Business Paper

Automatische Erzeugung schematischer Übersichtspläne aus Bahn-Geodaten

Johannes Elstner[1], Julius Bullinger[2], Julian Trujillo Lopez[3], Dr. Volkmar Bachmann[4]

[1,2] *SIGNON Deutschland GmbH*

[3] *Technischer Projektleiter Projekt D3iP*

[4] *Gesamtprojektleiter Projekt D3iP*

1 Einleitung

Schematische Darstellungen der Eisenbahninfrastruktur dienen dazu, funktionale Zusammenhänge sichtbar zu machen, die in einer lagerichtigen Darstellung nicht oder nur schwer visualisierbar wären. Dabei werden unter Vernachlässigung der Lagerichtigkeit die Fahrbeziehungen auf der Infrastruktur in eine abstrahierte Form überführt, in der diese besser erkennbar sind.

Traditionell werden diese Darstellungen von Hand gezeichnet und in einem iterativen Prozess weiterentwickelt, oft ohne klare formelle Regeln. Dieses Paper stellt eine Lösung vor, mit der schematische Übersichtspläne computergestützt und automatisch aus Bahn-Geodaten erzeugt werden können.

Für schematische Darstellungen im Allgemeinen haben sich die folgenden zwei grundsätzlichen Sichtweisen herausgebildet, die jeweils konkreten Anwendungsgebieten dienen (vgl. Gély 2010):

1.1 Netzgrafiken (makroskopische Sichtweise)

Hier werden einzelne Gleise gemäß ihrer verkehrlichen Bedeutung zu Strecken zusammengefasst. Dargestellt werden alle Strecken in einem relevanten Bereich, um die Fahrbeziehungen zwischen den Knotenbahnhöfen zu verdeutlichen. Bekanntestes Beispiel sind die klassischen Liniennetzpläne im Nahverkehr; darüber hinaus gibt es solche Grafiken in verschiedenen Formen auch für innerbetriebliche Zwecke zur Planung und Disposition von Zugfahrten.

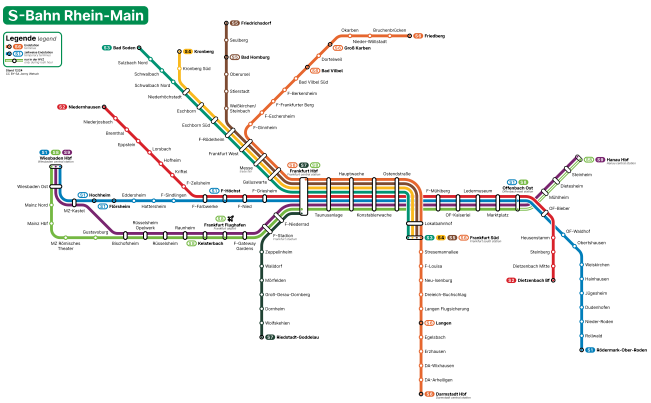

Abb. 1 Liniennetzplan der S-Bahn Rhein-Main, Fahrplanjahr 2025
Quelle: [2]

[1] Korrespondierender Autor: johannes.elstner@deutschebahn.com

1.2 Schematische Übersichtspläne, „Spurpläne“ (mikroskopische Sichtweise)

Die mikroskopische Sichtweise betrachtet topologische Knoten (Weichen und Gleisenden) und Kanten (Gleisabschnitte von Weiche zu Weiche bzw. Gleisende) als kleinste topologische Einheit, auf der Zugfahrten stattfinden, und stellt diese dar. Während Netzgrafiken sowohl in x- als auch in y-Richtung die geografischen Verhältnisse in vereinfachter Form wiedergeben, werden bei Spurplänen alle Gleise entlang einer willkürlich definierten *Hauptachse* horizontal im Plan abgewickelt, wobei sie ausschließlich als waagerechte oder diagonale Linien, in Ausnahmefällen auch als senkrechte Linien dargestellt werden.

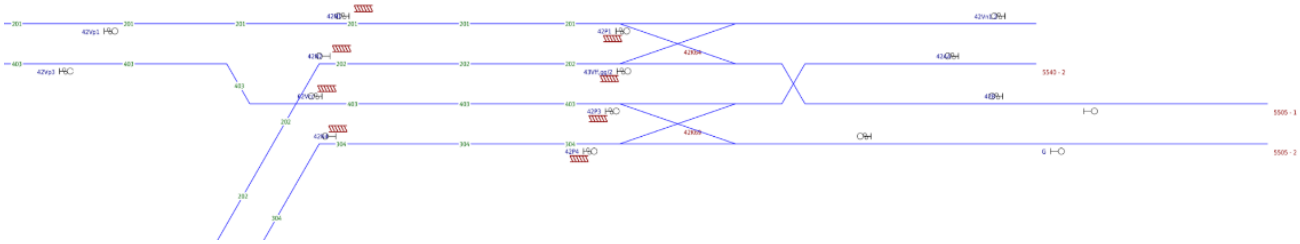

Abb. 2 Spurplan (Ausschnitt) der Betriebsstelle München-Donnerbergerbrücke
Quelle: [3]

Schematische Übersichtspläne sind nicht genordet. Bei der DB InfraGO definiert sich die Ausrichtung im Regelfall von links nach rechts entlang der aufsteigenden Kilometrierungsrichtung der maßgebenden Strecke [4].

2 Bestehende Ansätze

Die händische Erstellung von schematischen Übersichtsplänen ist aufwändig und erfordert viel Erfahrung, um die oft komplexen Zusammenhänge verständlich und effektiv darzustellen.

Zeitgemäßer ist der Ansatz, solche Pläne computerunterstützt zu berechnen. Für die makroskopische Sichtweise (Liniennetzpläne) gibt es dazu seit Jahrzehnten Forschung mit zahlreichen vorgeschlagenen Lösungen für unterschiedliche Ziele, z.B. (Wolff 2007), (Nöllenburg 2014), (Wu 2019), (Wu 2020).

Die mikroskopische Sichtweise (Spurpläne) ist bisher in der Forschung deutlich seltener betrachtet worden. Doch auch hier gibt es vielversprechende Ansätze. Eine Variante ist der Einsatz von Optimierungsmethoden wie z.B. ganzzahlige lineare Optimierung („integer programming“) (Nöllenburg 2011) oder die Betrachtung als Erfüllbarkeitsproblem („satisfiability problem“, SAT) (Luteberget 2021). Alternativ können die topologischen Daten als Graphenproblem betrachten werden, bei dem gerichtete oder ungerichtete zyklische Graphen durch prozedurale Heuristik (z.B. Brands 2016) oder einem „force-directed Layout“-Algorithmus (vgl. Fruchterman und Reingold 1991) ausgerichtet werden.

