

# Künstliche Intelligenz für das Verkehrsmanagementsystem der Zukunft

## Artificial Intelligence for the traffic management system of the future

Irene Sturm | Gereon Vienken | Andreas Söhlke | Johann Hartleb | Rihab Gorsane | Khalil Gorsan Mestiri | Daniel Tapia Martinez | Vincent Coyette | Minh Tri Truong | Michael Küpper

Ein intelligentes Verkehrsmanagementsystem im zukünftigen digitalisierten Bahnsystem muss in der Lage sein, Störungen im Betrieb automatisch zu beheben. Der hier vorgestellte Ansatz kombiniert Methoden aus dem Operations Research (OR) und der Künstlichen Intelligenz (KI), um das darunterliegende mathematische Optimierungsproblem schnell genug und mit hoher Qualität zu lösen. Die Ergebnisse demonstrieren, dass mit dem gewählten Ansatz aus KI und OR die Qualität von reinen OR-Lösungen erreicht werden kann. Die deutlich kürzere Rechenzeit für größere Störungsszenarien macht den gewählten Ansatz vielversprechend für den Einsatz im automatisierten Bahnbetrieb.

### 1 Motivation

Die Sektorinitiative „Digitale Schiene Deutschland“ (DSD) nutzt das Potenzial digitaler Technologien, um die Kapazität, Qualität und Effizienz des hoch komplexen Bahnsystems zu steigern. Zu den Zielen gehören hoch- und vollautomatisiertes Fahren, eine zugorientierte Sicherungstechnik mit fahrzeuggebundener Lokalisierung für das Fahren im wandernden Raumabstand sowie weitgehend automatisierte Störfall- und Diagnosesysteme. Ein Kernstück des so digitalisierten Netz- und Fahrbetriebs ist ein intelligentes Verkehrsmanagementsystem (Capacity and Traffic Management System – CTMS) für die vollautomatische Planung und Steuerung des Bahnbetriebs. Das CTMS ist ein zentraler Baustein, um die Potenziale von automatisiertem Fahren und digitaler Leit- und Sicherungstechnik auszuschöpfen.

Das bei DSD entwickelte CTMS zielt darauf ab, die Kapazität des Schienennetzes der DB Netz optimal an dessen Nutzer zu vergeben. Das CTMS ist grundsätzlich als universelles Werkzeug zu verstehen, welches diese Aufgabe über den gesamten Planungshorizont des Eisenbahnsystems erledigt, von der langfristigen Fahr- und Bauplanung bis hin zur Disposition der Zugfahrten im Betrieb. Eine eingehende Charakteristik von Umfang und Aufgaben des CTMS wird in [1] gegeben. Aus technischer Sicht bedeutet dies, dass in einem CTMS mathematische Optimierungsprobleme mit ähnlicher Struktur mit unterschiedlichen Optimierungskriterien gelöst werden müssen. Je nach Anwendungsfall variieren die Randbedingungen, wie z. B. Anforderungen an Rechenzeit. Um diesen Anforderungen gerecht zu werden, werden im Rahmen von DSD verschiedene Lösungsansätze aus dem OR, der KI und Kombinationen von beiden entwickelt und in die laufende Entwicklung des evolutionären CTMS-Produktprototyps integriert.

Der erste konkrete Anwendungsfall für CTMS wird die Disposition in einem Eisenbahnknoten sein, also die (Echtzeit-)Verkehrssteuerung im Betrieb. Diese Teilfunktionalität des CTMS liegt momentan

An intelligent traffic management system in the digitalised railway system of the future must be capable of responding automatically to any disturbances in operations. The approach presented here combines methods from Operations Research (OR) and Artificial Intelligence (AI) to resolve the underlying mathematical optimisation problem with the appropriate response time and solution quality. The results demonstrate that the approach based on AI and OR produces solutions of comparable quality to those derived purely using OR methods. Significantly shorter computing times for larger scenarios indicate that the chosen approach shows promise for application in automated rail operations.

### 1 Motivation

The “Digitale Schiene Deutschland” (DSD) sector initiative leverages the potential of digital technologies to increase the capacity, quality and efficiency of the railway system. DSD’s goals include both highly and fully automated driving, a train-oriented safety logic with vehicle-based localisation to enable driving with moving headways, as well as widely automated incident management and diagnostics. An intelligent Capacity and Traffic Management System (CTMS) for the fully automated planning and control of rail operations constitutes a key component in such digitalised infrastructure and traffic. The CTMS is instrumental in exploiting the potential of highly and fully automated driving and digital safety technology. The CTMS developed at DSD initially aims to optimally allocate the DB Netz railway network’s capacity to its users. The CTMS can be understood as a universal tool serving the railway system’s entire planning process, from long-term timetabling and construction planning to rescheduling in operations. An in-depth description of the scope and functionality of the CTMS is provided in [1]. From a technical point of view, a CTMS needs to solve various mathematical optimisation problems of a similar structure, but with varying optimisation criteria. The framework conditions, such as computing time requirements, also vary according to the specific use case. Various approaches from OR, AI and combinations thereof are being developed at DSD and integrated into the development of the evolutionary CTMS product prototype in order to meet these requirements.

The first actual CTMS use case will involve rescheduling, i.e. (real-time) traffic control in operations in a medium-sized railway node. The current optimisation development is fo-

im Fokus der Optimierer-Entwicklung und wird im Folgenden näher erläutert. Daher wird für jeden Zeitpunkt im Betrieb bereits ein mikroskopischer Ablaufplan vorausgesetzt, der detaillierte Fahrprofile der geplanten Zugfahrten sowie alle für sie notwendigen Infrastrukturbefehle enthält. Im ersten Anwendungsfall werden Schnittstellen zu den vorhandenen (Um-)Systemen, insbes. LeiDis (Leitsystem zur Netzdisposition), implementiert. Aus den darüber bezogenen Fahrprofilen wird der initiale Ablaufplan konstruiert. Im hochautomatisierten Eisenbahnbetrieb soll es möglich sein, diesen Plan zu jedem Zeitpunkt kurzfristig an neue Gegebenheiten oder veränderte Anforderungen anzupassen. Dies erfordert ein System, das in kurzer Zeit einen neuen mikroskopischen Ablaufplan berechnen kann. Im vorgestellten Ansatz wird das Szenario der Infrastrukturstörung betrachtet: Hier muss im laufenden Betrieb ein aktueller Ablaufplan aufgrund eines plötzlich unbefahrbaren Gleisabschnitts Neuberechnet werden.

