Real-time asynchronous conflict solving algorithms for computer aided train dispatching assistance systems

A. Kuckelberg & E. Wendler
Institute of Transport Science, RWTH Aachen University, Verkehrswissenschaftliches Institut, Aachen, Germany

Abstract

The computer aided real time dispatching assistance of train runs is a hard and complex problem. Although several approaches for dispatching and train surveillance exists – often limited to very specific aspects and situations of computer aided train operation – an integrated and flexible system covering train location, delay detection, computer aided (automated) conflict resolution and dispatching decision propagation is not available. The project DISKON is a development project initiated and assigned by DEUTSCHE BAHN AG that targets such a system. The proof of concept for this integrated approach has been made in 2007, when the system was tested several weeks at the Integrated Railway Laboratory (IEL) of the University of Dresden. In 2007 and 2008 the evaluation continued under real time conditions within operation control centres. Besides different modules and components for functionalities like train position detection, train run forecast, track assimilation and prognosis of arrival and departure times one scientific core of the DISKON system is a microscopic conflict solving component on blocking time level, enriched by another macroscopic component evaluating links of connection trains. The solving algorithm extends the ASDIS/L-system developed at the RWTH Aachen. It follows an asynchronous approach, considering conflicts chronologically and depending on priority values of the involved trains. The base algorithm was elementarily revised and enhanced by conflict situation detection and a derived conflict solving strategy builder. The result set of the strategy builder controls the parameter and the behaviour of the base algorithm. In this paper a rough architecture of dispatching systems is introduced, system requirement, environmental condition and behaviour are identified and a differentiated system evaluation is presented.

1 Introduction

In 2005 the DisKon project was launched, a development project initiated and assigned by Deutsche Bahn AG. The purpose of this project was to set up a computer aided dispatching and conflict solving prototype system for train operation under realistic conditions. This prototype system should demonstrate the ability of such assistance systems to perform the daily train operation in a more fluently and less resource consuming way, e.g. reducing energy usage due to reduced acceleration and braking phases (caused by conflicts) and increasing track usage capacities.

As academic project partners the Technical University of Aachen (RWTH Aachen) and the University of Dresden (TU Dresden) participated. Two approaches were coupled by the two project partners: a microscopic and a macroscopic approach for conflict solving and dispatching. The microscopic dispatching component implemented by RWTH Aachen is the core component of the DisKon system. It is designed to detect train positions, arise conflicts on blocking time level and establish dispatching plans in an asynchronous manner as well as passing modification information back to the real world. Moreover the component submits and retrieves data to and from the macroscopic dispatching component implemented by TU Dresden. The interaction between the microscopic and the macroscopic component is described in detail [2].

In any way the computer aided real time dispatching assistance of train runs is a hard and complex problem. The complexity is primarily implied by the real time conditions and the amount of data and information which has to be processed in a short time. Using the integrated model train laboratory of TU Dresden the project DisKon has shown in July 2007 that the chosen approach and concept are sufficient for real time dispatching assistance [1] under laboratory conditions.

The prototype showed, that a real time dispatching assistance system is possible which encapsulates different approaches and functionalities in a module based and flexible way.

The component which is focused in this paper is the microscopic dispatching and conflict solving component. It is based on the ASDIS/L-system developed at the RWTH Aachen [3]. ASDIS/L follows an asynchronous approach, considering conflicts chronologically and depending on the priority values of the involved trains.

The base algorithm was elementarily revised and enhanced by a conflict situation detector and a derived conflict solving strategy builder. The result determined by the strategy builder controls the parameter and the behaviour of the base asynchronous algorithm. This extension necessity is one of the research results of DisKon.

This paper presents the main microscopic DisKon components and their interaction and a rough functionality driven architecture in Section 2. The microscopic dispatching component ASDIS/L and the asynchronous approach implemented by this component is presented in Section 3 before the system is evaluated in Section 4. Section 5 finally concludes.

2 System architecture

A real time dispatching assistance system has to support several functionalities which are shown in Figure 1. This overview is independent from a concrete implementation and shows central components, data flow and data structures. Some aspects are deepened in this section.

