



**Fachhochschule Braunschweig/Wolfenbüttel**

Karl-Scharfenberg-Fakultät Salzgitter

Institut für Verkehrsmanagement

## **Diplomarbeit**

zum Thema:

### **Untersuchung von Methoden zur Vermeidung von Deadlocks in synchronen Eisenbahnsimulationsprogrammen**

**eingereicht bei:** Prof. Dr.-Ing. W.-R. Runge  
Dipl. Ing. A. Wulf

**vorgelegt von:** Jacob Kohlruss  
Matr.-Nr.: 40388355  
Kleewiese 12  
38229 Salzgitter

Salzgitter, den 15.08.2007

## Abkürzungsverzeichnis

AAK	Ausschlussverfahren durch abhängige Kanten
DLR	Deutsches Zentrum für Luft und Raumfahrt
DRR	Dynamic Route Reservation
EDV	Elektronische Daten Verarbeitung
IFS	Institut für Verkehrsführung und Fahrzeugsteuerung
MCA	Movement Consequence Analysis
ZV-Diagramm	Zugkraft/Geschwindigkeits-Diagramm

## Abbildungsverzeichnis

Abbildung 1: Deadlock-Situation in der EDV .....	4
Abbildung 2: Einfacher Deadlock im System Bahn.....	5
Abbildung 3: Ansicht der Infrastruktur im RailSys .....	8
Abbildung 4: Beispiel zur Definition der Abbildungsdarstellung .....	11
Abbildung 5: Teilinfrastrukturansicht in RailSys, Szenario 1 .....	14
Abbildung 6: Bildfahrplanansicht in RailSys, ein Takt im Szenario 1 .....	15
Abbildung 7: Teilinfrastrukturansicht in RailSys, Szenario 2.....	15
Abbildung 8: Bildfahrplanansicht in RailSys, ein Takt im Szenario 2 .....	16
Abbildung 9: Teilinfrastrukturansicht in RailSys, Szenario 3.....	17
Abbildung 10: Bildfahrplanansicht in RailSys, ein Takt in Szenario 3 .....	18
Abbildung 11: Teilinfrastrukturansicht in RailSys, Szenario 4, Bahnhof N .....	18
Abbildung 12: Teilinfrastrukturansicht in RailSys, Szenario 4, Bahnhof M.....	19
Abbildung 13: Bildfahrplanansicht in RailSys, ein Takt in Szenario 4 .....	20
Abbildung 14: Teilinfrastrukturansicht in RailSys, Szenario 5.....	21
Abbildung 15: Wartekette im System Bahn – Deadlock-Situation .....	26
Abbildung 16: Trivialer Deadlock .....	28
Abbildung 17: Deadlock-Vermeidung durch Kapazitätsausgleich.....	28

---

Abbildung 18: Deadlock in der Simulation durch nicht aktivierte Methoden .....	31
Abbildung 19: Einfacher Deadlock mit drei Zügen.....	31
Abbildung 20: Deadlock-Situation mit drei Zügen, Parallelfahrt im Bahnhof 13	31
Abbildung 21: Deadlock-Situation mit drei Zügen, Überholung durch Zug 03...	32
Abbildung 22: Deadlock-Situation mit vier Zügen aus Folgefahrt.....	34
Abbildung 23: Bevorstehende Deadlock-Situation bei größerer Kapazität .....	35
Abbildung 24: Deadlock-Situation bei größerer Kapazität durch Folgefahrt .....	36
Abbildung 25: Deadlock-Situation bei größerer Kapazität durch Parallelfahrt....	36
Abbildung 26: Deadlock im Bahnhof M .....	38
Abbildung 27: Deadlock-Situation im Bahnhof E.....	39
Abbildung 28: Deadlock in Szenario 5 durch Simulationsfehler .....	40
Abbildung 29: Deadlock-Situation durch Versagen der eingleisigen Disposition	41
Abbildung 30: Deadlock in Bahnhof E durch mangelnde Kapazität .....	41
Abbildung 31: Entstehen einer Deadlock-Situation trotz Fahrwegsreservierung	54
Abbildung 32: Entstehen einer Deadlock-Situation trotz Signalvorschaubereich	57
Abbildung 33: Deadlock-Vermeidung im Bahnhof M .....	58
Abbildung 34: Deadlock-Vermeidung durch abhängige Kanten .....	58
Abbildung 35: Darstellung der primären und sekundären Zufolge .....	66
Abbildung 36: Vermeidung eines trivialen Deadlocks durch MCA.....	67
Abbildung 37: Vermeidung eines Deadlocks aus Parallelfahrt durch MCA .....	68
Abbildung 38: Vermeidung eines Deadlocks durch MCA, keine Parallelfahrt ....	69
Abbildung 39: Vermeidung eines Deadlocks aus Folgefahrt mit MCA.....	70
Abbildung 40: Vermeidung eines trivialen Deadlocks durch DRR .....	74
Abbildung 41: Vermeidung eines Deadlocks aus Parallelfahrt durch DRR .....	75
Abbildung 42: Vermeidung eines Deadlocks aus Folgefahrt durch DRR.....	76
Abbildung 43: Darstellung der Leistungsfähigkeitsminderung durch DRR .....	77