Das beschriebene Szenario wird in der Literatur als Timetable-Rescheduling (TTR)-Problem beschrieben. TTR ist ein kombinatorisches Optimierungsproblem, bei dem die Rechenzeit für eine optimale Lösung exponentiell mit der Größe der Problem Instanz wächst, etwa mit der Anzahl der Züge und der Größe des betrachteten Netzausschnittes. Bestehende Ansätze modellieren TTR als ganzzahliges Optimierungsproblem [2], als gemischt-ganzzahliges (lineares) Programmierungsproblem [3] oder als Alternativen Graphen [4]. Für die Lösung der so formulierten Optimierungsprobleme existieren verschiedene Ansätze wie mathematische Suchmethoden (z. B. Branch and Bound [4]), heuristische Ansätze (z. B. First-In-First-Out (FIFO) oder First-Come-First-Served) [5], Metaheuristiken [6] (z. B. Taboo Search oder Genetische Algorithmen), Verwendung von Solvern (CPLEX) [6], Spieltheorie [8] oder simulationsbasierte Ansätze [9].

Im Folgenden wird ein bei DSD entwickelter Ansatz vorgestellt, der eine Problemformulierung aus dem OR mit der Technik des Deep Reinforcement Learning (DRL) kombiniert. DRL ist eine Methode der KI, in der KI-Modelle durch Erfahrung lernen, komplexe Entscheidungsprobleme zu lösen.

Für Störungsszenarien wird zunächst eine Problem Instanz in Form eines Ereignisgraphen erstellt. Aufbau und Funktionsweise des Graphen werden im nächsten Abschnitt im Detail beschrieben. Ein DRL-Agent wird darauf trainiert, aus der großen Anzahl der möglichen Interventionen diejenige Sequenz von Interventionen auszuwählen, die den gestörten Ablaufplan so anpasst, dass die Verspätung im gesamten Netzabschnitt minimiert wird. Weitere Einzelheiten sowie eine detaillierte mathematische Problemformulierung befinden sich in [11].

## 2 Lösungsansatz

### 2.1 Ereignisgraphen

Räumlich-zeitliche Ereignisgraphen (Timed Event Graphs) sind ein oft gewählter Ansatz, um Optimierungsprobleme im Bereich der Fahrplankonstruktion und der Fahrplanadaptation zu formulieren. Im vorgestellten Ansatz wird der Ereignisgraph folgendermaßen aufgebaut: Ereignisse, die den Fahrweg des Zuges bestimmen, werden durch die Knoten des Graphen repräsentiert. Die Kanten zwischen den Knoten stellen Abhängigkeiten zwischen diesen Ereignissen dar. Verschiedene Knotentypen repräsentieren Ein- und Austritt in den Netzabschnitt, Ankunft und Abfahrt an einem Halt sowie Ankunft an Infrastrukturelementen wie z. B. Weichen (Tab. 1). Die Knoten enthalten Informationen über die Ereignisse, die sie repräsentieren, etwa den geplanten Zeitpunkt des Ereignisses, die Geschwindigkeit des Zuges und ggf. dessen Verspätung.

discussed on this specific CTMS application and will be explained here. It is therefore assumed that a microscopic schedule containing detailed journey profiles of the planned train runs and all the infrastructure commands required for them exists for any time in the operations. Interfaces to the prevalent (surrounding) systems, especially LeiDis, are being implemented for the initial application of CTMS. The mission profiles pulled from these systems will serve as the basis for the construction of the initial schedule. The schedule in a highly automated railway system must be able to be quickly adapted to any changing circumstances or requirements at any time. This requires a system that can quickly calculate a microscopic schedule. The approach presented here addresses the use case of infrastructure disruptions: an existing schedule has to be re-calculated during operations after a track section becomes unavailable.

The literature refers to this scenario as a Timetable Rescheduling (TTR) problem. A TTR is a combinatorial optimisation problem where the computation time for finding an optimal solution increases exponentially with the size of the problem instance. The size of the problem instance is determined by the number of trains in the schedule segment and the size of the infrastructure section under consideration. Existing approaches model a TTR as an integer optimisation problem [2], a mixed-integer (linear) programming problem [3] or an alternative graph [4]. Various approaches exist for resolving the optimisation problems formulated in this way, such as mathematical search methods (e.g. Branch and Bound [4]), heuristic approaches (e.g. first-in-first-out (FIFO) or first-come-first-served) [5], metaheuristics [6] (e.g. Taboo Search or genetic algorithms), the use of solvers (CPLEX) [6], game theory [8] or simulation-based approaches [9].

The following describes an approach developed at DSD that combines problem formulation from OR with the Deep Reinforcement Learning (DRL) technique. DRL is an AI method, in which AI models learn to solve complex decision-making problems through experience. In the case of disruption scenarios, a problem instance is first created in the form of an event graph. The structure and functionality of the graph are described in detail in the next section. A DRL agent is trained to evaluate all the possible interventions and to select the one sequence of interventions that adapts the disrupted schedule in a way that minimises any resulting delays in the entire network. Further details, as well as a detailed mathematical problem formulation for this approach, can be found in [11].

## 2 Method

### 2.1 Event graphs

Timed event graphs are well-established for formalising optimisation problems in the fields of timetable construction and adaptation. The event graph in the presented approach is structured as follows: the graph nodes represent the events from a train run. The edges between the nodes represent the dependencies between these events. The different node types represent the entry and exit into/from a network section, arrival and departure at a commercial stop and arrival at infrastructure elements such as points (tab. 1). Each node contains information about the event it represents, such as its location, its scheduled time, the train speed and any delay, if applicable. The dependencies or constraints between the node pairs are represented by the edges of the event graph. The first type of

Kante	Abhängigkeit	Wert
Fahrt	Fahrzeit zwischen zwei Punkten	Minimale Fahrzeit
Halt	Vorgeschriebene Haltezeit	Minimale Haltezeit
Abstand Weiche	Zugfolgezeit zwischen Folgezügen an Weichen	Minimaler Zeitabstand
Abstand Halt	Zugfolgezeit zwischen Folgezügen an Haltestellen	Minimaler Zeitabstand