Auch gibt es verschiedene Softwarelösungen und Bibliotheken zur Optimierung und Visualisierung solcher Graphen, wie Apache TinkerPop[2], NetworkX[3], Graphviz[4] oder Visualisierungsbibliotheken wie d3.js[5] und Cytoscape[6]. Diese sind leistungsfähig bei der Modellierung von Netzwerken, stoßen jedoch bei der spezifischen

[2] https://tinkerpop.apache.org

[3] https://networkx.org

[4] https://graphviz.org

[5] https://d3js.org

[6] https://cytoscape.org

Berechnung von schematischen Übersichtsplänen an ihre Grenzen. Vor allem fehlen ihnen spezialisierte Algorithmen und Optimierungsmöglichkeiten, um die topologischen und geometrischen Anforderungen von Eisenbahninfrastrukturen effizient zu berücksichtigen und übersichtlich darzustellen.

Die hier vorgestellte Methode orientiert sich im Kern an der Algorithmik von (Brands 2016), erweitert diese aber an einigen wesentlichen Stellen. Andererseits können durch die Wahl eines anderen Datenmodells für die Eingangsdaten einige Schritte vereinfacht oder ausgelassen werden.

3 Projektkontext D3iP – Motivation und Zielsetzung

Das Projekt *D3iP* (Durchgängig digitale Datenhaltung im LST-Planungsprozess)[7] hat eine umfangreiche Modernisierung der Arbeitsweise in LST-Planungsprojekten zum Ziel: Der primäre Informationsträger an den Übergabeschnittstellen zwischen verschiedenen Planungsbeteiligten (Auftraggeber, Planer, Planprüfer usw.) ist bislang grundsätzlich Papier oder „digitales“ Papier (PDF-Dokumente, CAD-Zeichnungen). Diese Planungsdokumente sind menschen-, aber (inhaltlich) nicht maschinenlesbar, sodass eine maschinelle Durchführung von Planungs- oder Prüfaufgaben eine Überführung in ein geeignetes Format erfordert. Mit Blick auf einen singulären Arbeitsschritt übersteigt dieser Aufwand regelmäßig den Nutzen. Eine höhere Automatisierung gelingt nur, wenn der Datenaufbereitungsaufwand möglichst einmalig anfällt und allen nachfolgenden Prozessen zugutekommt.

D3iP steht daher für die Abkehr von (digitalem) Papier als primärem Informationsträger und die Hinwendung zu maschinenlesbar en Datenformaten, um die erforderliche Durchgängigkeit schaffen und Automatisierungspotentiale zu heben (vgl. Bachmann und Trinckauf 2024). Zentral ist hierfür die Einführung des PlanPro-Datenmodells, das zwischen allen Beteiligten und allen Leistungsphasen im LST-Planungsprozess den gesamten Umfang einer technischen Planung abbilden soll. Daneben werden aber auch gezielt Werkzeuge entwickelt und Prozesse überarbeitet, die für die Umstellung auf ein effizientes durchgängig digitales Arbeiten unabdingbar sind.

Schematische Übersichtspläne sind im von D3iP betrachteten Planungsprozess mindestens an folgenden Stellen erforderlich:

- Als Soll-Spurplan in der Betrieblichen Aufgabenstellung (BAst) zu Projektbeginn
- Als Sicherungstechnischer Übersichtsplan und ETCS-Übersichtsplan in der Stellwerks- und ETCS-Planung
- Als ZN-Übersichtsplan in der Planung der betrieblichen Leittechnik[8]

Für die Erzeugung der Pläne in der Stellwerks- und ETCS-Planung existieren teilautomatische Lösungen in den am Markt verfügbaren LST-Planungswerkzeugen, die stark auf den Anwendungsfall zugeschnitten sind.

Für die Aufgabenstellung und die Planung der Leittechnik entwickelt D3iP eigene Anwendungen, die *Digital Normierte Aufgabenstellung* (DNA) und den *PlanPro-ZN/ZL-Planer*. Um hierbei die Ziele von Automatisierung und durchgängig digitaler Datenhaltung zu erreichen, ist eine automatische Generierung der schematischen Übersichtspläne erforderlich.

Speziell im Fall der BAst liegen aufgrund der frühen Projektphase noch keine qualitätsgesicherten PlanPro-Daten vor. Der Soll-Spurplan, für den diese später erstellt werden, wird im Rahmen der BAst erst entworfen. Daher ist es notwendig, den Spurplan direkt auf Basis der Bestandsdaten zu erzeugen, die die DB InfraGO in

[7] https://digitale-schiene-deutschland.de/de/projekte/D3iP

[8] Zur Leittechnik gehören: ZN – Zugnummernmeldeanlage, ZL – Zuglenkung, ZLV-Bus - Zuglaufverfolgung

Form von Geometrie- und Attributinformationen im Geoinformationssystem *AVANI* vorhält. Über ID-Beziehungen ist eine Überführbarkeit der in der BAst formulierten Infrastrukturanforderungen in die später aus derselben Quelle erzeugten PlanPro-Daten gewährleistet.

Die D3iP-Anwendung *SOfA* (Schema- und Objektaufbereitung für die Aufgabenstellung) hat daher zum Ziel, schematische Übersichtspläne automatisch aus AVANI-Daten zu erzeugen. Außerdem soll sie die Bearbeitungsumgebung vom Ist- zum Sollspurplan bereitstellen. Die in SOfA verarbeiteten Daten eignen sich auch zur Befüllung von Tabellen und Listen innerhalb der BAst.

Der im Rahmen von SOfA entwickelte Algorithmus zur Erzeugung schematischer Übersichtspläne soll auch im PlanPro-ZN/ZL-Planer zum Einsatz kommen, dort aber auf Basis der dann vorliegenden PlanPro-Daten.

Grundsätzlich ist das Ziel des hier vorgestellten Ansatzes, für die meisten Betriebsstellen mindestens eine 80%-Lösung erzeugen zu können, mit der Möglichkeit der anschließenden manuellen Optimierung.