---

Abbildung 44: Darstellung der Regel 01 der Methode AAK.....	80
Abbildung 45: Darstellung der Regel 02 der Methode AAK.....	81
Abbildung 46: Darstellung der Regel 03 der Methode AAK.....	82
Abbildung 47: Darstellung der Regel 04 der Methode AAK.....	83
Abbildung 48: Darstellung der Regel 05 der Methode AAK.....	84
Abbildung 49: Darstellung der Regel 06 der Methode AAK.....	85
Abbildung 50: Darstellung der Regel 07 der Methode AAK.....	86
Abbildung 51: Darstellung der Regel 08 der Methode AAK.....	87
Abbildung 52: Beispiel für weitere Regeln nach AAK .....	88
Abbildung 53: Analyse der Leistungsfähigkeitsbeeinflussung durch DRR.....	90
Abbildung 54: Einfachen Folgefahrt durch Zug 03; Ergänzung DRR.....	92
Abbildung 55: Mehrfachen Folgefahrt durch Zug 03; Ergänzung DRR .....	92
Abbildung 56: Vermeidung einer komplexen Deadlock-Situation durch DRR ....	93

## **Tabellenverzeichnis**

Tabelle 1: Deadlock-Wahrscheinlichkeiten durch Zielgleisänderung .....	52
Tabelle 2: Deadlock-Wahrscheinlichkeiten durch Fahrwegsreservierung.....	54
Tabelle 3: Deadlock-Wahrscheinlichkeiten durch Signalvorschaubereich .....	56
Tabelle 4: Deadlock-Wahrscheinlichkeiten durch Methoden in RailSys .....	60
Tabelle 5: Übersicht der Bewertung der untersuchten Methoden .....	98

---

## Inhaltsverzeichnis

<b>1</b>	<b>Einführung .....</b>	<b>1</b>
1.1	Institut für Verkehrsführung und Fahrzeugsteuerung .....	1
1.2	Motivation der Arbeit .....	1
1.3	Vorgehensweise der Arbeit .....	2
<b>2</b>	<b>Grundlagen .....</b>	<b>4</b>
2.1	Definition Deadlock .....	4
2.2	Unterscheidung der Simulationsverfahren.....	6
2.3	Grundsätzliche Anti-Deadlock-Strategien .....	6
2.4	RailSys .....	7
2.5	Zugsicherungsverfahren .....	9
2.6	Definition der Abbildungsdarstellung .....	10
2.7	Definition der Aussagenbeschreibung.....	11
<b>3</b>	<b>Definition der Szenarien .....</b>	<b>13</b>
3.1	Annahmen zur Infrastruktur- und Fahrplankonstruktion.....	13
3.2	Szenario 1 .....	14
3.3	Szenario 2 .....	15
3.4	Szenario 3 .....	16
3.5	Szenario 4 .....	18
3.6	Szenario 5 .....	20
3.7	Infrastrukturnetz der Simulation .....	22
3.7.1	Erzeugen von gestörten Fahrplänen .....	22
3.7.2	Erzeugen von Teilnetzen.....	23
<b>4</b>	<b>Deadlock-Betrachtung.....</b>	<b>25</b>
4.1	Voraussetzungen für einen Deadlock.....	25
4.2	Randbedingungen für einen Deadlock .....	27
4.3	Analyse der Deadlock-Situationen.....	29
4.3.1	Trivialer Deadlock .....	29
4.3.2	Deadlock mit drei Zügen (aus Parallelfahrt) .....	31
4.3.3	Deadlock mit vier Zügen (aus Folgefahrt) .....	33