Tab. 1: Abhängigkeiten im Ereignisgraphen

Abhängigkeiten oder Einschränkungen zwischen Knotenpaaren werden durch die Kanten des Ereignisgraphen repräsentiert. Kanten können zum einen zeitliche Abhängigkeiten repräsentieren, etwa die minimale Zeit, die ein Zug gemäß der erlaubten Höchstgeschwindigkeit des Streckenabschnitts und seiner Beschleunigungs-/Bremscharakteristik benötigt, um die Strecke zwischen zwei Ereignissen zurückzulegen. Zum anderen gibt es Kanten, die (geschwindigkeitsabhängige) einzuhaltende Zugfolgezeiten repräsentieren, etwa von zwei Zügen an einer Weiche oder an einem Halt. Diese Formulierung bildet die sogenannte Moving-Block-Sicherungslogik ab, die Teil des Zielbilds der DSD ist. Der Ereignisgraph kann in eine Form gebracht werden, die einem Weg-Zeit-Diagramm des betrachteten Ablaufplans ähnelt, das mit zusätzlichen Informationen angereichert ist.

Bild 1 zeigt links den betrachteten Netzausschnitt sowie den geplanten Fahrweg zweier Züge. Die rechte Grafik zeigt die zugehörigen Zeit-Weg-Linien, wobei auf der y-Achse Elemente der Infrastruktur verzeichnet sind (Weichen und ein Haltepunkt, engl. Stopping Location, SL) und auf der x-Achse die Zeit. Die roten und grünen Linien stellen den Fahrweg der Züge über die Zeit dar. Die blauen Kanten beschreiben Mindestzugfolgezeiten an Infrastrukturelementen (Same-Junction Edge und Same-Stopping-Location Edge), die von beiden Zügen befahren werden.

Edge	Dependence	Value
Run	Travel time between two points	Minimum travel time
Stop	Required dwell time at a stop	Minimum dwell time
Headway at point	Time interval between consecutive trains at points	Minimum time interval
Headway at stop	Time interval between consecutive trains at stops	Minimum time interval

Tab. 1: The dependencies in the event graph

edge represents temporal dependencies, such as the train's minimum travel time between the locations of two events, while considering the maximum permitted speed on the track section and the train's acceleration / deceleration characteristics. The second type of edge represents the (speed-dependent) headways between consecutive trains, for example at a point or a stop. This formulation reflects the so-called moving-block safety logic, which constitutes part of DSD's objective.

The event graph can be transformed into a format similar to a distance-time diagram of the schedule enhanced with additional information.

The left-hand part of fig. 1 shows the routes of two trains passing infrastructure elements on the network. The right-hand part shows the respective time-distance lines. The infrastructure elements are depicted on the y-axis (points and a stopping location, SL); the time is represented on the x-axis. The red and green lines represent the train paths over time. The blue edges describe the minimum headways at the infrastructure elements (same-junction edge and same-stopping-location edge) used by both trains.

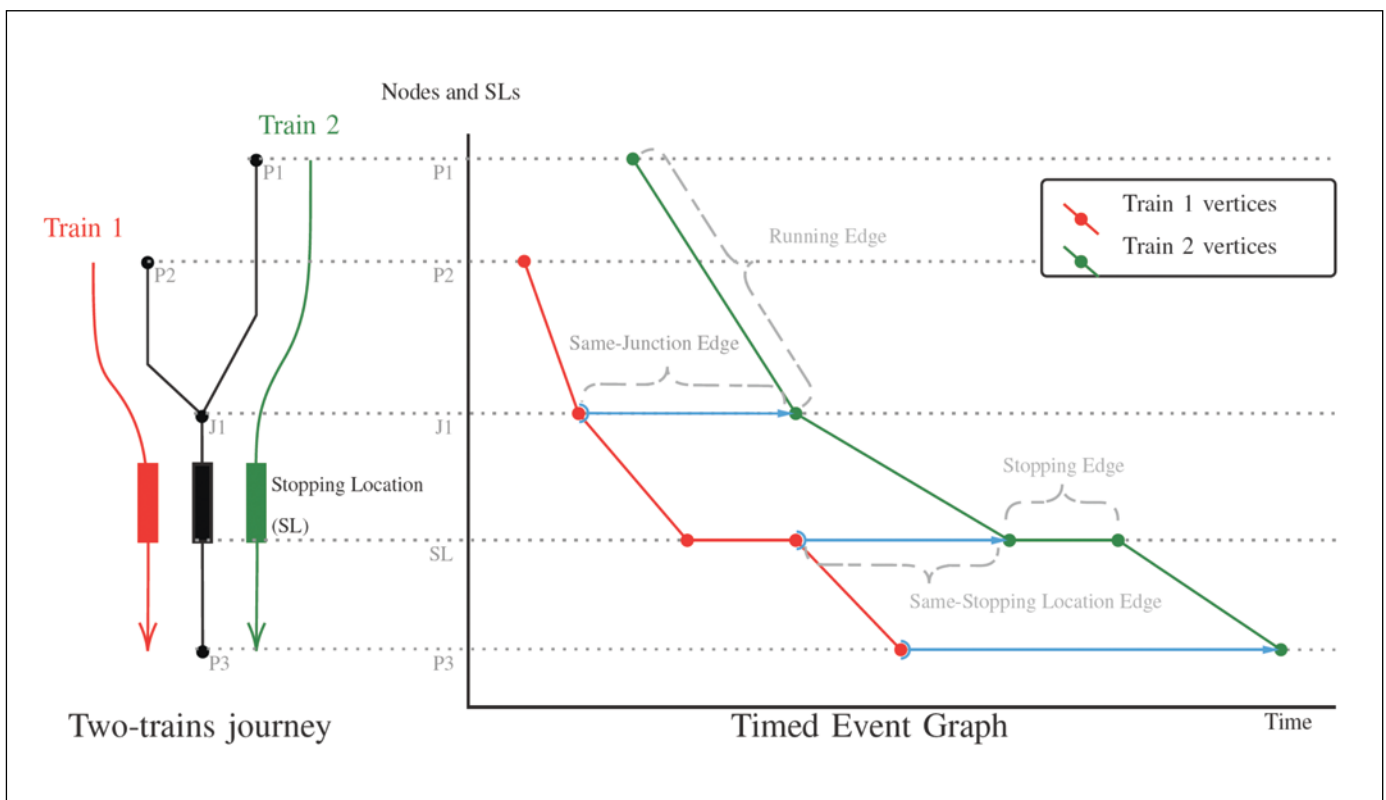


Bild 1: Ereignisgraph und Weg-Zeit-Diagramm

Fig. 1: An event graph and distance-time diagram

Quelle / Source: Digitale Schiene Deutschland / DB Netz

## 2.2 Optimierungskriterien

Im betrachteten TTR existiert ein gestörter Fahrplan, der wieder fahrbar gemacht und optimiert werden muss. Die dabei angestrebte Qualität des Fahrplans kann unterschiedliche Dimensionen berücksichtigen:

