



Conflict resolution in railway traffic control by a distributed intelligence approach

M.Mazzarello, R.Copello

On AIR s.r.l, Italy

Abstract

Modern railway traffic control systems need fast and reliable conflict resolution capabilities in order to produce optimised schedules when normal ones are not suitable. We propose here a distributed intelligence approach in which global optimisation is achieved by co-operation of local resolution centres. The approach was validated by means of two test cases in real environments. The work was performed under the MARCO project in the Telematics framework.

1 Introduction

This paper deals with the problem of conflict resolution in railway traffic control. The work was performed under the project MARCO (Multilevel Advanced Railways Conflict resolution and Operation control), financed by EC-DGXIII [1]. The main objective of MARCO project was to find suitable algorithms to optimise train traffic when normal schedules are somehow disturbed, i.e. trains may be delayed, or tracks may be interrupted [2]. The traffic problem was modelled as a constraint satisfaction optimisation problem [3].

The proposed solution is able to detect and solve conflicts, producing optimised schedules in real time. It relies on a distributed intelligence approach in which control intelligence is spread over the network nodes through local conflict resolution centres (CRC) [4]. They know the state of the controlled resources, the users applying for them, and the state of the neighbouring nodes. The search for a good solution of a traffic problem is accomplished by the integration between local solving strategies and global optimisation criteria.

The algorithm was tested over real data provided by the end-users. Two test cases were analysed: a High Traffic Area (HTA) which includes complex junctions, stations, local lines, and a Global Area Network (GAN), which



includes several HTAs and main lines. Two demonstrators were developed to validate the approach: the HTA demonstrator (managing the Milan Junction node), and the GAN demonstrator (managing the Belgian network from Brussel to the German border). Both were tested in different scenarios, with many traffic problems, due to delayed trains, single and double track blocked for a long time in overcrowded parts of the networks. The results are very encouraging, in producing feasible circulation plans for even quite complex conflicts.

The paper is organised as follows. In section 2 we introduce the conceptual framework of the approach. Then in sections 3 and 4 we describe the demonstrators used for algorithm testing and validation. Section 5 is devoted to results and further work about conflict resolution techniques in railway control.

2 Distributed intelligent search

2.1 Motivation

Train traffic control is a complex task, involving computationally intractable problems and requiring great skill in analysing conditions and situations. Train traffic control includes large-scale dynamic and combinatorial problems, hard to be analysed by conventional approaches. Flexible simulation and intelligent decision support are considered useful in solving these problems [5].

In MARCO project different methods for optimisation purposes have been analysed. They can be divided in two classes with respect to their specific features: global and local methods. The analysis of a suitable number of algorithms (*Branch and bound*, *Branch and cut*, *Random search*, *GENET*, *Intelligent search*, *Distributed intelligent search*, *Simulated annealing*, *Genetic algorithms* [6]), led to adopt the most promising algorithms in realistic cases.

Taking into account as indicators the resolution and computing performances and the capability of managing different conflicts types, for the HTA demonstrator a greedy rule-based method (*Distributed intelligent search*) was adopted. It is fast, solves all the conflicts present in test scenarios and imitates the decision process of a human train dispatcher, causing minimal changes to the regular timetable [7]. Also the GAN optimisation kernel was based on the same method, improved and specialised to cope with GAN requirements. In the distributed intelligence approach the search for a feasible solution of a given traffic problem is accomplished by the integration between local solving strategies and global optimisation criteria. The main functional components of the optimisation kernel are:

- an intelligent search engine, including the set of control rules and a rough simulator, based on a simplified model useful to foresee the consequences of the control actions;
- an internal evaluator that guides the choice between different partial solutions, taking into account the performance indexes.

The control problem is modelled in terms of users that apply for resources to meet their schedules and of managers that are individually responsible for



allocation of assigned resources. The final control goal is to reduce the weighted delay of trains and the cost/performance ratio in railway operations.