4 Algorithmus zur Erzeugung des Gleisschemas

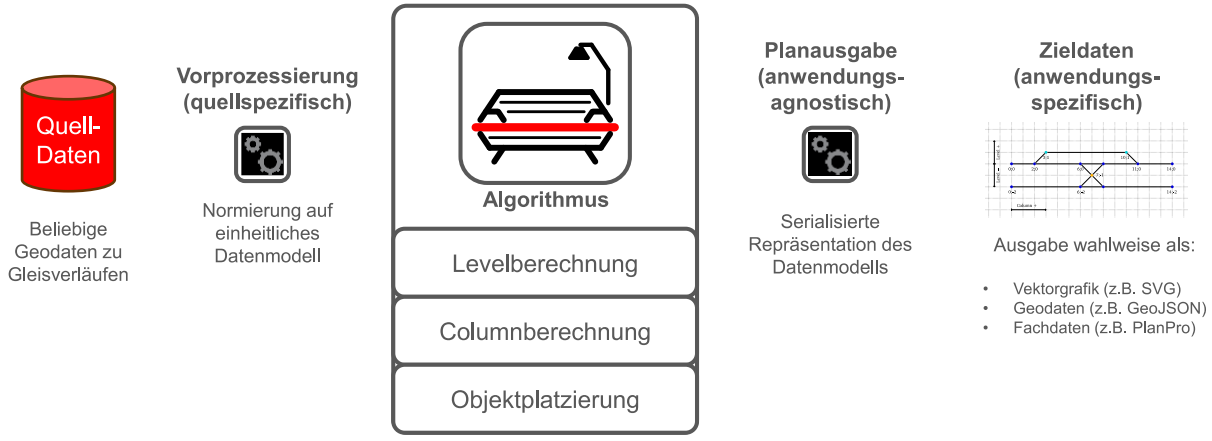

Abb. 3 Ablaufschema der Anwendung *SOfA*
(Eigene Darstellung)

4.1 Festlegungen

Die Abgrenzung des Algorithmus von vor- und nachbereitenden Schritten ist in Abb. 3 dargestellt.

Um eine einheitliche Darstellung der erzeugten Übersichtspläne sicherzustellen, wurden einige Festlegungen und Darstellungsprinzipien definiert. Die Modellierung orientiert sich dabei am PlanPro-Modell: Gleisverzweigungen und -enden werden als Knoten modelliert, die Gleise dazwischen als Kanten.

Um eine grundlegend horizontale Ausrichtung zu gewährleisten, werden topologische Kanten vorrangig waagerecht dargestellt, mit abzweigenden Kanten im 45°-Winkel. Die Längenverhältnisse der Kanten ergeben sich zunächst nach betrieblichen und später auch aus optischen Gesichtspunkten. So muss zum Beispiel Platz für die Anzeige von Beschriftungen und Infrastrukturobjekten vorgesehen werden.

Die Winkelweite von 45° für abzweigende Kanten wurde gewählt, da sie sich direkt aus der entwickelten Modellierung ergibt (siehe folgendes Kapitel). Je nach gestalterischen Anforderungen können Abzweigungen alternativ z.B. mit 60°-Winkel abgebildet werden.

Für die erste Umsetzung wurden Gleisschleifen und -dreiecke explizit ausgeklammert, u.A. da diese für eine korrekte und verständliche Darstellung Kanten mit vertikalen Anteilen voraussetzen.

Um die in Kap. 3 erwähnten Optionen für die Eingangsdaten (AVANI / PlanPro) bedienen zu können, wurde für den eigentlichen Algorithmus ein normalisiertes Datenmodell entwickelt. Kennzeichen dieses Modells ist, dass alle für den Algorithmus notwendigen Informationen in Form von topologischen Knoten und Kanten sowie zugehörigen Attributen vorliegen. Auf die Datenquelle zugeschnittene Pre-Processing-Routinen überführen die Eingangsdaten in dieses normalisierte Modell. Hierzu gehören:

- Sicherstellung eindeutiger ID-Beziehungen zwischen topologischen Knoten und Kanten
- Erkennen oder Definieren von Abzweigrichtungen der Weichenschenkel, sodass Fahrbeziehungen und kreuzende Gleise daraus ablesbar sind (geometrisch oder attributiv)
- Einheitliche Modellierung aller Stellen, an denen sich Gleise überschneiden (siehe Kap. 4.5.4)

Um den Fokus auf die Entwicklung der Algorithmik legen zu können, wurde die freie Geoinformationssystemsoftware QGIS als Entwicklungs- und Laufzeitumgebung genutzt. Diese bietet bereits die nötige Funktionalität, Geodaten in verschiedenen Formaten einzulesen, eine Kartendarstellung, Bearbeitungsmöglichkeiten etc. Die Entwicklung erfolgte anhand von einfachen Topologien wie einem simplen Überholbahnhof (Limburg Süd) bis zu mittelkomplexen Betriebsstellen wie zum Beispiel Jena-Göschwitz. Komplexere Betriebsstellen können auf weniger komplexe Topologien aufgeteilt und dann einzeln gelöst werden.

4.2 Ziel-Datenmodell

Der Algorithmus verfolgt das Ziel, sämtlichen topologischen Knoten (*TOP-Knoten*) der gegebenen Infrastruktur kartesische Koordinaten zuzuordnen, sodass die dazwischenliegenden topologischen Kanten (*TOP-Kanten*) ausschließlich als waagerechte oder diagonale Linien gezeichnet werden können.

TOP-Kanten, die in der Zeichnung anteilig diagonal und waagerecht abgebildet werden sollen, erhalten vom Algorithmus zusätzliche *Gerüstknoten*, die ebenfalls mit einem Koordinatenpaar versehen werden (türkisfarbene Knoten in Abb. 4).

Wie in Abb. 4 dargestellt, heißen die y-Koordinaten während der Berechnung *Level*, die x-Koordinaten *Column*. Sie sind zunächst ganzzahlig definiert. Für die spätere Plandarstellung können sie in beliebige Einheiten des Planrasters (z.B. 1 Level = 10px) umgerechnet werden.

Der Level 0 liegt direkt auf der bereits in Kap. 1.2 erwähnten *Hauptachse* des Bahnhofs (z.B. durchgehendes Hauptgleis km-aufwärts der bedeutendsten Strecke), sodass Knoten oberhalb der Hauptachse positive Level erhalten und Knoten unterhalb der Hauptachse negative. Die Column 0 liegt am Beginn der Hauptachse, sodass der Spurplan km-aufwärts von links nach rechts gezeichnet wird und im Regelfall ausschließlich positive Column-Werte vorkommen.

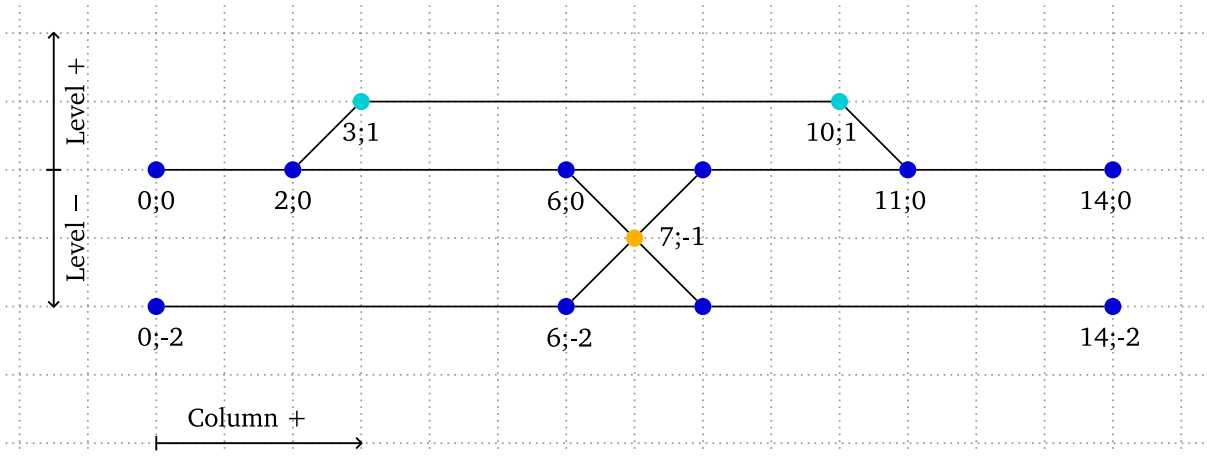

Abb. 4 Koordinatensystem. Top-Knoten blau, Gerüstknoten türkis, Doppelknoten orange dargestellt. (Eigene Darstellung)