- Diverse Verspätungsmaße in verschiedener Granularität (Verspätung eines Zuges, einer Gruppe oder aller Züge, aggregierte Verspätung an einer Betriebsstelle etc.)
- auf Reiseketten bezogene Maße (Anzahl erreichter/verpasster Umsteigeverbindungen, Reisendenpünktlichkeit etc.)
- Umsatz, Energieverbrauch, Verschleiß und andere relevante Größen.

Diese Qualitätsmaße können gewichtet und miteinander kombiniert werden. Im Prinzip kann der vorgestellte Ansatz für verschiedenste Optimierungskriterien angepasst werden. Für die hier vorgestellte erste Implementierung wurde eine einzige Dimension gewählt: eine einfache Definition von Gesamtverspätung. Für jeden Halt und auch den Austritt aus dem Netz gibt es im existierenden Ablaufplan Zeitfenster. Die einzelne Verspätung ergibt sich aus der Diskrepanz von aktueller Ankunft am Halt bzw. Netzaustritt zu dem im initialen Plan definierten Zeitfenster. Diese Werte werden aufsummiert über alle Halte und Züge.

## 2.3 Anwendung des KI-Modells

Das Zusammenspiel zwischen der Erstellung des Ereignisgraphen und dem im Folgenden vorgestellten DRL-Agenten im hier betrachteten Störungsszenario ist wie folgt: Ausgangspunkt ist ein Szenario, bestehend aus gewünschten Zugfahrten. Jede angefragte Zugfahrt beinhaltet Zugtyp, Gleis und Zeitfenster für Ein- und Austritt in/aus dem Netzausschnitt sowie ggf. eine Sequenz von Passagierhalten mit Gleis- und Zeitfensterangaben. Im vorgestellten TTR-Szenario existiert ein initialer Ablaufplan, der die angefragten Zugfahrten auf dem Netzausschnitt realisiert. Im Fall einer auftretenden Störung wird ein Ereignisgraph erstellt, der den aktuellen Plan vom Zeitpunkt der Störung an in der veränderten Situation darstellt.

Dieser Ereignisgraph wird zunächst auf Validität geprüft. Ein valider Ereignisgraph hat einen intakten Fahrweg für jeden Zug und wahrt alle Mindestfahrzeiten. Er enthält keine Inkonsistenzen wie z. B. physikalisch unmögliche Überholreihenfolgen oder sich blockierende wechselseitige Abhängigkeiten von Zugreihenfolgen. Operationen auf dem Ereignisgraphen, wie Einstellung der Geschwindigkeitsprofile, Berechnen alternativer Routen oder Festlegen von Zugreihenfolgen, machen in einem ersten Schritt einen nicht-validen Ereignisgraphen valide. Dieser valide, aber typischerweise nicht optimale Graph wird dem KI-Modell zur Optimierung übergeben.

Das KI-Modell entscheidet über eine Sequenz von Interventionen auf Kanten des Ereignisgraphen, um die Summe der Verspätungen im Netzabschnitt zu minimieren. Mögliche Interventionen sind Umrouten von Zügen und Reihenfolgeänderung an Infrastrukturelementen. Die gewählten Interventionen werden auf dem Ereignisgraphen ausgeführt. Mittels der in [10] beschriebenen Formulierung als gemischt-ganzzahliges Optimierungsproblem (Mixed Integer Programming) werden alle nötigen Änderungen an Folgeereignissen im Ereignisgraphen realisiert und alle Geschwindigkeitsprofile angepasst. Falls die vom KI-Modell gewählten Interventionen zu invaliden Lösungen führen, muss die letzte Intervention rückgängig gemacht oder die gesamte Lösung verworfen werden.

## 2.4 Reinforcement Learning Agent

Zwar garantieren Methoden des OR optimale Lösungen von TTR auf Ereignisgraphen, doch bei großen Probleminstanzen und folglich großen Suchräumen sind die benötigten Rechenzeiten deutlich zu lang. Obwohl dieses Problem in der Literatur durch eine Vielzahl verfeinerter Suchstrategien und -methoden adressiert wurde, ist den

## 2.2 The optimisation criteria

The given TTR problem requires a disrupted schedule to be returned to feasibility and then optimised. The solution can be optimised according to a range of quality criteria:

- various delay measures at different levels of granularity (the delay of a single train, a group of trains or all the trains, the aggregated delay at a station etc.)
- measures related to passenger trips (the number of connections achieved/missed, passenger-centric punctuality measures etc.)
- revenue, energy consumption, wear and tear and other relevant variables

These quality measures can be combined with and weighted against one another. In principle, the presented approach can accommodate a wide variety of optimisation criteria. Only one dimension has been chosen for the initial implementation presented here; a simple definition of global delay. The schedule includes time windows for each passenger stop as well as for exiting the network. A discrepancy between the actual arrival and the appropriate time window defined in the initial plan defines an individual delay at a stopping or exit location. These values are added up over all the stops and train runs.

## 2.3 Applying the AI model

The following describes the interaction between the creation of the event graph and the application of the DLR agent when resolving the disruption scenario: the starting point involves a scenario consisting of a set of desired train runs. Each train run request specifies a train type, the locations and the time windows for the entry and exit to/from the network and, if applicable, a sequence of passenger stops with time and platform details. The presented TTR problem includes an initial schedule that regulates the requested train runs on a network section. In the case of a disruption, an event graph is created to represent the current schedule in the newly changed situation, starting at the moment of the disruption.

This event graph is initially checked for validity. A valid event graph has a feasible path for each train run and respects all the minimum travel times. It must not contain any inconsistencies, such as physically impossible overtaking sequences or cyclic dependencies of the events. Operations on the event graph, such as setting the speed profiles, calculating alternative routes or re-ordering trains, transform a non-valid event graph into a valid one in this step. The thus validated, but typically not optimal, event graph is then submitted to the AI model for optimisation.