2.2 Conflict Resolution Centres

Control intelligence is distributed throughout the network, where each node stores only local information and can communicate with the neighbouring nodes. Each node is also supplied with decisional power in order to perform local control actions. We can see it as a Conflict Resolution Centre (CRC) which knows the state of the controlled resources, the users applying for them, and the state of neighbouring nodes. So the elements of the model are: *resources* (block sections), *users* (trains) competing to obtain resources and *managers* (CRCs) allocating resources.

CRCs are located where the line is characterised by high complexity in traffic data or different routing choices (mainly in stations). Each CRC oversees a fixed number of block sections and tracks so that each block section (or track) belongs to a CRC (and only to that CRC). Many CRC can be assigned to a station (for instance, one CRC for each line direction).

The main characteristics of CRCs are:

- control strategy: each CRC is provided with local decision capacity, but a global agreement must always be achieved;
- conflict avoidance: CRCs attempt to avoid conflicts, implementing local control rules and suggesting control actions to be accepted by linked CRCs;
- conflict resolution: when an user applies for a resource two actions can be triggered: allocation on request or refusal of allocation. Each action is the result of a local decision process involving one or more CRCs handling the resources.

Conflicts can be *internal* or *external*. A conflict is internal if the involved trains belong to the same CRC, otherwise it is external. Internal conflicts are solved delaying or stopping the trains, by applying the dynamic priorities of the trains. External conflicts are managed by a comparison between the dynamic priorities associated to the CRCs. Both train priority and CRC priority are dynamic quantities, varying with traffic intensity and complexity, and can be evaluated locally in a straightforward manner.

2.3 Priority definition

For each CRC we consider both static and dynamic data. Static data describe the topology of the CRC network, identifying for each CRC the station where it is placed, the linked block sections, the controlled tracks and the linked CRCs. A static priority is assigned to each CRC, and it expresses the weight of the CRC within the network.

Dynamic quantities are traffic dependent. The dynamic state of each train is expressed by means of the following quantities:

- position and delay of the train (if occurred)



- number of stations that have still to be visited by the train T_k (for dynamic priority evaluation): $Numst_k$
- dynamic priority: $P_d(T_k) + Numst_k$

$$P_d(T_k) = A[P_s(T_k) + kf(\Delta_k) + \lambda \max\{P_s(T_k)^q\}] + BP_d(\text{lastCRC})$$

where:

$$P_s(T_k)$$

static priority of the train

$$f(.)$$

delay-weighting function

$$\Delta_k$$

delay of the train T_k

$$P_s(T_k)^q$$

static priority of the q th train queuing up after T_k ,
accessing the same CRC

$$A, B, k, \lambda$$

normalisation constants

For each CRC the following dynamic data are provided.

- list of trains inside a CRC
- list of trains asking for a CRC
- condition of block sections overseen by a CRC (*free/ engaged/ booked/ locked*)
- condition of tracks overseen by a CRC (*free/ engaged/ booked/ locked*)
- list of solutions suggested by the previous CRC
- final conflict solution
- dynamic priority: $P_d(\text{CRC}) = P_s(\text{CRC}) + P_d(T_i)$.

This is a dynamic parameter expressing, at any time, the amount of traffic controlled by a CRC.

2.4 Distributed intelligence model and conflict detection

Figure 1 outlines the main steps of the distributed intelligence approach. The first step of the model consists in detecting conflicts, due to traffic perturbations (delays, track faults), by means of a traffic simulation algorithm.