---

4.3.4	Deadlocks bei höherer Kapazität der Infrastruktur .....	34
4.3.5	Deadlocks in Bahnhöfen mit Doppelblock .....	37
<b>5</b>	<b>Implementierte Deadlock-Vermeidung in RailSys .....</b>	<b>42</b>
<b>5.1</b>	<b>Methoden durch den Disponenten .....</b>	<b>42</b>
5.1.1	Disposition auf eingleisigen Stecken .....	43
5.1.2	Ausweichgleise .....	44
5.1.3	Zielgleisänderung .....	44
5.1.4	Disposition für Alternativfahrwege .....	45
<b>5.2</b>	<b>Weitere Methoden in RailSys .....</b>	<b>47</b>
5.2.1	Fahrwegsreservierung .....	47
5.2.2	Signalvorschaubereich .....	48
5.2.3	Abhängige Kanten .....	48
<b>5.3</b>	<b>Test der Methoden in RailSys .....</b>	<b>49</b>
5.3.1	Vorstellung des Testverfahrens .....	49
5.3.2	Test der Zielgleisänderung .....	50
5.3.3	Test der Disposition für Alternativfahrwege .....	52
5.3.4	Test der Fahrwegsreservierung .....	53
5.3.5	Test des Signalvorschaubereiches .....	55
5.3.6	Test der abhängigen Kanten .....	57
5.3.7	Test der Methoden in Kombination .....	59
<b>5.4</b>	<b>Fazit über die Methoden in RailSys .....</b>	<b>62</b>
<b>6</b>	<b>Deadlock-Vermeidung nach der Literatur .....</b>	<b>65</b>
<b>6.1</b>	<b>Movement Consequence Analysis (MCA) .....</b>	<b>65</b>
6.1.1	Beschreibung der Methode MCA .....	65
6.1.2	Test der Methode MCA .....	67
6.1.3	Auswertung von MCA .....	71
<b>6.2</b>	<b>Dynamic Route Reservation (DRR) .....</b>	<b>72</b>
6.2.1	Beschreibung der Methode DRR .....	72
6.2.2	Test der Methode DRR .....	74
6.2.3	Auswertung von DRR .....	76
<b>7</b>	<b>Entwurf von Verfahrnung zur Deadlock-Vermeidung .....</b>	<b>79</b>
<b>7.1</b>	<b>Ausschlussverfahren durch abhängige Kanten (AAK) .....</b>	<b>79</b>
7.1.1	Regeln für Deadlocks aus Überholungen .....	79

---

7.1.2	Regeln für Deadlocks aus Kreuzungen .....	82
7.1.3	Regel aus der Implementierung in RailSys.....	85
7.1.4	Bewertung der Methode .....	88
<b>7.2</b>	<b>Ergänzung der Methode DRR .....</b>	<b>89</b>
7.2.1	Definition der Regeln.....	89
7.2.2	Test der ergänzten Methode DRR .....	92
7.2.3	Bewertung der Methode .....	94
<b>8</b>	<b>Diskussion zur Deadlock-Vermeidung .....</b>	<b>95</b>
<b>9</b>	<b>Zusammenfassung und Ausblick .....</b>	<b>100</b>
	<b>Literatur und Quellenverzeichnis .....</b>	<b>103</b>
	<b>Eidesstattliche Erklärung.....</b>	<b>105</b>
	<b>Anhang .....</b>	<b>106</b>

---

# 1 Einführung

## 1.1 Institut für Verkehrsführung und Fahrzeugsteuerung

Seit März 2001 besteht das Institut für Verkehrsführung und Fahrzeugsteuerung (IFS) beim Deutschen Zentrum für Luft- und Raumfahrt (DLR). Es deckt damit einen Teil des vierten Forschungsschwerpunktes „Verkehr“ beim DLR ab.

Unter der Leitung von Prof. Dr.-Ing. Karsten Lemmer entwickeln Wissenschaftler der unterschiedlichsten Fachbereiche neue Lösungen zur Erhöhung der Sicherheit und Effizienz im straßen- und schienengebundenen Verkehr. Das Spektrum der Arbeit reicht von der Grundlagenforschung bis zu der Erstellung von Prototypen und orientiert sich am Bedarf von Wirtschaft und Gesellschaft. [DLR06]

## 1.2 Motivation der Arbeit

In den Fragen des Eisenbahnbetriebs und seiner Bewertung hat der Einsatz von Simulationssoftware in den letzten 20 Jahren eine immer größere Bedeutung gewonnen. Bei der Planung und Bewertung von Fahrplan- und Infrastrukturvarianten ist die Simulation heute nicht mehr wegzudenken. Im Bereich der Eisenbahnbetriebssimulation haben sich zwei prinzipiell unterschiedliche Verfahren – die synchrone und asynchrone Simulation – entwickelt. Die Einsatzmöglichkeiten sowie die Vor- und Nachteile des jeweiligen Verfahrens, werden in der Literatur je nach Sichtweise des jeweiligen Autors unterschiedlich bewertet.