2.1 Published timetable and basic data structures for dispatching assistance systems

The timetable is the central element for reliable, efficient and predictable train operation and the core data structure for dispatching and rescheduling systems: The system starts with the published timetable $tt_{orig} = dp_0$. The forecasted train runs are based on this timetable and are successively modified due to detected train positions and dispatching decisions. Each forecast modification transforms a dispatching plan dp_i into another plan dp_{i+1}. All operations are always applied to the last dispatching plan.

2.2 Train position detection

A correct and reliable detection of train positions is a requisite for dispatching systems. Different solutions are possible to enable discrete or continuous detection.

DISKON follows successfully the standardized UIC-telegrams [5] for discrete position detection containing time t, train tr and position p.

First the route r_{orig} of tr is derived from the current dispatching plan dp_i. If p is found in r_{orig} the train is adjusted in p at t. Otherwise a new route r_{new} is calculated which *could* be the current route of tr in which the train is adjusted. A succeeded adjustment transforms dp_i into dp_{i+1}.

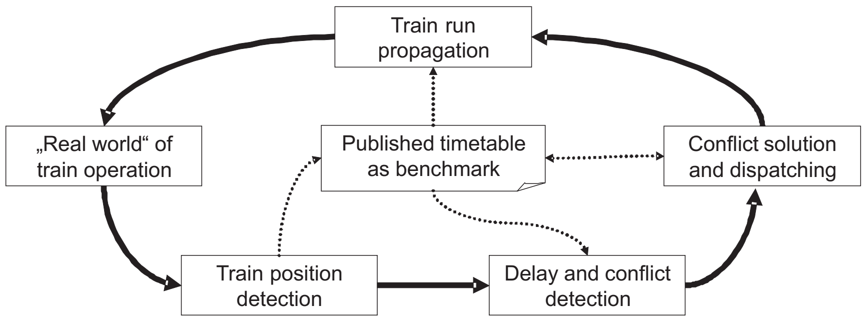
Train run propagation
„Real world“ of train operation
Published timetable as benchmark
Conflict solution and dispatching
Train position detection
Delay and conflict detection

Figure 1: Overview of train dispatching system components and data structures.

2.3 Delay and conflict detection

Within a dispatching plan arising disturbances (e.g. a high train density) and conflicts (e.g. overlapping blocking times) are detected. The detection considers reduction functions to simulate the uncertainty of future operation situations.

2.4 Rescheduling

If conflicts are detected within a dispatching plan dp_i, a new conflict free plan dp_{i+1} is produces by the conflict solving component.

The primary goal for a microscopic dispatching system is the elimination of blocking time overlaps. If equivalent solutions exist additional weighting criteria are applied, e.g. weighted sum of new delays, kept train links etc.

The dispatching system implemented within the DISKON project extends the ASDIS/L-algorithm, which is described in Section 3.

2.5 Dispatching plan propagation

Decisions made by the dispatching systems must be propagated and submitted to interlocking stations and/or train drivers. Depending on the technical, organisational and legal environment different integration levels are possible but not considered in this paper. The propagation process is a safety relevant task of such systems.

3 Asynchronous conflict solving

The conflict solving component of the DISKON project follows the asynchronous approach. Its key feature is the stepwise population of an initially empty dispatching plan by trains according to associated priority values.

For each step of the dispatching plan population the (overlapping time) conflicts are detected and chronologically solved. After an applied conflict solution the new (or remaining) conflicts of the dispatching plan are detected.

Solution options depend on the train priority value and on the current environmental conditions, e.g. if a train currently stops at a platform or if it is currently running without stopping before the conflict. These conditions usually limit the number of possible solutions the conflict solving component has to consider.

3.1 Synchronous vs. asynchronous, global vs. local

As mentioned before the asynchronous conflict solving algorithm is controlled by train priority values. Due to these priorities the dispatching plan is incrementally populated and available modifications are detected and heuristically evaluated for single trains. The system time defines a horizon, until which modifications are not possible any more and which defines the reduction for conflict detection.

In contrast to the priority driven asynchronous algorithm a synchronous approach manages all trains at the same time. Train movement is performed in parallel for running trains. Infrastructure is allocated exclusively in advance for trains. Whenever this allocation conflicts to another infrastructure allocation the train has to decrease its speed. The importance of trains is reflected e.g. by the length of infrastructure allocation ahead the train. One of the major disadvantages of the synchronous approach is, that priorities of trains are reflected in a quite unrealistic way and that synchronous conflict solving tends to deadlocks. Specific deadlock protection strategies must extend the core algorithm. On the other hand the original ASDIS/L-approach is deadlock free. With the extensions implemented for DisKon (e.g. time horizon, reduced dispatching abilities) this is not proven yet, but within tests no (conceptual) deadlocks were observed yet. A theoretical evaluation will be subject of [4].