4.3 Level-Berechnung

Zunächst wird die vertikale Schichtung (Level) der Gleisinfrastruktur ermittelt, aus der später die y-Koordinaten des Plans gewonnen werden. Hierzu werden ausgehend von der Hauptachse alle TOP-Kanten durchlaufen und jedem TOP-Knoten wird ein Level zugeordnet. Bei Bedarf werden zusätzlich Gerüstknoten eingefügt und mit Leveln versehen.

An spitz befahrenen Weichen wird zuerst der Strang besucht, der zurück in Richtung Hauptachse führt, d.h. links der Hauptachse der rechte Strang und umgekehrt. So wird gewährleistet, dass die vergebenen Level die tatsächliche aufsteigende Schichtung der TOP-Kanten aus Sicht der Hauptachse wiedergeben.

Stumpf befahrene Weichen werden nur in Richtung der Spitze durchlaufen. Um alle von der Hauptachse erreichbaren Kanten zu besuchen, erfolgt die gesamte Traversierung des Graphen erst links, dann rechts der Hauptachse in Richtung km-aufwärts, dann links und rechts der Hauptachse in der Gegenrichtung. Berechnet eine spätere Traversierung einen höheren Level für ein und denselben Knoten, wird der eingetragene Wert überschrieben.

Aus graphentheoretischer Sicht handelt es sich weder um eine reine Tiefensuche noch um eine reine Breitensuche, sondern um einen steten Wechsel zwischen Tiefen- und Breitensuche in bestimmten Teilgraphen. Nichtsdestoweniger ist der Vorgang deterministisch.

Die ersten Schritte der Levelberechnung auf einem fiktiven Beispiel-Bahnhof km-aufwärts links der Hauptachse sind in Abb. 5 dargestellt und erläutert (folgende Seite).

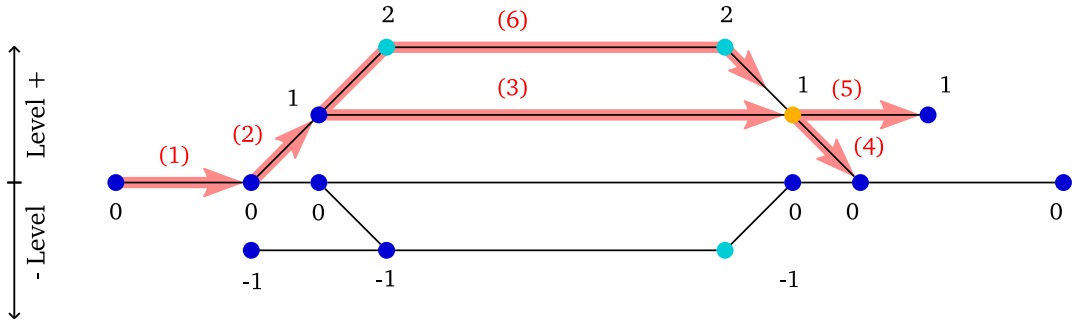

Abb. 5 (1) Ablaufen der Hauptachse (Level 0)
(2) Abzweigen links aus der Hauptachse → nächster TOP-Knoten erhält Level 1
(3) Rechten Strang ablaufen, nächster TOP-Knoten wurde noch nicht besucht → erhält Level 1
(4) Rechten Strang ablaufen, nächster TOP-Knoten ist Teil der Hauptachse → behält Level 0
(5) Linken Strang ablaufen, Gleisende → erhält Level 1
(6) Linken Strang ablaufen, nächster Knoten wurde bereits besucht → behält Level 1. Linker und Rechter Strang ab letztem TOP-Knoten führen zu gleichem TOP-Knoten → 2 Gerüstknoten auf Level 2 einfügen (hellblau)
(Eigene Darstellung)

4.4 Column-Berechnung

Die horizontale Ausdehnung des Gleisgraphen im Plan wird in Form von *Columns* (TOP-Knoten, im Plan auf derselben horizontalen Position untereinander anzuordnen sind) abgebildet.

Die ersten Schritte der Columnberechnung auf dem fiktiven Beispielbahnhof sind in Abb. 6 dargestellt und erläutert.

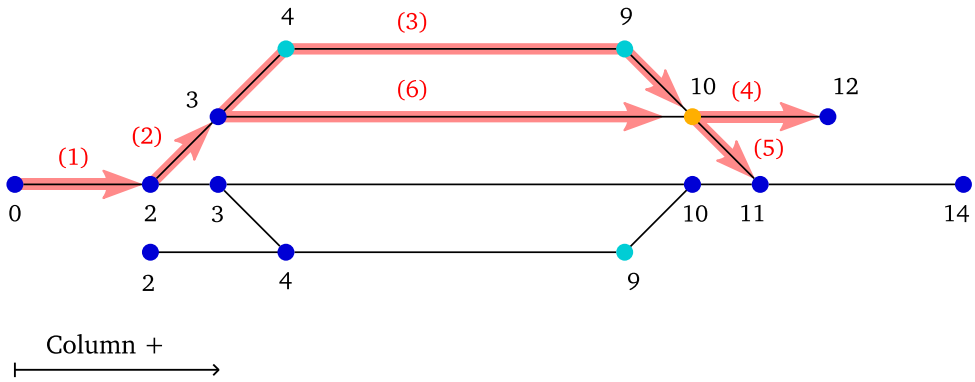

Abb. 6 (1) Ablaufen der Hauptachse ab Column 0 → nächster TOP-Knoten erhält wegen Kantenlänge Column 2
(2) Abzweigen aus der Hauptachse → nächster TOP-Knoten erhält Column 3
(3) Linken Strang ablaufen → erster Gerüstknoten erhält Column 4.
Waagerechter Anteil der Kante wird wegen Kantenlänge gestreckt → nächster Gerüstknoten erhält Column 9 und nächster TOP-Knoten Column 10
(4) Linken Strang ablaufen → Gleisende erhält aufgrund der Gleislänge Column 12
(5) Rechten Strang ablaufen → Top-Knoten auf Hauptachse erhält Column 11. Äußerster durchgehender Pfad für den betrachteten Bereich (2-3-4-9-10-11) gefunden.
(6) Rechten Strang ablaufen → nächster TOP-Knoten behält Column 10.
(Eigene Darstellung)

Ähnlich wie bei der Level-Berechnung wird der Graph vollständig abgelaufen und der Column-Wert hochgezählt. Anders als bei der Levelberechnung wird hierbei ausgehend von der Hauptachse an jedem TOP-Knoten zuerst der Strang besucht, der weiter von der Hauptachse wegführt, d.h. von der Hauptachse nach links ausgehend bevorzugt der linke Strang und erst danach der rechte. Der Algorithmus besucht daher ausgehend

von der Hauptachse zunächst die nach außen weisenden Stumpfgleise, dann findet er einen *äußersten durchgehenden Pfad*, der zur Hauptachse zurückführt und anschließend die zwischen dem äußersten durchgehenden Pfad und der Hauptachse gelegenen Gleise.