Ein bekanntes und allgemein anerkanntes Problem der synchronen Simulation ist, dass in bestimmten Situationen Deadlocks auftreten können. Ein Deadlock ist eine Situation, in der sich die im Fahrplan abgebildeten Zugfahrten so ineinander verklemmen, dass kein Zug mehr in der Lage ist, das System in der Wunschrichtung zu befahren. Deadlocks können immer dann auftreten, wenn Gleisabschnitte im Zweirichtungsbetrieb befahren werden. Die gängigen Simulationsprogramme versuchen diese Situationen durch die Anwendung von Algorithmen und Regeln zu vermeiden. Dies gelingt jedoch nur bedingt.

---

Zurzeit werden Simulationsdurchläufe (in RailSys) mit Deadlocks nicht betrachtet, wobei nicht unterschieden wird, ob der Deadlock aufgrund der Simulation, von Fehlern in der Infrastruktur und/oder des Fahrplanes entstanden ist. Weiterhin ist bei großen Netzbetrachtungen mit bestimmten Gegebenheiten das Problem vorhanden, dass kein Simulationsdurchlauf abgeschlossen werden kann, weil jeder Durchlauf durch einen Deadlock abgebrochen wird. Dieses Phänomen kann auch schon bei kleinen Unregelmäßigkeiten auftreten. Dies macht eine Untersuchung großer Netze unmöglich und schränkt somit die Anwendbarkeit der synchronen Simulation ein.

### **1.3 Vorgehensweise der Arbeit**

Die vorliegende Arbeit beschäftigt sich mit dem Thema Deadlock in synchronen Eisenbahnsimulationen. Beispielhaft wird hier das Simulationsprogramm RailSys untersucht.

Aufbauend auf einen kurzen Überblick über Verfahren der Eisenbahnbetriebssimulation soll die Problematik der Deadlock-Entstehung in synchronen Simulationsprogrammen hergeleitet und dargestellt werden. Hierzu werden geeignete Szenarien erstellt. Anhand derer wird gezeigt, wie und warum Deadlocks entstehen.

Das betrachtete Simulationsprogramm RailSys wird auf seine Möglichkeiten zur Vermeidung von Deadlocks untersucht. Es werden Verfahren aufgezeigt, die in dem Simulationstool vorhanden sind. Ihre optimale Parametrierung wird ermittelt und weiterhin die Möglichkeiten und Grenzen dieser Algorithmen aufgezeigt.

Anschließend daran werden Verfahren vorgestellt, die die Literatur zu diesem Thema bereitstellt.

Die Generierung eigener Verfahren und Algorithmen zur Vermeidung von Deadlocks ist ein weiterer Schwerpunkt der Arbeit. Die Verfahren sind mittels der aufgestellten Szenarien zu überprüfen. Eine Implementierung dieser Verfahren in das Simulationstool erfolgt dabei nur bedingt.

---

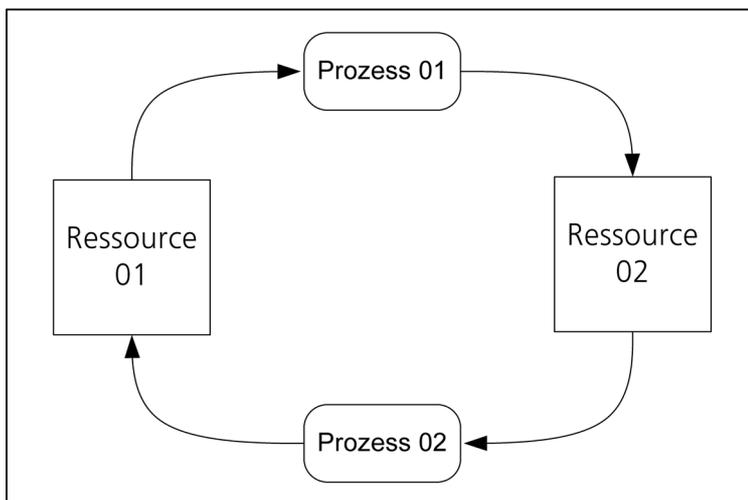
Sowohl die in der Literatur gefundenen als auch die eigenen aufgestellten Lösungsvorschläge werden hinsichtlich ihrer Möglichkeiten und Einsatzgrenzen bewerten (z. B. Komplexität der Infrastruktur).