Another aspect is discussed and studied: the optimality of results. While a lot of existing approaches claim to find *globally optimised* conflict solutions the asynchronous approach produces *locally optimised* solutions.

Global optimisation finds the best solution determining extreme values of a weighting function rating found solutions. On the other hand the ASDIS/L-approach uses a heuristic based algorithm (Figure 2) which can only be locally optimising due to the asynchronism itself:

- No ‘global’ view as long as trains with lower priority values are missing. Optimisation can only be optimal for the current (partially filled) dispatching plan.
- The solving algorithm for each phase considers (overlapping time) conflicts chronologically and pairwise with respect to trains.

Solutions found may turn out to be less optimal, when further trains are added. The usage of the asynchronous algorithm in several projects in the last years and

```
function makeConflictFree (dispatchingPlan dp) {
    dp.trains.deactivate;
    foreach prio ∈ dp.priorities.ascending {
        prio.trains.activate;
        while (dp.FirstConflict.Exists) dp.FirstConflict.Solve;
    }
    return dp;
}
```

Figure 2: The base algorithm of the microscopic conflict solving component: All trains of *dp* are deactivated before any conflict solving is performed. For each priority value two steps are performed: 1. All trains of the current priority value are activated and become visible for the conflict detection; and 2. The while-loop is performed for all trains up to the current priority level as long as conflicts exist (within the considered time horizon). The first detected conflict in time is solved.

```
function Conflict.DetectSituation {
    if ($this.oppositeDirection$) {return $crossing$}
    if ($this.occupation1.start \leq this.occupation2.start$) {
        return $train1before2$
    } else {return $train2before1$};
}
```

Figure 3: The conflict situation detection.

the analysis of measurement results has shown, that the algorithm is effective and discussion about superior optimisation approaches is a theoretical one. Moreover locally optimising approaches match real time (and real world) conditions in a better way than global ones do.

3.2 Conflict situation detection and strategy builder

Conflict situation detection (Figure 3) and strategy builder are major extensions of the old ASDIS/L-algorithm.

The situation detected is used by the strategy builder to select possible conflict solutions for both trains. Possible strategies are:

- searching alternative routes ahead, within or behind the area of the current conflict (alternative routes and acceptable delays are differently evaluated for the two trains);
- modifying stops and stopping times of the two trains within the associated overtaking blocks;
- decreasing or increasing train speed.

The possible strategies are related to the currently handled conflict. They are returned as a list which is processed by the single conflict solving method.

3.3 Single conflict solving

A rough overview of the single conflict solving is shown in Figure 4. It extends the original ASDIS/L-method [3].

The *searchForSolution*-Method can shortly be described: First the train with the higher priority value is tested to be modified within narrow modification limits. If no modification - including rerouting - of this train solves the conflict modifications of the second train are tested with larger limits. If no solution is found, limits are expanded. The *last* solution is the deceleration of the trains, sometimes down to a stop.

```
procedure Conflict.Solve ($OverlappingConflict\ OC$) {
    SET $situation$ := $OC.DetectSituation$;
    SET $OB$ := $OC.DetermineOvertakingBlocks$;
    SET $strategy$ := $OB.DetermineStrategy$;
    $OB.AdjustRoutes$;
    SET $sol$ := $\emptyset$;
    do {
      SET $sol$ $:\cup=$ $OC.searchForSolution$ $\cup$
                $OC.searchForAlternativeRoutes$ ($strategy$);
      if ($OC.samePriorities$) {
          SET $sol$ $:\cup=$ $OC.exchange.searchForSolution$ $\cup$
                  $OC.exchange.searchForAlternativeRoutes$ ($strategy$);
      }
      SET $terminate$ := $sol.Success$ or $sol.NoMoreSoutions$;
      if (not $terminate$) $OB.enlargeOvertakingBlocks$;
    } until ($terminate$);
}
```

Figure 4: The solution of an overlapping conflict (OC) adjusting a route means, that trains may be shifted to main tracks within the overtaking block (OB) if a stop can be removed; the Method $searchForSolution$ is the original method of ASDIS/L; $OC.exchange$ swaps the meaning of train 1 and 2 within OC and $a :\cup = b$ means $a := a \cup b$.