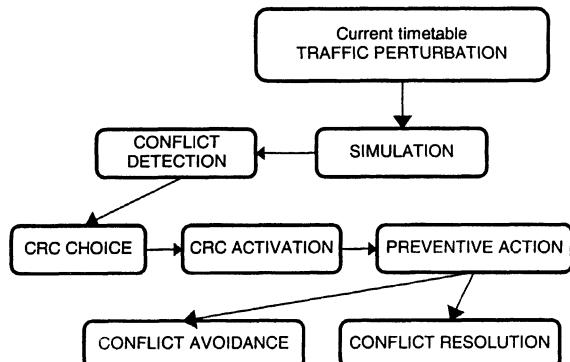


Figure 1: distributed intelligence model



The simulation algorithm:

- starts when a traffic mismatch is pointed out;
- lasts till a conflict is detected;
- is performed on the basis of the current timetable and assumes that the train dynamic is unchanged;
- provides a matrix $[T]$ concerning the future timetable:
 $(t)ij=1$ if only one train is in the block section i at time j
 $(t)ij=-1$ if more than one train is in the block section i at time j
 $(t)ij=0$ otherwise (empty block section at time j)

The simulation algorithm gets the following data:

- trains that are late: $w1_1, \dots, w1_n$;
- start of delays (hours): $t_delay_1, \dots, t_delay_n$;
- amount of delays (seconds): $delay_1, \dots, delay_n$.
- locations of the train $w1_i$ at time t_delay_i : bs_delay_i ;

and provides the following foreseen values:

- trains in conflict with the late trains $w1_1, \dots, w1_n$: $w2_1, \dots, w2_m$ (a train $w1_i$ can be in conflict with more than a train);
- location of the conflict between $w1_i$ and $w2_j$: $bs_conflict_{i,j}$;
- time interval of train “overlap” in $bs_conflict_{i,j}$ (seconds): $overlap_{i,j}$.
- location of the station in which the train $w1_i$ missed a link with another train, and identifier of this train : st_miss_i , t_miss_i .

2.5 CRC choice

The next task in distributed intelligence search consists in choosing the most promising CRC that attempts to avoid (or to solve) the first foreseen conflict in time [8]. To better explain the approach, we can suppose that:

- the first traffic problem is at time t_0 ;
- two trains w_x and w_y are in conflict in $(CRC)_z$ (the final one) at time t_z .
- from t_0 to t_z , the trains w_x and w_y go through, respectively, N_x and N_y CRCs.

Then we use the train w_i with the highest N_i in order to choose the most promising CRC. At time t_0 , the train w_i is in $(CRC)_i$, at station ST_k . If the train w_i is leaving ST_k , we choose the next $(CRC)_{i+1}$, otherwise we choose $(CRC)_i$.

The main goal of the chosen CRC is conflict avoidance, based on:

1. generation of the feasible control actions. Local rules that implement allowable control actions on the current time table are tried: delay of trains, re-route of trains, use of recovery time.
2. testing of the proposed control actions by means of local simulation and agreement of linked CRCs.

The above steps are repeated by each linked CRC, up to the last $(CRC)_z$. If the global agreement is achieved, then the local preventive actions can be implemented and the conflict is avoided. Otherwise, the conflict must be solved by the last CRC (i.e. the CRC where the conflict should take place).



2.6 Conflict resolution algorithm

For each late train $w1_i$, the algorithm tries to solve the first (in time) conflict involving $w1_i$ ($w1_i - w2_j$), as the next ones can be conflicts risen from the first one. The resolution phase takes place within a time interval, called *resolution window*, from the first t_{delay}_{ij} to the last t_{end}_{ij} where t_{end}_{ij} is defined as the maximum between a fixed time after $t_{conflict}_{ij}$ (15 min) and the end of the schedule of the trains $w1_i$ and $w2_j$, assuming free running. We take into account such window, extending as far as t_{end}_{ij} , in order to validate the resolution inside a window large enough to forecast the effects of the resolution (by means of agreement of linked CRCs).

In the following we describe how the algorithm solve the first conflict, involving the trains $w1_i$ and $w2_j$. To simplify the notation, we will denote t_{delay} , $t_{conflict}$, and $overlap$ instead of t_{delay}_{ij} , $t_{conflict}_{ij}$ and $overlap_{ij}$.