---

## 2 Grundlagen

### 2.1 Definition Deadlock

Der Begriff Deadlock entstammt der Elektronischen Daten Verarbeitung (EDV). Er bezeichnet im einfachsten Fall eine Situation, in der zwei Prozesse auf jeweils eine Ressource zugreifen, die wiederum der andere Prozess in Besitz hat. So entsteht ein Kreislauf der Bedingungen, der nicht wieder gelöst werden kann. Abbildung 1 stellt diese Situation schematisch dar.



**Abbildung 1: Deadlock-Situation in der EDV**

Im Bahnwesen wird ein Zustand als Deadlock (Verklemmung) bezeichnet, wenn im Regelbetrieb eine nicht mehr lösbare Konfliktsituation, die meist durch einen Dispositionsfehler eines mit der Betriebslenkung beauftragten Person (z.B. Fahrdienstleiter) oder bei einer Zuglenkung durch den Einsatz von Software, die solche Konflikte nicht verhindert oder die fehlerhaft ist, entsteht. [NAU04]

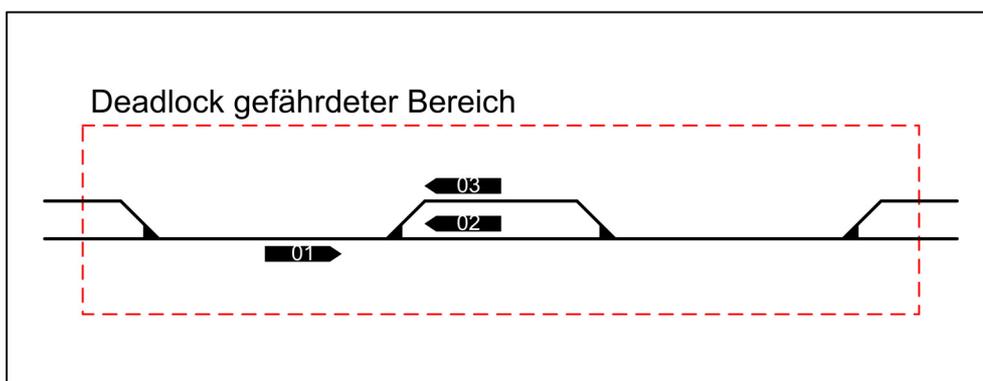
In der Realität werden Deadlocks erst einmal durch einen konfliktfreien Fahrplan vermieden. Das Problem entsteht, wenn der Fahrplan nicht mehr planmäßig verläuft und von Verspätungen überlagert wird. In diesem Fall sorgen im realen Betrieb Disponenten für einen reibungslosen Ablauf und die Vermeidung von Deadlocks.

Weiterhin beschreibt ein Deadlock den Zeitpunkt in einem System, wenn es schon zu einer Verklemmung im System gekommen ist. Ob ein Deadlock

entsteht, entscheidet sich jedoch schon viel früher. Der Zeitpunkt wird durch den „Deadlock gefährdeten Bereich“ bestimmt. Die Größe dieses Bereiches ist von der Kapazität der Infrastruktur und der Anzahl der Züge im System abhängig. Man spricht deshalb auch vom Deadlock-Zustand, der den Deadlock zur Folge hat. Ist der Deadlock-Zustand erst einmal eingetreten, kann er auch nicht mehr durch dispositive Maßnahmen beseitigt werden. Darum ist das Erkennen eines Deadlock vor der Einfahrt in den „Deadlock gefährdeten Bereich“ zwingend notwendig. [FER04]

Da der Begriff „Zustand“ die Attribute eines Objektes beschreibt und ein Deadlock kein Objekt ist, wird die Wortkombination (Deadlock-Zustand) nicht weiter verwendet. Vielmehr spiegelt das zukünftige Entstehen eines Deadlocks der Begriff „Deadlock-Situation“ wieder. Eine Situation ist der bestimmte zeitliche und/oder räumliche Zusammenhang eines Sachverhaltes. Diese Definition trifft auf einen Deadlock zu.

Ein Beispiel für einen Deadlock zeigt die Abbildung 2. Hier stehen drei Züge vor einer eingleisigen Strecke sich untereinander gegenüber. Zug 01 wartet auf die Einfahrt in den Bahnhof, wobei die Züge 02 und 03 auf die Ausfahrt warten. Unter diesen Bedingungen besitzt der „Deadlock gefährdete Bereich“ die dargestellte Größe. Die Deadlock-Situation wird in dem Augenblick hergestellt, in dem Zug 01 in den „Deadlock gefährdeten Bereich“ eingefahren ist. Ab diesem Zeitpunkt kann ein Deadlock nicht mehr verhindert werden. Welche Kriterien auf die Größe des „Deadlock gefährdeten Bereich“ einen Einfluss haben, wird im Kapitel 4.2 beschrieben.



