00   OD:   00,0 [min]
# 2   CT:   00:34   ID:   00:14   OD:   00,0 [min]
# 3   CT:   00:31   ID:   00:22   OD:   00,0 [min]
# 4   CT:   00:28   ID:   00:05   OD:   00,0 [min]
# 5   CT:   00:25   ID:   00:03   OD:   00,0 [min]
# 6   CT:   00:11   ID:   00:10   OD:   07,5 [min]
Total:   08 [min]

Line: F4
--------
# 0   CT:   00:54   ID:   00:37   OD:   38,0 [min]
# 1   CT:   00:48   ID:   00:22   OD:   00,0 [min]
Total:   RE
```

The program and its source code are available under the GPL license from http://ktsnet.hu/~fmark/RailwayConflict_FIS_v11.zip.

Conclusions

The method and the implemented program proves the viability of the concept of utilizing a hierarchical fuzzy logic system in the dispositional tasks of a real-life railway traffic situation. The system presented in this paper could be further optimized by the usage of bacterial memetic algorithms as it was shown in a simpler case without a propagation delay estimator and feedback in our previous paper [2]. A further and last improvement may concern the subject of vertex capacities and the introduction of a fuzzy rule base optimization technique.

Acknowledgement

This paper was supported by a Széchenyi University Main Research Direction Grant 2009 and National Scientific Research Fund Grant OTKA K75711.

References

[1] Héray, T., Rózsa, G. and Kóczy, L. T.: The application of fuzzy logic for the treatment of conflicts in the dispositional tasks of a railway traffic control center, 9th IEEE AFRICON Conference 2009, Nairobi, Kenya

[2] Farkas, M., Héray, T. and Kóczy, L.T.: The improvement of an existing fuzzy logic rule base for the treatment and simulation of conflicts in the dispositional tasks of a railway traffic control center, IEEE SOFA Conference 2009, Arad, Romania

[3] L. T. Kóczy, K. Hirota and L. Muresan: Interpolation in hierarchical fuzzy rule bases, *J. Intell. Fuzzy Syst 1* (1999) pp. 77-84

[4] P. Vansteenwegen, D. Van Oudheusden: Developing railway timetables which guarantee a better service, *European Journal of Operational Research 173* (2006) pp. 453-463

[5] T. Héray - G. Rózsa - L. T. Kóczy: The possible applications of fuzzy logic for the treatment of conflicts in the dispositional tasks of railway traffic control center, 8th IEEE AFRICON Conference 2007, Windhoek, Namibia, 6p.

[6] L. A. Zadeh: Fuzzy Sets, *Inf. Control 8* (1965) pp. 338-353

[7] L. A. Zadeh: Outline of a New Approach to the Analysis of Complex Systems and Decision Processes, *IEEE Trans. Syst. Man and Cybernetics SMC-1* (1973), pp. 28-44

[8] K. Nachtigall – S. Voget: Minimizing waiting times in integrated fixed interval timetables by upgrading railway tracks, European journal of operational research 103 (1997) 3 pp 610-627

[9] J. Opitz - K. Nachtigall: Automatische Erzeugung von konfliktfreien Taktfahrlagen, *Eisenbahningenieur* 58 (2007) 7 pp. 50-55

[10] U. Grabs: Konflikterkennung und -lösung für dispositive Aufgaben in Betriebszentralen, *Signal und Draht*, 87(1995) pp. 255-256

[11] R. Hundt: Automatische Konflikterkennung und -lösung dispositiver Aufgaben in der Betriebsleittechnik, *Signal und Draht*, 87(1995) pp. 437-438

[12] P. Hille: Konfliktlösungsmodelle, *Signal und Draht*, 91(1999) pp. 15-18

[13] L. T. Kóczy and A. Zorat: Fuzzy systems and approximation, *Fuzzy Sets and Systems 85* (1995) pp. 203-222