The AI model decides on a sequence of interventions at the edges of the event graph in order to minimise the sum of the delays in the network section. Possible interventions include re-routing trains and changing train orders at infrastructure elements. The selected sequence of interventions is executed on the event graph. The mixed-integer optimisation formulation described in [10] means that all the necessary changes to the subsequent events in the event graph are performed and all the speed profiles are adjusted. The last intervention is reversed or the entire solution is discarded if the AI model's interventions lead to any invalid solutions.

## 2.4 The Reinforcement Learning Agent

In principle, OR methods guarantee mathematically optimal solutions to any TTR on event graphs. However, the required computing times are prohibitively long for large problem in-

Autorinnen und Autoren zum Zeitpunkt des Verfassens dieses Artikels keine Methodik bekannt, deren Rechenzeit nicht mindestens polynomial mit der Menge der Züge oder Größe der Infrastruktur wächst. Der vorliegende DRL-Ansatz entscheidet sich schnell genug für eine Lösung, um die Anforderung von automatisierter Disposition zu erfüllen, verzichtet jedoch dafür ggf. auf das mathematische Optimum einer Funktion. Für den Echtzeitbetrieb ist es wichtig, schnell auf Störungen zu reagieren.

Das vorgestellte KI-Modell besteht aus mehreren Komponenten: ein Graph Neural Network (GNN), also ein neuronales Netzwerk für die Verarbeitung von Graphstrukturen, das den Ereignisgraphen verarbeitet und eine Sequenz an Interaktionen auswählt, die die globale Verspätung minimiert. Eine weitere Komponente bewertet und optimiert den Lernvorgang. Eine genaue Beschreibung der Architektur des KI-Modells und der Trainingsalgorithmen findet sich in [11].

**2.5 Evaluation**

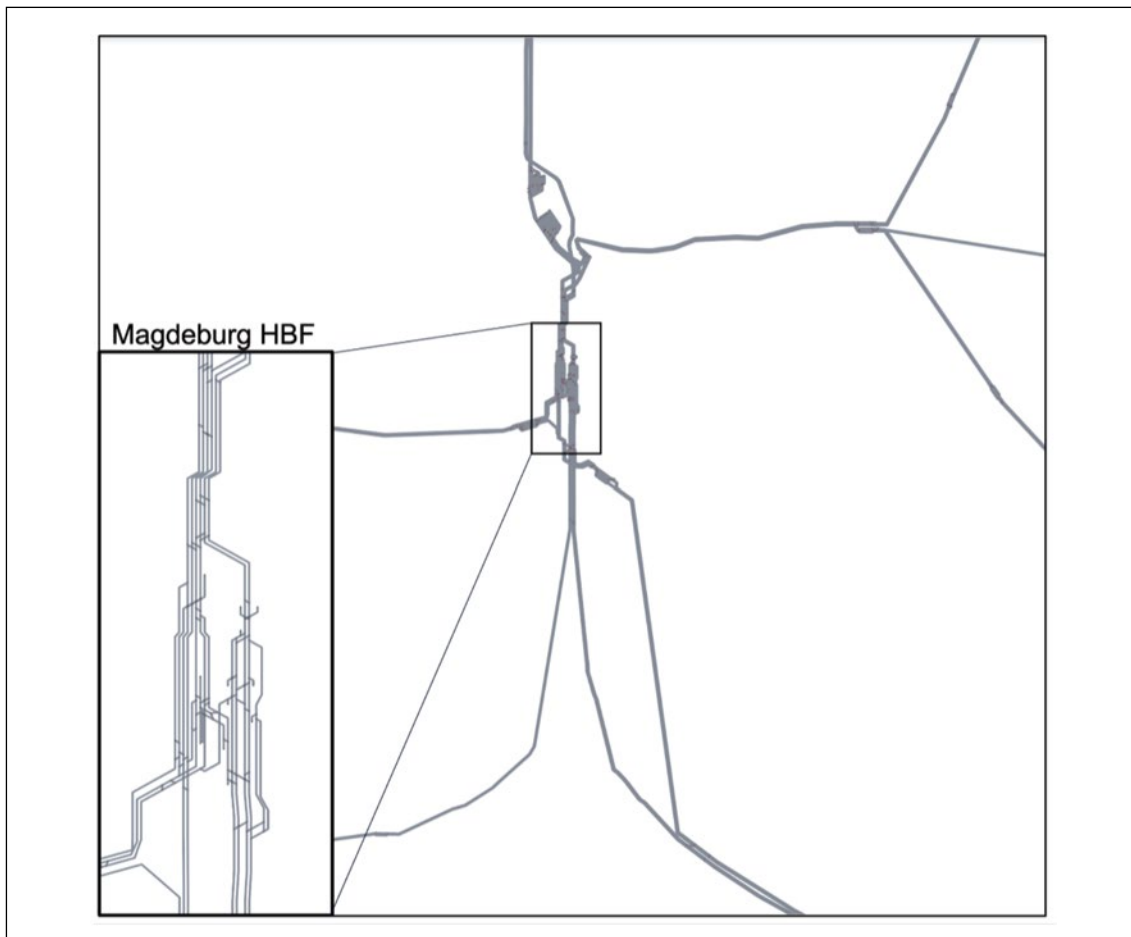
Der vorgestellte Ansatz wird auf einem realen Ausschnitt des deutschen Schienennetzes um den Hauptbahnhof Magdeburg evaluiert (Bild 2). Für diesen Netzausschnitt wurden sieben Testszenerien generiert, die jeweils zwei Stunden Betriebszeit umfassen, in denen zwischen 20 und 35 Züge den Netzabschnitt befahren. Dabei wurde bewusst eine Verkehrsdichte erzeugt, die insbesondere in den Bahnhöfen mit der realen Verkehrsdichte im heutigen Betrieb vergleichbar ist. Wie oben beschrieben, wurde für jedes Szenario zunächst ein initialer Ablaufplan berechnet. In jedem initialen Ablaufplan wurden ausgewählte, befahrene Gleisabschnitte gestört, also unbefahrbar gemacht. Für die jeweilige Störungssituation wurde dann ein vali-

stances with consequently large search spaces. Even though this problem has been addressed by a variety of refined search strategies and methods in the literature, at the time of writing this article the authors are unaware of any approach where the computation time does not grow at least polynomially with the number of trains or the infrastructure size. The present DRL approach meets the requirement for a rapid response to any disruptions in rail operations, but may only approximate a mathematically optimal solution. It is essential to react to any disruptions in live operations quickly.