**Abbildung 2: Einfacher Deadlock im System Bahn**

## 2.2 Unterscheidung der Simulationsverfahren

Simulationen können nach verschiedenen Kategorien unterteilt werden. Man unterscheidet nach makroskopischer und mikroskopischer Simulation aber auch nach synchroner und asynchroner Durchführung.

Bei einer synchronen Simulation wird zu jeder Zeit das gesamte zu simulierende System betrachtet. Hieraus leitet sich der Begriff synchron ab, da alle Elemente des Systems synchron betrachtet werden. Meistens erfolgt dies in Zeitschritten. Je kleiner die Zeitschritte sind, desto genauer ist die Simulation. Mit dem Kleinerwerden der Zeitschritte steigt jedoch auch der Rechenaufwand.

Weiterhin ist es möglich, zusätzlich das Verhalten der Simulation ereignisorientiert zu steuern und zu lenken. In diesem Bereich muss die Deadlock-Vermeidung angesiedelt werden, da ein Deadlock ein Ereignis ist, das nicht eintreten darf.

Bei der asynchronen Simulation werden die Elemente im System nicht zeitgleich, sondern in einer bestimmten Reihenfolge betrachtet. Die Reihenfolge kann z.B. durch die Priorität eines Elementes festgelegt werden. Im Beispiel der Eisenbahnsimulation bedeutet dies, dass immer der gesamte Zuglauf auf die betrachtete Infrastruktur umgelegt wird. Kann ein Zuglauf nicht eingelegt werden, wird er so lange zeitlich nach hinten geschoben, bis er eingelegt werden kann.

Das Prinzip der asynchronen Eisenbahnsimulation schließt von vornherein das Entstehen von Deadlocks aus, da immer der gesamte Zuglauf eingelegt wird und dieses nicht für zwei Züge gleichzeitig geschieht.

## 2.3 Grundsätzliche Anti-Deadlock-Strategien

Nach Bruns werden Strategien zur Deadlock-Behandlung grundsätzlich in drei Kategorien eingeteilt:

- Deadlock-Verhinderung
  - Deadlock-Vermeidung
-

- Deadlock-Erkennung und -Beseitigung

[BRU83]

Die Deadlock-Verhinderung setzt auf Verfahren, die das Entstehen von Deadlocks unmöglich machen. Ein Beispiel nach Pachl ist das Einstellen des Fahrweges für den gesamten Laufweg des Zuges, also vom Start- bis zum Zielbahnhof. Dieses Verfahren kommt der asynchronen Simulation gleich, die in Kapitel 2.2 beschrieben wird.

Verfahren zur Deadlock-Vermeidung kommen in Systemen zur Anwendung, die grundsätzlich Deadlock fähig sind. Man versucht hier durch Kontrolle des Systems einen Deadlock zu vermeiden. Das Fahren nach einem festen, konfliktfreien Fahrplan ist z.B. eine solche Maßnahme. Die Überwachung des Fahrplans ist dabei ein wesentliches Element. In frühen Zeiten war das Überwachen komplexer Systeme jedoch nicht möglich, da die Rechenleistung nicht zur Verfügung stand.

Die dritte Strategie ist die Deadlock-Erkennung und -Beseitigung. Dieses Verfahren findet allgemein nur Anwendung in Systemen, die eine geringe Deadlock-Wahrscheinlichkeit aufweisen. Erkennung bedeutet in diesem Fall, dass die Situation erkannt wird, in der es schon zu einem Deadlock gekommen ist. Dieser Deadlock wird dann beseitigt. Im realen Eisenbahnbetrieb lässt sich diese Strategie nicht anwenden, weil der Aufwand, den Deadlock zu beseitigen, zu groß ist. Daraus folgt weiterhin, dass diese Strategie nur Anwendung findet, wenn der Aufwand der Deadlock-Beseitigung kleiner, als für die Deadlock-Vermeidung oder -Verhinderung ist. [PAC93] Dieses ist in der synchronen Eisenbahnsimulation zurzeit der Fall. Deadlocks werden detektiert und anschließend die Verklemmung gelöscht oder der Simulationslauf in der Auswertung nicht betrachtet.