Within the resolution window, for each pair of trains in conflict, the chain of CRCs involved is investigated. Each chain is defined by the route of the train and is identified by the block sections of the routing. The resolution process is based on a proper integration between local resolution rules, implementing local control actions, and the centralised detection function, applying global verification criteria.

2.7 Resolution rules

The kernel of the resolution algorithm is an inference process based on three rules implementing the elementary control actions that can be applied to solve a rail traffic conflict: use of recovery time, re-route of trains, delay of trains. Each overall conflict solution results from a combination of the three basic rules.

Rule I (recovery rule) can be applied to the train $w1_i$ if its timetable includes recovery time > 0 . The rule is applied to the late train even if no conflict is foreseen. It is only assured that the speed increase does not cause a conflict, controlling that the train recovers at least $overlap$ seconds, from t_{delay} to $t_{conflict}$. If such conditions are true, the rule I can be applied as follows:

- for each CRC belonging to the chain $chain1_i$ of the train $w1_i$, starting from the CRC including bs_{delay}_i
 - for each block section of the current CRC
 - calculate the residual time to be recovered, taking into account that the departure time must be observed in station;
 - apply the allowed recovery time;
 - test if a conflict occurs
 - the linked CRC receives the new timetable.
 - If no conflict occurs, the rule I is successfully applied.

Rule II (re-route rule) can be applied to trains $w1_i$ and $w2_j$ if the conflict occurs on a part of the network where re-routing is admitted. The main steps are:



- starting from the CRC including the conflict or from the previous one, it is controlled if re-routing is possible
- if it is possible, a re-routing model is applied to the current train (starting from wI_i). The re-routing model defines the new *stay_time* for wI_i on the block sections of the new CRC (the CRC belonging to the new train route).

Rule III (delay rule) is applied if neither Rule I and Rule II are feasible, nor a combination of them. First of all we calculate the amount of time we have to consider to avoid the conflict: it is t_delay . Then we apply the Rule III at the train with the lower dynamic priority, as follows:

- for each CRC of the chain starting from the CRC including *bs_conflict*
 - for each block section of the current CRC
 - delay the train as much as possible, without changing the scheduled sequence of trains;
 - calculate the residual delay to be applied;
 - the linked CRC receives the new timetable.
- If the applied total delay is less than t_delay , proceed with a deeper search, combining the rules and/or lengthening the stopping time in the station nearest to the conflict, up to the residual t_delay .

The integration of the rules within the distributed intelligence framework enables to find a global optimised solution. Two criteria have been implemented in the inference process, in order to obtain different solutions that can be submitted to the dispatcher. The first one is based on the sequence criterion: the application of the delay rule keeps a FIFO sequence. The second observes the priority criterion: the highest priority train leaves the station before the others. Overtaking between trains is applied only if the station layout makes it possible.

3 HTA demonstrator

The HTA demonstrator deals with conflicts in a part of the Milano Centrale junction, one of the most complex in Europe. It investigates traffic disturbances (track unavailability and train delays) causing conflicts such as headway, overtaking and routing. Conflicts occur in a limited area with high train rate and many constraints to be satisfied. Passenger traffic is based on six timetables. Moreover, there are freight trains, crossing through the area. Almost all the lines are double track and some have four or more tracks. In a limited area (main diameter is 25 Km) there are 10 stations. Trains running on the node can be high speed trains, inter-city trains, local trains and freight trains. Within the considered time-frame (from midnight to 9 a.m.) about 200 trains are running, with an average of 40 trains per hour.

The basic elements of the demonstrator model are stations and block sections between stations. The features describing each block section are length (the shortest ones are 200 meters long) and direction (single/double). The junction can be divided into tree logical areas.