The AI model consists of different components, such as a Graph Neural Network (GNN), i.e. a neural network for processing graph structures that processes the event graph in order to learn to select a sequence of interactions that minimises the global delay. A further component evaluates and optimises the learning process. A detailed description of the AI model's architecture and training algorithms can be found in [11].

**2.5 Evaluation**

The DRL approach has been evaluated on a section of the German rail network centred on the Magdeburg Central Station (fig. 2). Seven test scenarios have been generated for this network section, each comprising two hours of operating time during which 20 to 35 trains pass through the network. The traffic density has been config.d to be representative of the actual traffic density in today's operations, in particular at the stations. As described above, an initial schedule was constructed for each scenario. Disruptions were generated in each scenario by selecting and blocking the utilised tracks. A valid



**Bild 2: Netzausschnitt Magdeburg mit 29 Betriebsstellen und 343 Gleiskilometern**  
 Fig. 2: The Magdeburg network section with 29 operating points and 343 kilometres of track Quelle / Source: Digitale Schiene Deutschland/ DB Netz

der Ereignisgraph erzeugt und dem trainierten KI-Modell zur Optimierung übergeben.

Die vom KI-Modell erstellten Pläne werden mit zwei Referenzlösungen verglichen: Die erste Referenzlösung benutzt das Branch-and-Cut (B&C)-Modell aus [10]. Die maximale Zeit für das Berechnen dieser Lösung wurde auf eine Stunde begrenzt. Die Ergebnisse wurden mit einem Gurobi-Solver auf einer Cloud-Instanz mit 16 CPU berechnet. Die zweite Referenzlösung benutzt eine First-In-First-Out (FIFO)-Heuristik, die verwendet wird, um zu entscheiden, welcher Zug ein gemeinsam genutztes Infrastrukturelement zuerst überfährt.

### 2.6 Training

Der vorgestellte Ansatz wird in zwei Schritten evaluiert: Im ersten Schritt werden KI-Modelle jeweils auf demselben Szenario trainiert und evaluiert (Modell A). In diesem Schritt hat das KI-Modell sozusagen die Möglichkeit, die Lösung eines speziellen Problems „auswendig zu lernen“. Im zweiten Schritt wird ein KI-Modell (Modell B) auf einer Reihe von verschiedenen Szenarien (auf demselben Netzausschnitt) trainiert und muss dann im Evaluationsschritt Szenarien lösen, die es nicht im Training gesehen hat. Diese Vorgehensweise prüft die Generalisierungsfähigkeit des KI-Modells, also die Fähigkeit, auf unbekannte Situationen richtig zu reagieren. Schließlich soll das System im späteren Betrieb eine möglichst große Bandbreite von Problemen lösen, die naturgemäß nicht von vornherein bekannt sind. Die Trainingszeit für die Modelle betrug zwischen 43 und 73 Stunden auf einer Cloud-Instanz mit elf CPU, einer NVIDIA T4 GPU und maximal 27 GB Speicher.

### 3 Ergebnisse

Die Ergebnisse in Tab. 2 zeigen, dass Modell A in sechs von sieben Testszenarien Lösungen generiert, die sowohl das B&C-Modell als auch die FIFO-Heuristik an Qualität deutlich übertreffen. Dieses Ergebnis zeigt, dass der gewählte DRL-Ansatz prinzipiell das Potenzial hat, diese Probleme in hoher Qualität zu lösen.

In der Praxis müssen KI-Modelle ungesehene Probleminstanzen lösen, auf denen sie nicht trainiert wurden. Diese Fähigkeit wurde im KI-Modell B erfolgreich geprüft. Die Ergebnisse von Modell B sind in den meisten Testszenarien nicht besser als B&C, in den meisten Fällen aber in einer ähnlichen Größenordnung. In allen Fällen ist das KI-Modell besser als die FIFO-Heuristik.

Bild 3 zeigt die für die Lösung benötigte Rechenzeit für Szenarien mit 20, 25, 30 und 35 Zügen auf einer logarithmischen Skala. Hier wird deutlich, dass die Rechenzeit des KI-Modells relativ konstant bei ca. 10-20 Sekunden liegt und auch bei den größten Probleminstanzen von 35 Zügen nicht wesentlich größer ist als bei kleineren Problemen. B&C benötigt bei allen Zuganzahlen mehr Zeit für die Berechnung als die KI. Bei 35 Zügen schöpft B&C die gegebene Maximalzeit von einer Stunde voll aus und benötigt damit bereits

Szenario	Gestörter Plan vor Optimierung	FIFO	B&C	DRL Modell B	DRL Modell A
1 (20 Züge)	954	721	127	61±047	2±000
2 (20 Züge)	639	1685	24	140±041	4±000
3 (25 Züge)	792	481	85	45±023	11±015
4 (25 Züge)	1560	1482	109	1037±375	10±006
5 (30 Züge)	692	541	43	619±070	130±139
6 (35 Züge)	1493	3459	88	998±077	25±004
7 (35 Züge)	1349	4174	379	427±086	96±152

Tab. 2: Durchschnittliche Verspätung pro Zug in Sekunden

event graph was generated for each such disrupted scenario and submitted to the trained AI model for optimisation.

The schedules produced by the AI model have been compared with two reference solutions: the first reference solution was created using the branch-and-cut (B&C) model from [10], where the processing time to calculate the solution was capped at one hour. The results were calculated using a Gurobi solver on a cloud instance with 16 CPUs. The second reference solution applied a first-in-first-out heuristic (FIFO) used to decide on the train order at any shared infrastructure elements.

### 2.6 Training

The presented approach is evaluated in two stages: in the first stage, the DRL models are trained and evaluated on the same scenario (Model A). Here, the DRL model has the opportunity to “memorise” the solution to a specific problem. In the second stage, a DRL model (Model B) is trained on a number of different scenarios (on the same network section), so that it has to solve previously unseen scenarios in the evaluation step. This approach tests the DRL model’s generalisation capability, i.e. its ability to resolve unknown situations. After all, the system will be required to solve a large variety of problems in real operations that naturally were unknown to it beforehand. The training time for the models ranged from 43 to 73 hours on a cloud instance with eleven CPUs, one NVIDIA T4 GPU and a maximum of 27 GB of memory.