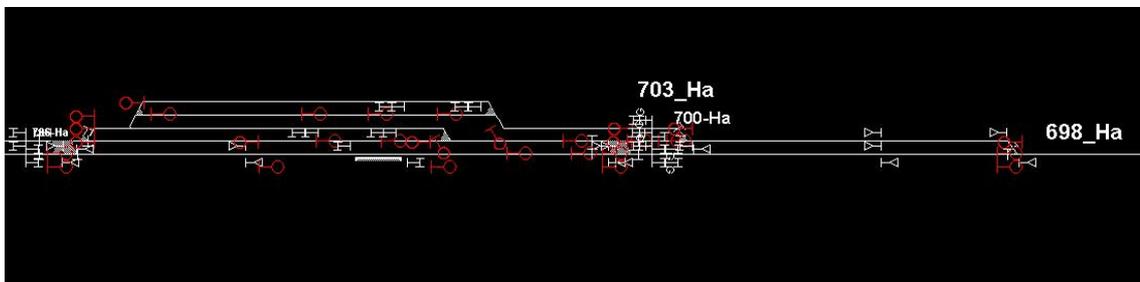
## 2.4 RailSys

Das Simulationstool RailSys der Firma RmCon ist ein Programm zur Analyse, Planung und Optimierung von Betriebsabläufen und -Anlagen in beliebig großen Netzen spurgeführter Verkehrssysteme. [RMC07]

---

Es wird dazu eine synchrone Durchführung der Simulation angewandt. Die Datengrundlage für die Simulation ist mikroskopischer Natur. Das Tool ist in der Lage die Infrastruktur auf einen Meter genau abzubilden.

Die Infrastruktur wird als Knoten-Kanten-Graph eingegeben, wobei die Kanten Gleise abbilden und die Knoten weitere Infrastrukturelemente, wie Signale, Weichen usw., darstellen. Die Kanten können mit Werten wie Länge, Geschwindigkeit, Neigung und Radius attribuiert werden. Aus diesen Attributen werden in der Simulation der Streckenwiderstand und andere Werte ermittelt. Abbildung 3 zeigt die Darstellung einer Infrastruktur im Infrastrukturmanager.



**Abbildung 3: Ansicht der Infrastruktur im RailSys**

Nach der Eingabe der Infrastruktur muss in einem weiteren Programmteil ein Fahrplan für die erzeugte Infrastruktur erstellt werden. Die Definition von Triebfahrzeugen und Zuggattungen bildet die Grundlage für den zu erstellenden Fahrplan. Die Fahrdynamik der Züge resultiert aus der Attributierung der Zuggattungen. Hier werden individuelle Zugkraft/Geschwindigkeits-Diagramme (ZV-Diagramm) hinterlegt, aus denen der Zugkraftüberschuss in Abhängigkeit vom Streckenwiderstand und somit das Beschleunigungsvermögen ermittelt wird. Mit der im Zeitschrittverfahren ermittelten Beschleunigungskurve wird eine realitätsnahe Abbildung der Fahrdynamik gewährleistet.

Der Fahrplan kann nun konstruiert werden, in dem der Laufweg für jeden Zug festgelegt wird. Weiterhin müssen für jeden Halt in einer Station die Ankunfts- und die Abfahrzeit bestimmt werden. Ist es fahrdynamisch nicht möglich diese Zeiten einzuhalten, wird eine Meldung ausgegeben. Hat ein Zug mehr Zeit als er benötigt, fährt er langsamer, um pünktlich in der nächsten Station anzukommen.

Um eine Betriebssimulation durchzuführen, müssen von Verspätungen überlagerte Fahrpläne für jeden Betriebstag erstellt werden. Diese gestörten

Fahrpläne werden auf der Grundlage von zu definierenden Verteilungsfunktionen ermittelt.

Zur Auswertung solcher Simulationen stellt RailSys eine Vielzahl von Protokollen und Kennwerten zur Verfügung. Ein eigener Programmteil widmet sich dieser Aufgabe ausführlich. Hier können Grafiken beliebiger Art erstellt und exportiert werden.

## 2.5 Zugsicherungsverfahren

Aufgrund der Systemmerkmale des Verkehrsträgers Eisenbahn ist der Bremsweg oft länger als der Sichtweg. Die wesentlichen Merkmale sind dabei

- die möglichen hohen Geschwindigkeiten,
- die Spurführung,
- die großen Massen, sowie
- ein geringer Haftreibbeiwert für das System Rad-Schiene.