Der äußerste durchgehende Pfad ist auf vielen gängigen Weichenanordnungen („Weichenharfe“) der Pfad mit den meisten Zwischenstationen zwischen zwei Punkten auf der Hauptachse; damit ist er eine geeignete Heuristik zur schnellen Ermittlung der erforderlichen Längsausdehnung. Wenn hingegen andere Pfade mehr Zwischenknoten enthalten als der äußerste durchgehende Pfad, sind diese maßgeblich für die Längsausdehnung. Diese Fälle offenbaren sich, wenn die Traversierung auf einen TOP-Knoten mit bereits vergebenem Column-Wert stößt und aufgrund des aktuell begangenen Pfades einen höheren Column-Wert erfordert (siehe Kap. 4.5.2).

Die Traversierung zur Columnberechnung erfolgt nur vorwärts entlang der Hauptachse (nach links und rechts). Gleise, die von der Hauptachse nur bei Rückwärts-Traversierung direkt erreichbar wären, werden ebenso wie Gleise, die aus ausschließlich indirekt erreichbar sind, gesondert behandelt (siehe Kap. 4.5.1).

Eine weitere, in Abb. 6 in Schritt 1, 3 und 4 dargestellte Heuristik ist die Auswertung der realen geometrischen Längen der TOP-Kanten, um zwischen kurzen Verbindungsgleisen und Gleisen mit einer betrieblich nutzbaren Länge zu unterscheiden. Dies sorgt einerseits für einen optisch ausgewogenen Plan, der einer handgefertigten Zeichnung sehr nahekommt. Andererseits reduziert sich die Anzahl der nachträglich zu bereinigenden Platzkonflikte – wenn ein Gleis in der Realität eine große Entwicklungslänge aufweist, ist die Wahrscheinlichkeit hoch, dass parallel verlaufende Pfade weitere Zwischenknoten beinhalten, sodass sie einen höheren Platzbedarf haben. Durch die großzügige Streckung solcher Gleise bei der Columnberechnung wird genug Raum gelassen und es kommt seltener zu Konflikten, auch wenn der äußerste durchgehende Pfad (erste Heuristik) nicht der tatsächliche Pfad mit dem maßgebenden Platzbedarf ist. Für minimalistische Berechnungen kann es sinnvoll sein, diese zweite Heuristik zu unterdrücken.

4.5 Spezialfälle

Der bis hierhin beschriebene Algorithmus funktioniert nur auf einfachen Gleistopologien mit klaren, baumartigen Verzweigungen in beiden Bahnhofsköpfen. Im Folgenden soll auf die wichtigsten topologischen Situationen eingegangen werden, die eine Herausforderung für den Algorithmus darstellen.

4.5.1 Indirekt angebundene Bereiche

Indirekt angebunden sind Gleise, die der Algorithmus von der Hauptachse ausgehend nicht erreicht, weil dieser auf stumpf befahrenen Weichen nur geradeaus weiterläuft. Bei der Levelberechnung wird km-aufwärts und km-abwärts traversiert, sodass nur aus beiden Richtungen indirekt erreichbare Gleise hierzu zählen. Bei der Columnberechnung wurde auf die Traversierung km-abwärts verzichtet, da diese ohnehin nur eine heuristische Annahme anhand des äußersten durchgehenden Pfades sucht, bei der Traversierung in Gegenrichtung also zu keinem anderen Ergebnis kommen kann.

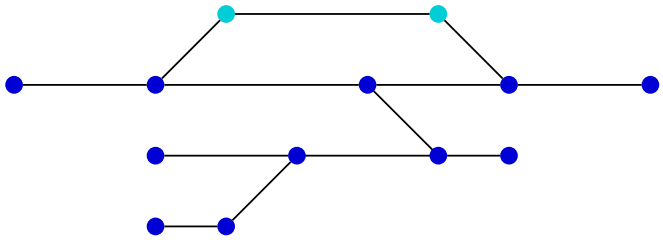

Abb. 7 Indirekt angebundener Bereich unterhalb der Hauptachse
(Eigene Darstellung)

Der Lösungsansatz besteht darin, nach Abschluss der von der Hauptachse ausgehenden Traversierungen die noch unbesuchten TOP-Knoten zu finden. Ausgehend von bereits besuchten Nachbarknoten kann dann eine Traversierung in den indirekten Bereich gestartet werden, die grundsätzlich nach den gleichen Regeln der Level- und Columnberechnung erfolgt.

4.5.2 Column-Konflikte: Heuristikversagen, Stumpfgleise

Wie in Kap. 4.4 bereits erwähnt, basiert die Columnberechnung auf zwei Heuristiken, mit denen eine grafische Darstellung der Gleistopologie gefunden werden soll: der Suche nach dem äußersten durchgehenden Pfad (1) und der Kategorisierung nach der realen Entwicklungslänge der TOP-Kanten (2).

Verschiedene Situationen können dazu führen, dass Column-Werte nachträglich korrigiert werden müssen, um alle TOP-Kanten überlappungsfrei und ausschließlich waagerecht oder diagonal zu darzustellen:

- Die Columnberechnung unterhalb der Hauptachse kommt für Knoten auf der Hauptachse zu höheren Column-Werten als die durch die Traversierung oberhalb der Hauptachse ermittelten Werte; hier haben die Heuristiken keine Einflussmöglichkeit
- Der äußerste durchgehende Pfad ist nicht der maßgebliche Pfad für die Längenentwicklung, d.h. Heuristik (1) versagt und Heuristik (2) fängt das im konkreten Fall nicht ab
- Stumpfgleise werden von den Heuristiken (1) und (2) nur unzureichend erfasst; da sie nicht zurück zur Hauptachse führen, muss ihr Platzbedarf in der weiteren Entwicklung des Bahnhofs gesondert berücksichtigt werden, sonst werden nachfolgende Kanten ggf. überlappend dargestellt.