### 3 Results

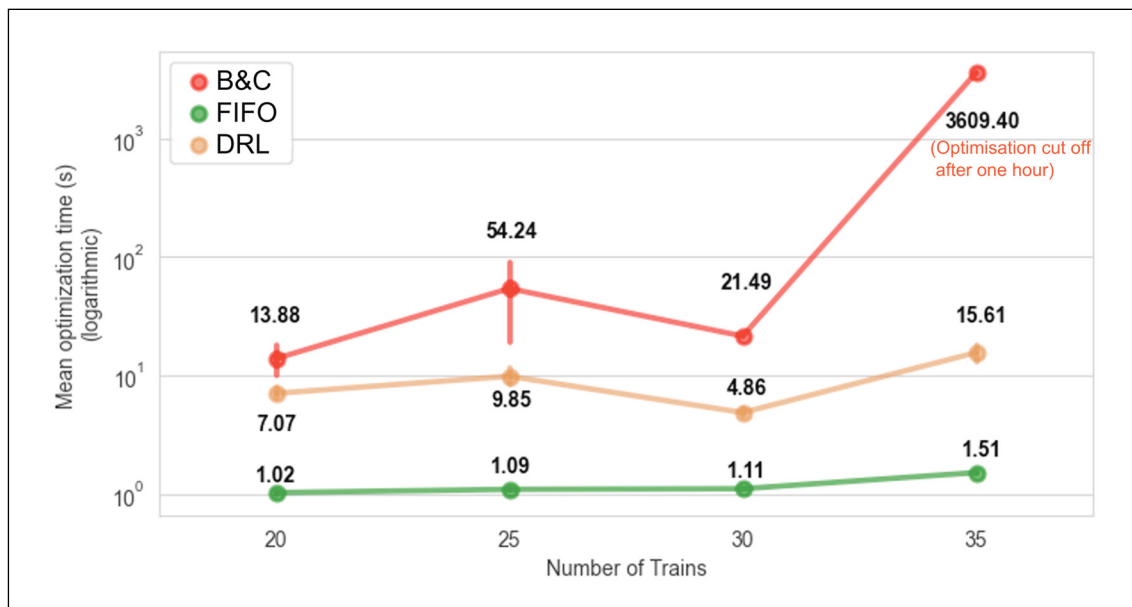
The results in tab. 2 show that DRL Model A generated solutions that significantly outperformed both the B&C model and the FIFO heuristics in terms of quality in six out of seven test scenarios. This shows that the chosen DRL approach has the potential to produce high quality solutions to such problems.

In practice, the DRL models have to solve unseen problem instances that they have not faced in training. This capability has been successfully demonstrated in DRL Model B. The results of DRL Model B have not outperformed those of B&C in most test scenarios, but they have reached a similar level of quality in the majority of test instances. The DRL model has outperformed the FIFO heuristics in all cases.

Fig. 3 shows the computation time required to solve scenarios with 20, 25, 30 and 35 trains on a logarithmic scale. The DRL model’s processing times are relatively constant in a range of about 10-20 seconds. The computation time for the largest problem instances with 35 trains is not significantly greater than for the smaller problems. B&C takes considerably more time to produce solutions than the DRL agents for all instances. B&C makes full use of the maxi-

Scenario	Disrupted plan before optimisation	FIFO	B&C	DRL Modell B	DRL Modell A
1 (20 trains)	954	721	127	61±047	2±000
2 (20 trains)	639	1685	24	140±041	4±000
3 (25 trains)	792	481	85	45±023	11±015
4 (25 trains)	1560	1482	109	1037±375	10±006
5 (30 trains)	692	541	43	619±070	130±139
6 (35 trains)	1493	3459	88	998±077	25±004
7 (35 trains)	1349	4174	379	427±086	96±152

Tab. 2: The average delay per train in seconds



**Bild 3: Rechenzeit: Die Rechenzeit des B&C-Modell wurde auf eine Stunde begrenzt.**

Fig. 3: Computation time: The computation time for the B&C model was limited to one hour. Quelle / Source: Digitale Schiene Deutschland / DB Netz

ein Vielfaches der Rechenzeit des DRL-Agenten bei nur moderatem Qualitätsvorteil.

#### 4 Schlussfolgerungen und Ausblick

Die Ergebnisse zeigen, dass der vorgestellte Ansatz des DRL-Agenten mit GNN auf der Event Graph-Problemformulierung Lösungen erzeugt, die vergleichbar in Qualität mit bewährten Methoden des OR sind. Der DRL-Ansatz erweist sich als deutlich besser skalierbar auf Probleminstanzen mit mehr Zügen. Insbesondere für den Anwendungsfall in einem automatisierten Verkehrsmanagementsystem, das flexibel und rapide auf Störungen reagieren muss, ist diese Eigenschaft ein kritischer Faktor.

Auf dem Weg zum bahnbetrieblichen Einsatz in einem produktreifen CTMS muss dieser Ansatz in verschiedenen Teilen weiterentwickelt werden: Erweiterte Möglichkeiten bei der Formulierung von Verkehrswünschen und Randbedingungen, zusätzliche Dispositionsmaßnahmen (z.B. Ausfall oder Umplanen von Verkehrshalten inklusive Abstimmung verkehrlicher Maßnahmen mit den Verkehrsunternehmen) sowie weitere Optimierungskriterien und verfeinerte Qualitätsmetriken. Neue Methoden aus dem sich schnell entwickelnden Feld der KI sollen eingebracht werden, um die Generalisierungsfähigkeit und die Skalierbarkeit des Systems sowie die Qualität der Lösungen weiter zu steigern. ■

imum allowed time of one hour for problem instances involving 35 trains and thus requires a multiple of the processing time required by the DRL agents with only a moderate increase in quality.

#### 4 Conclusions and outlook

The results show that the presented approach using a DRL agent with GNN for event graph problem formulation produces solutions that are comparable in quality to those produced using established OR methods. The DRL approach has proven to be much more scalable to problem instances with a higher number of trains. This property is essential for its use in automated real-time traffic control, where flexible and rapid reactions to disruptions are crucial.