Area 1 is the most complex one. The main feature is its re-routing capability, due to alternative tracks for each traffic direction. The re-routing is applied to solve locked tracks problems and to optimise the traffic in consequence of late arrival of trains. Area 2 is mainly used by freight trains and its block sections are longer than any other. This causes headway conflicts and reduces the capacity of the lines. Area 3 is simpler but it is more constrained. There are only two types of tracks, where legal and illegal directions are possible. So, the re-routing control action is adopted only to solve track faults problems.

The HTA demonstrator forecasts the railway traffic and detects conflicts caused by traffic disturbances. Then it solves conflicts, taking into account constraints and capabilities of the involved areas, and creates a new conflict free timetable. A user interface provides the dispatcher with a step-by-step explanation of the decision process. The dispatcher can evaluate the proposed solutions by means of performance indexes that describe both the initial traffic status (number of the late trains, total initial delay, locked tracks and duration of the traffic interruption) and the final status (trains involved in proposed control actions, total added delay).

4 GAN demonstrator

The GAN demonstrator manages the major part of the network from Brussel to the German border, including the main line Brussel-Liege and the relevant alternative lines. The most important conflict is the temporary unavailability of tracks due to maintenance works, incidents or accidents. The possible control actions are: to set-up the single line operation, to re-route trains via alternative lines or to organise shuttle services. The problem of locked tracks leads to load peaks in other parts of the network, as a consequence of traffic re-routing. A good solution of such conflicts requires a global approach.

The general aim of the GAN demonstrator is to react to a track locking problem occurring along the main line. The objectives of the search procedure for a new optimised traffic plan are the reduction of the total delay (weighted by the number of passengers), both of trains directly affected by the problem and of trains indirectly involved, and the avoidance of local load peaks.

4.1 The GAN optimisation kernel

To build the optimisation kernel, a *problem-solving* method was considered the most suitable approach. From the analysis of GAN characteristics, goals and constraints, the chosen optimisation model was a *propose-and-revise* model, where two kinds of knowledge are represented: knowledge proposing solutions and knowledge revising solutions.

The knowledge proposing solutions is further on characterised by:

- knowledge analysing the problem of train traffic regulation;
- knowledge identifying the trains that will be affected by the traffic problem and classifying them according to static and dynamic properties;
- knowledge applying local solutions, specific for the given classes of trains;



- knowledge supporting locally the choice of the most promising solution, on the basis of performance criteria.

Also the knowledge revising the solutions can be more detailed as follows:

- knowledge performing conflict detection capability, to identify traffic conflicts caused by the proposed solutions;
- conflict resolution knowledge;
- knowledge evaluating the solutions from a global point of view, considering the load balance on the whole area network.

The input of the optimiser is the reference circulation plan, representing the normal traffic without perturbations. The optimiser produces a new circulation plan including the proposed solution, described in terms of train routings, timing parameters (new arrival/departure times) and control actions. The optimiser takes in account: line constraints (capacity), traffic constraints (single track), train constraints (related to priority and type of trains). To be more effective, the optimiser includes the experience of train traffic operators, acting as a decision support for the dispatcher, providing explicit information about the solutions.

The analysis suggested a design of the GAN optimisation kernel based on the *knowledge* approach, to represent the experience of the dispatcher, and on the *distributed intelligent search* framework, to integrate local solutions in a global optimised solution. The *distributed intelligent search* approach implemented in HTA demonstrator produced encouraging results and suggested to adopt such kind of approach also in the GAN optimiser.

The analysis of the available knowledge provided a basic scheme modelling the human approach. At first, the track interruption problem must be characterised according to the type (single or double track), the duration, the location. Then the applicable basic control actions must be examined. The core of the reasoning is the selection of the best combination of control actions. The selection must be accomplished from a global perspective (see Figure 2).