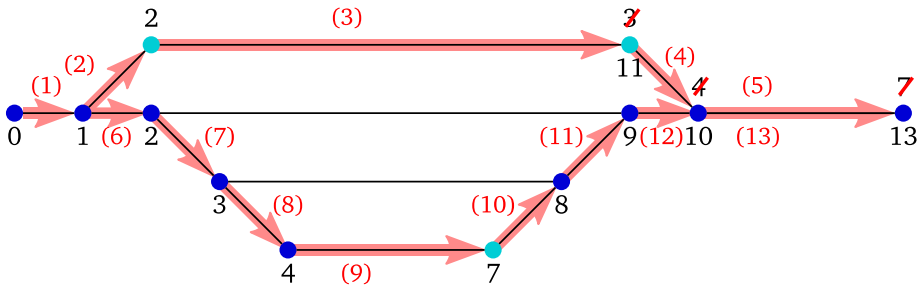

Abb. 8 Die Schritte 12 und 13 berechnen höhere Column-Werte als die vorherigen Schritte 1–5. Die Column-Werte 4 und 7 auf und 3 oberhalb der Hauptachse werden entsprechend angepasst.
(Eigene Darstellung)

Die Lösung besteht in allen Fällen in einer nachträglichen horizontalen Streckung der Darstellung: alle diagonalen Kanten, die direkt an eine Konfliktstelle anschließen, müssen seitlich mitgeschoben werden; alle waagerechten Kanten können beliebig gestreckt werden. Die Verschiebung erfolgt in Richtung der Traversierung, d.h. in (im aktuellen Durchlauf) noch nicht traversierte Bereiche hinein.

4.5.3 Level-Konflikte: Parallele Gleise ohne Zwischenknoten, äußere Umfahrung

In Abb. 5 wird in Schritt 6 bereits die Notwendigkeit von Gerüstknoten an parallelen Gleisen mit gleichem Start- und Endknoten gezeigt. Auch in weiteren Konstellationen können Gerüstknoten notwendig sein, um Level sinnvoll vergeben zu können.

Darüber hinaus baut die Levelberechnung auf der Hypothese auf, dass Verzweigungen in jedem Bahnhofskopf baumartig von innen nach außen stattfinden, d.h. nach Abzweig eines äußeren Gleises zweigt aus der Mitte des Graphen nicht noch einmal ein inneres Gleis ab. In manchen Bahnhöfen ist diese Randbedingung aus keiner Traversierungsrichtung gegeben, dann handelt es sich um eine äußere Umfahrung innerer Gleise. Wird ein solcher Fall erkannt, muss der gesamte Graph der äußeren Umfahrung (es kann sich hier auch um einen größeren zusammenhängenden Bereich handeln) um eine entsprechende Anzahl Level angehoben werden.

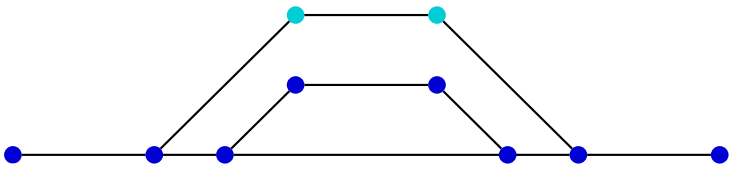

Abb. 9 Äußere Umfahrung innerer Gleise
(Eigene Darstellung)

4.5.4 Kreuzende Gleise: Kreuzung, EKW, DKW, Brücken

Eine große Herausforderung für einen traversierenden Algorithmus stellen alle Arten von kreuzenden Gleisen dar, d.h. höhenfreie Kreuzungen (Brücken), höhengleiche Kreuzungen sowie einfache und doppelte Kreuzungsweichen (EKW/DKW). Für eine korrekte Darstellung sind bei Letzteren die Fahrmöglichkeiten in eine Symboldarstellung umzuwandeln, d.h. die innenliegenden TOP-Kanten dürfen nicht gezeichnet werden. Stattdessen soll die gesamte Kreuzungsweiche als einziger Punkt mit Level und Column berücksichtigt werden.

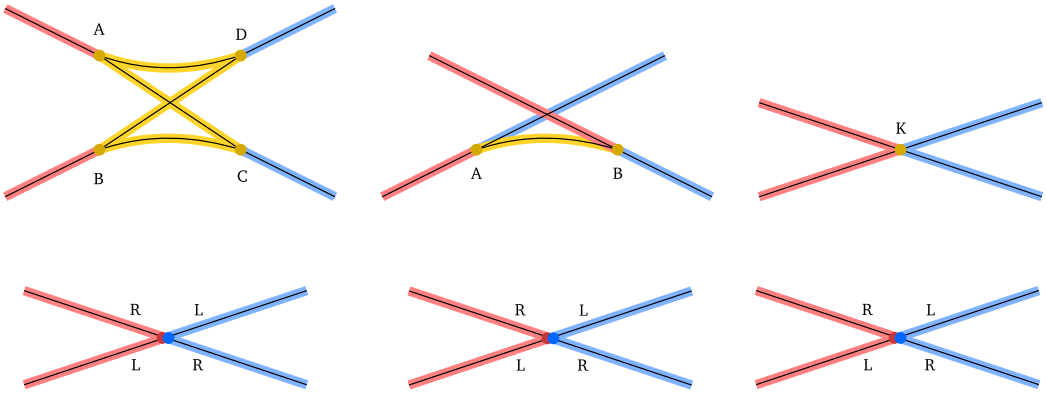

Abb. 10 Modellierung von DKW, EKW und Kreuzung als Doppelknoten
(Eigene Darstellung)

Für die Berücksichtigung im Algorithmus wurde der *Doppelknoten* als Unterklasse des TOP-Knotens entwickelt (vgl. Abb. 10), der eine besondere Berücksichtigung bei der Traversierung erfährt: Während an einer Weiche bei spitzer Befahrung beide Stränge der stumpfen Seite als Nachbarn ausgegeben werden und bei stumpfer Befahrung nur die spitze Seite, wird bei einem Doppelknoten immer das in Befahrungsrichtung gegenüberliegende Kantenpaar als Nachbarn ausgegeben.

Mit dieser Logik kann der Doppelknoten wie jeder andere TOP-Knoten vom Algorithmus behandelt werden. In der Darstellung der Level- und Columnberechnung (Abb. 5 und Abb. 6) ist tatsächlich ein solcher Doppelknoten (orange gefärbt) enthalten. Für den Algorithmus ist es unerheblich, ob ein Doppelknoten eine DKW, EKW oder eine einfache Kreuzung ist – er wertet immer beide gegenüberliegenden Kanten aus, auch wenn sie aus der eingefahrenen Richtung in der Realität gar nicht erreicht werden können.