Several aspects of this approach need to be further developed in order to prepare for application in a CTMS product in real operations: the traffic demand, constraints and business rules need to be modelled in a more detailed and realistic manner, the range of dispatching measures needs to be extended (e.g. stop cancellation, including the coordination of such measures with the railway undertakings) and more refined optimisation criteria and quality metrics need to be introduced. New methods from the rapidly developing field of AI should be explored in order to further increase the system's ability to generalise and its scalability, as well as to enhance the quality of its solutions. ■

#### LITERATUR | LITERATURE

- [1] Küpper, M.: „Das Capacity & Traffic Management System für die Digitale Schiene“, EI 10/2023, S. 60-64
- [2] Schachtebeck, M.; Schöbel, A.: „IP-based techniques for delay management with priority decisions“, in 8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (AT-MOS'08). Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2008
- [3] Acuna-Agost, R.; Michelon, P.; Feillet, D.; Gueye, S.: „Sapi: Statistical analysis of propagation of incidents. a new approach for rescheduling trains after disruptions“, European Journal of Operational Research, vol. 215, no. 1, S. 227–243, 2011
- [4] D'Ariano, A.; Pacciarelli, D.; Pranzo, M.: „A branch and bound algorithm for scheduling trains in a railway network“, European journal of operational research, vol. 183, no. 2, S. 643–657, 2007

#### AUTOREN | AUTHORS

**Dr. Irene Sturm**  
Product Owner und Expertin Künstliche Intelligenz /  
Product Owner and Expert Artificial Intelligence  
DB Netz AG  
Adresse / Address: Stresemannstraße 123a, D-10963 Berlin  
E-Mail: irene.sturm@deutschebahn.com

**Gereon Vienken**  
Experte für Künstliche Intelligenz / Expert Artificial Intelligence  
DB Netz AG  
Adresse / Address: Stresemannstraße 123a, D-10963 Berlin  
E-Mail: gereon.vienken@deutschebahn.com

- [5] Palgunadi, S.; Supraba, D.; Harjito, B.: „Job-shop scheduling model for optimization of the double track railway scheduling:(case study: Solo-yogyakarta railway network),“ in 2016 International Conference on Information & Communication Technology and Systems (ICTS). IEEE, 2016, S. 90–95
- [6] Corman, F.; D’Ariano, A.; Pacciarelli, D.; Pranzo, M.: „A tabu search algorithm for rerouting trains during rail operations,“ Transportation Research Part B: Methodological, vol. 44, no. 1, S. 175–192, 2010
- [7] Zhou, M.; Liu, X.; Hou, Z.; Shang, J.; Yue, Y.; Song, H.: „Integrated optimization of dispatching decision and speed trajectory for high-speed railway under disturbances,“ in 2020 Chinese Automation Congress (CAC). IEEE, 2020, S. 4656–4661
- [8] Norio, T.; Yoshiaki, T.; Noriyuki, T.; Chikara, H.; Kunimitsu, M.: „Train rescheduling algorithm which minimizes passengers’ dissatisfaction,“ in Innovations in Applied Artificial Intelligence: 18th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA/AIE 2005, Bari, Italy, June 22–24, 2005. Proceedings 18. Springer, 2005, S. 829–838
- [9] Lei, N.: „Study on the strategy of train operation adjustment on high speed railway,“ Journal of the China Railway Society, 2001
- [10] Borndörfer, R.; Denißen, J.; Heller, S.; Klug, T.; Küpper, M.; Lindner, N.; Reuther, M.; Schlechte, T.; Söhlke, A.; Steadman, W.: „Microscopic timetable optimization for a moving block system,“ 2021
- [11] Gorsane, R.; Gorsan Mestiria, K.; Tapia Martinez, D.; Coyette, V.; Makhlof, M. B.; Vienken, G.; Truong, M. T.; Söhlke, A.; Hartleb, J.; Kerkeni, A.; Sturm, I.; Küpper, M.: „Reinforcement learning based train rescheduling on event graphs“, The IEEE Intelligent Transportation Systems Society Conference Management System, accepted, 2023

#### Andreas Söhlke

Product Owner und Experte Operations Research /  
Product Owner and Expert Operations Research  
DB Netz AG  
Adresse / Address: Stresemannstraße 123a, D-10963 Berlin  
E-Mail: andreas.soehlke@deutschebahn.com

#### Johann Hartleb

Product Owner und Experte Operations Research /  
Product Owner and Expert Operations Research  
DB Netz AG  
Adresse / Address: Stresemannstraße 123a, D-10963 Berlin  
E-Mail: johann.hartleb@deutschebahn.com

#### Rihab Gorsane

Forschungsingenieur und Teamleitung / Research Engineer and Team Lead  
InstaDeep  
Adresse / Address: ICC3 building, block D, 4th floor. North Urban Center,  
TN-Tunis 1082  
E-Mail: r.gorsane@instadeep.com

#### Khalil Gorsan Mestiri

Forschungsingenieur / Research Engineer  
DB Netz AG  
Adresse / Address: Stresemannstraße 123a, D-10963 Berlin  
E-Mail: k.gorsan@instadeep.com

#### Daniel Tapia Martinez

Forschungsingenieur / Research Engineer and Product Owner  
InstaDeep  
Adresse / Address: 40 Rue du Faubourg Poissonnière, F-75010 Paris  
E-Mail: d.martinez@instadeep.com

#### Vincent Coyette

Forschungsingenieur / Research Engineer  
InstaDeep  
Adresse / Address: 40 Rue du Faubourg Poissonnière, F-75010 Paris  
E-Mail: v.coyette@instadeep.com

#### Minh Tri Truong

Forschungsingenieur / Research Engineer  
InstaDeep  
Adresse / Address: 40 Rue du Faubourg Poissonnière, F-75010 Paris  
E-Mail: m.truong@instadeep.com

#### Dr. Michael Küpper

Capacity & Traffic Management System Ambassador /  
Capacity & Traffic Management System Ambassador  
DB Netz AG  
Adresse / Address: Stresemannstraße 123a, D-10963 Berlin  
E-Mail: michael.m.kuepper@deutschebahn.com

**Zukunft ist Digitalisierung**

**SAFERAIL**  
www.safetrail.de

**Ab 01. März 2024  
mit weiterem Standort in Karlsruhe**