4 Evaluation

The evaluation of the presented approach to computer aided real time dispatching assistance systems must consider several aspects, which can only be discussed partially. The evaluation is experience driven based on DISKON test weeks at the TU Dresden and ongoing evaluation within operation control centres of the DB NETZ AG.

4.1 Effectiveness

The DISKON-system has proven its effectiveness within the test weeks in 2007 in the IEL of TU Dresden. It was shown that the components, data flows and interactions are sufficient for computer aided dispatching assistance systems. Evaluating the first experience drawn from the real world test showed, that the system and the asynchronous approach is sufficient for real applications, too. Beside several problems like data consistency and availability or technical and semantic surprises the first tests of the prototype are promising. Some recognitions made in the IEL are:

- Conflicts were detected reliable, even when assisting human dispatcher failed.

- The asynchronous algorithm terminated whenever a solution was possible, no (conceptual) deadlocks were produced.
- The heuristic of the locally optimising algorithm produces reasonable and 'globally acceptable' solutions.

4.2 Usability and system acceptance

For DISKON as a prototype system to prove the concept of underlying algorithms the usability is not such an important point. In principle the system could run without any user interface expect from the way, how the dispatching plan is propagated. But in practice the usability is a very important aspect of its acceptance.

For a better acceptance due to the visibility of results and the ability to interact the prototype was extended by e.g. blocking time graphs, manual event triggering ability etc.

First feedbacks of human dispatcher evaluating the system showed that after a certain period of exercise the system supports the daily work in a quite good way.

4.3 Efficiency and system performance

The performance of a computer aided dispatching assistance system depends on several aspects:

- Data structures, data size and system design: The performance of the dispatching plan modification functionalities is directly implied by the data size, e.g. size of infrastructure or length of train runs. Our tests moreover showed that data structure is crucial for the system performance as well as an optimised data flow and therefore the design of system components and their interfaces.
- Information density: Another important performance factor is the density of incoming data, e.g. train position information per second.
- Algorithm complexity and performance: The complexity of single components and the conflict solving component was evaluated. Results will be presented in [4]. Experiences showed that the interaction of the components as well as the execution of the component functionality is possible in real time for the evaluated area.

5 Conclusions and further work

Computer aided real time dispatching assistance is still a hard and complex problem. Within the DISKON project an integrating prototype was successfully developed. The prototype as well as the requirements which have to be covered by such systems were presented roughly in this paper.

The experience collected from laboratory test in 2007 and from ongoing system tests in the real word showed that a system as it was intended is possible and that the concepts chosen for the DISKON-systems are sufficient.

One focus of this paper was on the central conflict solving component. It follows the asynchronous algorithm introduced by ASDIS/L and extends it.

The last chapter gave a rather short evaluation of the system with respect to effectiveness, usability and performance.

Future work will extend the functionality of the conflict solving component to face advanced conflict solving techniques and more specific conditions of train operations in Germany. Additionally the optimisation of data structure and system performance will go on to enhance the system for larger regions.

References

[1] *DisKon - Rat aus dem Rechner*, bahntech 02/07, pp. 4–9, 2007, Deutsche Bahn AG, www.db.de/bahntech

[2] Kuckelberg, A. & Kurby, S. *Kopplung mikroskopischer Belegungsdisposition und makroskopischer Anschlussdisposition im Eisenbahnnetz.*, Proceedings of 21. Verkehrswissenschaftlichen Tage, Dresden, 2007

[3] Jacobs, J. *Reducing delays by means of computer-aided 'on-the-spot' rescheduling*, Computers in Railways (Comprail) IX, pp. 603–612, 2004

[4] Kuckelberg, A. *Mikroskopische Disposition spurgebundener Verkehrsmittel unter Echtzeitbedingungen*, Thesis work to be published, RWTH Aachen

[5] Pause, T. *Systeme für Disposition und Lokalisierung*, EI-Eisenbahningenieur, 05/08, pp. 21–24.