Die Doppelknoten werden vor dem Beginn des Algorithmus in einem Vorprozessierungsschritt erzeugt, siehe Kap. 4.1

Dieser Ansatz für die Behandlung von Kreuzungsweichen und Kreuzungen bewährte sich so gut, dass er auf sämtliche kreuzenden Gleise ausgeweitet wurde – hierzu zählen höhenfreie Gleiskreuzungen (Brücken) und höhengleiche Gleiskreuzungen, die nicht als solche in den AVANI-Daten modelliert sind[9]. In diesen Fällen werden also durchlaufende TOP-Kanten, die sich geometrisch in einem Punkt schneiden, an diesem Schnittpunkt unterbrochen und mit einem Doppelknoten verbunden, auch wenn hier keine navigierbare Wegebeziehung zwischen den kreuzenden Gleisen vorliegt. Der Doppelknoten heftet die beiden kreuzenden Kanten zusammen, sodass diese in der gesamten Behandlung durch den Algorithmus nicht mehr gegeneinander verschoben werden können.

5 Platzierung von Objekten

Für die Platzierung von Objektsymbolen im Plan ist ein gerichteter Bezug zur Gleistopologie erforderlich: Punktförmige Objekte (z.B. Signale) sind im Regelfall einer TOP-Kante zuzuordnen, flächige Objekte (Bahnsteig, Brücke, Bahnübergang) können mehrere TOP-Kanten referenzieren. Der gerichtete Bezug besteht im Verweis auf die TOP-Kante und einer Information zur Ausrichtung des Objekts (Signale in/entgegen km-Richtung, Ausrichtung von Weichensymbolen an die anliegenden TOP-Kanten).

Ein solcher Bezug liegt in AVANI derzeit nicht vor und muss daher geometrisch ermittelt und ggf. vom Anwender korrigiert werden. Dies ist durch eine Umkreissuche für punktförmige Objekte bzw. eine Hüllpolygonbildung und Schnittkantensuche für flächige Objekte realisierbar.

Punktförmige Objekte werden anschließend durch vorgefertigte Symbole und attributgesteuerte Textbeschriftungen auf oder neben ihrer Bezugskante dargestellt. Flächige Objekte erhalten eine Liniensymbolik entlang ihrer Bezugskanten (z.B. Doppellinie bei Bahnsteigkanten und Querlinie am Bahnsteigbeginn und -Ende)

Der Platzbedarf von Symbolen kann Auswirkungen auf die erforderliche Gesamtgröße des Plans haben. In vertikaler Richtung kann der Platzbedarf durch das Vorhalten standardisierter „Platzierungsbänder“ auf und zwischen den Gleisen systematisch bereitgestellt werden, die Symbol- und Textgröße kann hieran angepasst werden. In horizontaler Richtung kann es passieren, dass die Anzahl zu platzierender Symbole das Platzangebot auf der TOP-Kante übersteigt. Dann ist es erforderlich, den Plan nochmals nach den in Kap. 4.5.2 erwähnten Prinzipien zu strecken.

6 Evaluation des Ansatzes

Abb. 11 zeigt beispielhaft ein Ergebnis unseres Algorithmus im Vergleich zu lagegenauen Eingangsdaten und einem handgezeichneten Plan. Das Ziel, schematische Pläne für Betriebsstellen von einfacher bis mittlerer

[9] Anm.: die Modellierung von Kreuzungen als eigene topologische Knoten in den Bahn-Geodaten wurde zwar schon im Jahr 2005 im AVANI-Vorgängersystem zum neuen Standard erklärt, aber seither nicht flächendeckend auf die Bestandsdaten ausgerollt.

Komplexität zu erzeugen, die keinen oder einen überschaubaren Nachbearbeitungsaufwand übriglassen, wurde erreicht.

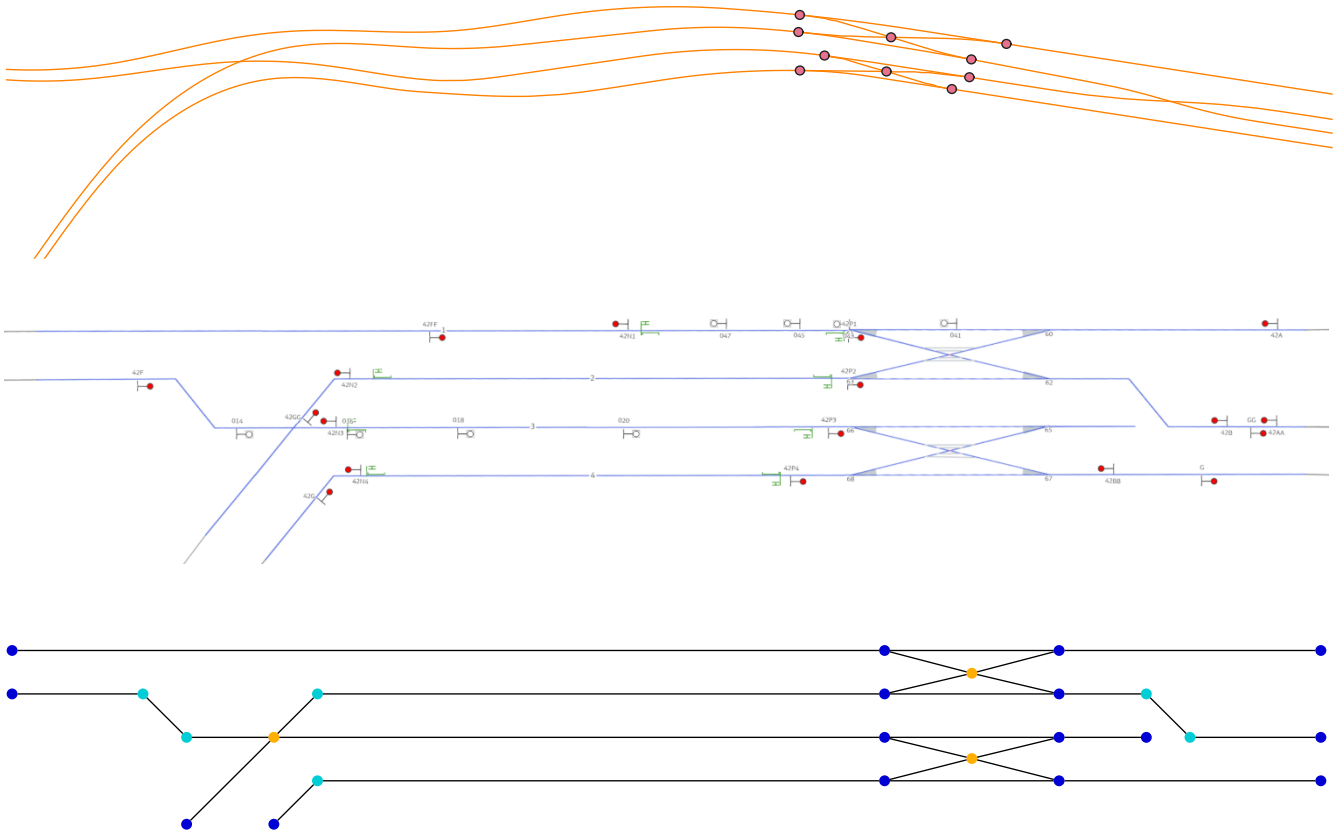

Abb. 11 Bf. München-Donnersbergerbrücke
Oben: Lagegenaue AVANI-Daten – zur besseren Erkennbarkeit auf doppelte Höhe skaliert
Mitte: Darstellung im Trassenfinder
Unten: Beispielhafte Ausgabe des Algorithmus
Quelle: Eigene Darstellung, Mitte: [13]