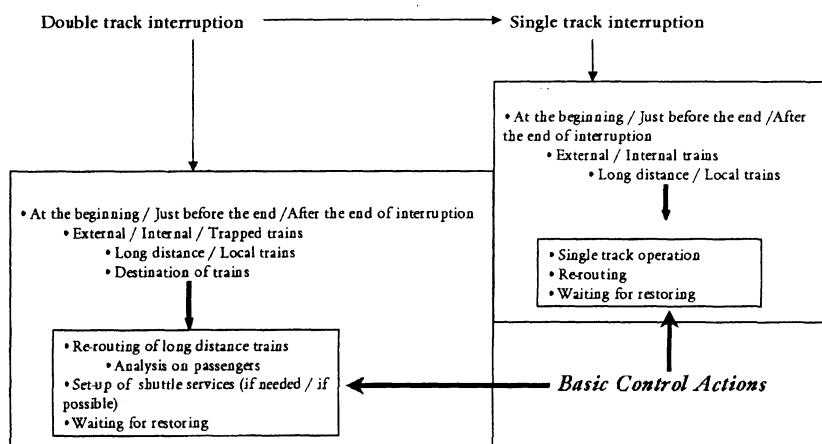


Figure 2: reasoning model



5 Results and conclusions

Tests on HTA and GAN demonstrators have been carried out for realistic scenarios. They were witnessed by an experienced dispatcher, who analysed in detail every solution [9]. The demonstrators were able to detect conflicts and find feasible solutions (HTA) and to re-route trains and detect and solve conflicts originated by interruptions involving single or double tracks (GAN). Computational time was very short (2-3 sec. for HTA and 10-15 sec. for GAN) and fulfilled the requirements of a real operational environment.

The MARCO research goes on with the new COMBINE project. The latter deals with the conflict detection and resolution concept in a moving block environment. The lack of real moving block systems and, consequently, of any practical experience in this field, has suggested to limit the COMBINE objectives to the lowest regulation level. The moving block system implies a train-to-ground radio link. This enable to give the drivers clear indications about an "optimised" train speed in terms of "suggested speed" regularly updated by the control centre or about "space/speed target" sent typically only once, when the need of changing train speed is detected. A very important result of this technique is that the objective function to be optimised can consider important parameters, such as energy consumption and brakes wear, in addition to train delays.

References

- [1] Savio, S. Detection and resolution of traffic conflicts in rail and metro. The EU project MARCO, *Business briefing: global mass transit system*, World Market Series, World Market Research Centre, London, 1999.
- [2] De Vilder F. MARCO: A European project to develop tools for conflict detection and resolution in railway and metro applications, *Proc. 8th IFAC Int. Symp. on Transport Systems*, Chania Greece, pp. 1209-1214, 1997.
- [3] Kumar, V. Algorithms for Constraint-Satisfaction Problems: A Survey, *AI Magazine*, (1), pp. 32-44, 1992.
- [4] Vernazza, G. & Zunino, R. A distributed intelligence methodology for railway traffic control, *IEEE Tr. Vehicular Tech.*, **39**(3), pp.263-270, 1990.
- [5] Dutta, A. Integrating AI and optimization for decision support: A survey, *Decision Support Systems*, **18**, pp.217-226, 1996.
- [6] Davenport, A. & Tsang, E.P. GENET: a connectionist architecture for solving constraint satisfaction problems by iterative improvement, *Proc. 12th Nat. Conf. on Artificial Intelligence*, 1, pp.325-330, 1994.
- [7] Cheng, Y. Rule based traffic reactive simulation model, *Applied Artificial Intelligence*, **12**, pp.5-27, 1998.
- [8] Polat, F., & Shekhar, S., Distributed conflict resolution among co-operating expert systems, *Expert Systems*, **10**(4), pp.227-236, 1993.
- [9] D'Addio, G.F., Mazzucchelli, M. & Savio, S. Automatic conflict detection and resolution in metro-rail systems: evaluation approach of MARCO EU project, *Proc. 4th Int. Conf. Urban Transport*, Lisboa, 1998.