Charakteristisch für den SOfA-Algorithmus sind im Vergleich zu den Ansätzen in Kap. 2 folgende Merkmale:

1. Es handelt sich um einen deterministischen Ansatz, der zur Gewinnung der schematischen Darstellung den Graphen mehrfach in einer Mischung aus Breiten- und Tiefensuche abläuft.
2. Die Koordinaten (hier Level und Column genannt) werden den Knoten zugewiesen, nicht den Kanten. Ggf. werden zusätzlich erforderliche Gerüstknoten eingefügt.
3. Es wird die reine Topologie ausgewertet; alle geometrischen Betrachtungen werden in eine Vorprozessierung ausgelagert und dem Algorithmus in Form der Topologieobjekte und ihrer Attribute übergeben.
4. Der Algorithmus erhält nur die Beziehung rechter Strang / linker Strang an jeder Weiche aufrecht; nicht aber die Information, welcher Strang die geradlinige Verlängerung der Weichenspitze bildet. Dadurch werden die Pläne platzsparender gezeichnet als z.B. in (Luteberget 2021).
5. Die Traversierung berücksichtigt eine Hauptachse des Plans und berechnet Level vorwärts und rückwärts von der Hauptachse nach beiden Seiten, Columns nur vorwärts. Jeweils anschließend werden die indirekt angebundenen Bereiche analysiert.
6. Die Traversierung läuft von der stumpfen Seite einer Weiche nur über die Spitze weiter und ignoriert den Nachbarstrang, um die Komplexität zu reduzieren.

7. In der Column-Berechnung kommen zwei Heuristiken zur Anwendung:
    a. Äußerster durchgehender Pfad
    b. Streckung nach Längenverhältnissen
8. Alle Arten von höhengleichen und höhenfreien Kreuzungen und Kreuzungsweichen werden durch die Vorprozessierung als Doppelknoten modelliert und einheitlich vom Algorithmus behandelt – die reale Navigierbarkeit spielt keine Rolle.

7 Ausblick

Durch die Behandlung der in Kap. 4.5 aufgezählten Spezialfälle ist ein gebrauchsfähiger Algorithmus entstanden. Trotzdem gibt es aber in einzelnen Betriebsstellen weiterhin topologische Sonderfälle, gegen die der Algorithmus robust gemacht werden muss.

Die erzeugten Pläne müssen sich nun in den Augen derer, die täglich mit ihnen Arbeiten, bewähren. Da wie beschrieben die optischen Anforderungen an einen schematischen Plan nur partiell in Form von Regelwerk formalisiert sind, musste im Entwicklungsprozess in großem Umfang auf Annahmen aufgrund vorliegender Beispielpläne zurückgegriffen werden.

Der SOfA-Algorithmus ist als Java-Bibliothek implementiert und wurde für die Anwendungsfälle der BAst-Erstellung und des PlanPro-ZN/ZL-Planers entwickelt. Er kann und soll aber darüber hinaus auch in anderen Anwendungsfällen zum Einsatz gebracht werden. Vorbedingung ist die Schaffung einer entsprechenden Vorprozessierung der jeweiligen Eingangsdaten auf den Informationsbedarf und das Datenmodell des Algorithmus.

8 Literaturverzeichnis

[1] Gély L.; Dessagne, G.; Pesneau, P., Vanderbeck F.: A multi scalable model based on a connexity graph representation. In: Computers in Railways XII (2010). WIT Press, S. 193–204.

[2] User:Jonnykwe: Liniennetz S-Bahn Rhein-Main.svg [Online] Available at: https://commons.wikimedia.org/wiki/File:Liniennetz_S-Bahn_Rhein-Main.svg (Accessed: 17 June 2025). Lizenziert unter Creative Commons Namensnennung-Share Alike 4.0 International.

[3] Infrastrukturregister der DB InfraGO AG [online]. Available at: https://geoviewer.deutschebahn.com/maps/?locale=de#/context/SPLN/275634?Spurplan=253706 (Accessed: 17 June 2025)

[4] Ril. 819.0102A01: Sicherungstechnische Pläne und Tabellen 3.0. DB InfraGO AG (Stand 01.03.2025)

[5] Wolff, Alexander: Drawing subway maps: A survey. In: Informatik-Forschung und Entwicklung (2007) 22, S. 23–44

[6] Nöllenburg, Martin: A survey on automated metro map layout methods. In: Schematic Mapping Workshop (2014)

[7] Wu, Hsiang-Yun; Niedermann, Benjamin; Takahashi, Shigeo and Nöllenburg, Martin: A Survey on Computing Schematic Network Maps: The Challenge to Interactivity. In: The 2nd Schematic Mapping Workshop (2019)

[8] Wu, Hsiang-Yun; Niedermann, Benjamin, Takahashi; Shigeo, Roberts; Maxwell J.; Nöllenburg, Martin: A survey on transit map layout–from design, machine, and human perspectives. In: Computer Graphics Forum (2020), Vol. 39. No. 3.

[9] Nöllenburg, Martin; Wolff A.: Drawing and Labeling High-Quality Metro Maps by Mixed-Integer Programming. In: IEEE Transactions on Visualization and Computer Graphics (2011), vol. 17, no. 5, S. 626–641

[10] Luteberget, Bjørnar; Johansen, Christian: Drawing with SAT: four methods and A tool for producing railway infrastructure schematics. In: Formal Aspects of Computing (2021), Nr. 33, S. 829–854

[11] Brands, Angela: Automatic generation of schematic diagrams of the Dutch railway network. Unpublished MS thesis. Radboud University, Nijmegen, the Netherlands (2016). Available at https://www.math.ru.nl/~bosma/Students/AngelaBrandsMSc.pdf (Accessed: 27 May 2025)

[12] Bachmann, Volkmar; Trinckauf, Jochen: Digitalisierung und Projektbeschleunigung als Systemlösung. In: Signal und Draht (2024), Nr. 01-02, S. 6–12.

[13] Trassenfinder der DB InfraGO AG [online] Available at: https://trassenfinder.de/ (Accessed: 18 July 2025)

[14] Fruchterman, Thomas M. J.; Reingold, Edward M.: Graph drawing by force-directed placement. In: Software: Practice and Experience (1991), Volume 21, Issue 11, S. 1129–1164