Diese Eigenschaften und der daraus resultierende lange Bremsweg fordern ein systemgerechtes Zugsicherungsverfahren, um die Kollision von Zügen zu verhindern. Grundsätzlich unterscheidet man nach drei Verfahren:

- Fahren im festen Raumabstand
- Fahren im festen Zeitabstand
- Fahren auf Sicht

Eine Zugsicherung durch Fahrten auf Sicht, wie im Straßenverkehr, ist im schienengebundenen Verkehr aufgrund der Systemmerkmale nur bedingt möglich. Sind bestimmte Randbedingungen gegeben, wie geringe Geschwindigkeiten, kann jedoch auch auf Sicht gefahren werden.

Beim Fahren im festen Zeitabstand wird eine Kollision bei Folgefahrten durch einen zeitlichen Abstand der folgenden Züge vermieden. Bei Gegenfahrten erfolgt die Zugsicherung durch den konfliktfreien Fahrplan, der zwingend eingehalten werden muss. In Nordamerika hat sich bis ins 20. Jahrhundert diese Sicherung auf nicht signalisierten Strecken mit dem Betriebsverfahren „Timetable

---

& Train Order“ als zweckmäßig erwiesen. Die großen Distanzen zwischen den Kreuzungsbahnhöfen auf eingleisigen Strecken gestaltet das Fahren im festen Raumabstand schwierig. Bis heute basieren einige Betriebsverfahren in den USA auf diesem Prinzip. [PAC01]

In Europa hat sich das Fahren im festen Raumabstand durchgesetzt. Schon in der zweiten Hälfte des 19. Jahrhunderts kam dieses Verfahren der Zugsicherung zur Anwendung. [PAC01] Beim Fahren im festen Raumabstand wird die Gleisinfrastruktur in Abschnitte (Blöcke) unterteilt. Jeder Block muss mindestens so groß sein, dass der Zug, der den Block belegt, in diesem Block bremsen kann. Die Zugsicherung ist dadurch gewährleistet, dass sich immer nur ein Zug in einem Block befinden darf.

## **2.6 Definition der Abbildungsdarstellung**

In dieser Arbeit werden vermehrt Abbildungen eingeführt, die dem besseren Verständnis dienen sollen. Insbesondere theoretische und von Abhängigkeiten geprägte Sachverhalte lassen sich besser visuell darstellen, als sie nur in Textform zu beschreiben.

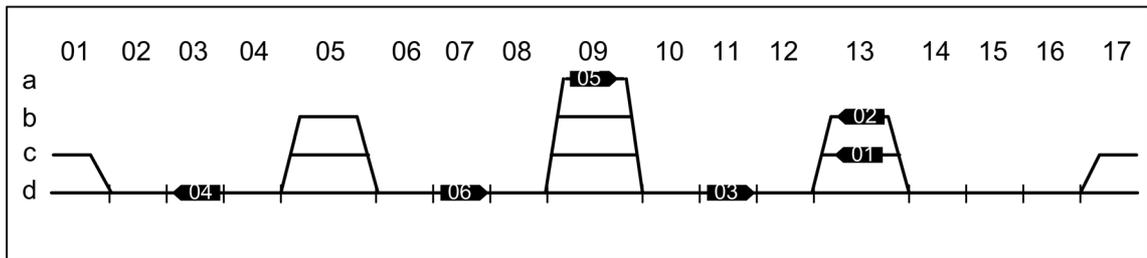
Gerade für die Abbildung von Betriebssituationen auf Gleisinfrastrukturen bittet sich eine grafische Darstellung an. Da in den folgenden Kapiteln die Position einzelner Züge auf der betrachteten Infrastruktur eine wesentliche Rolle spielt, wird diese Infrastruktur mit einem Koordinatensystem überlagert. Das Koordinatensystem bezieht sich hierbei auf die Streckenblöcke der Gleisinfrastruktur.

In den Abbildungen wird der Verlauf der Strecke überwiegend auf der horizontalen Ebene und parallele Gleise auf der vertikalen Ebene dargestellt. Abbildung 4 zeigt ein Beispiel für eine Gleisinfrastruktur, in der das Koordinatensystem Anwendung findet.

So erhält jeder Block eine eigene Koordinate, die sich aus einer Zahl und einem Buchstaben zusammensetzt. Die Position eines Zuges kann nun über die Koordinate des Blockes, in dem der Zug sich befindet, bestimmt werden. So

---

befindet sich Zug 01 z.B. in Block 13c. Ein weiteres Beispiel stellt sich für Zug 04, der sich in Block 03d befindet, dar.



**Abbildung 4: Beispiel zur Definition der Abbildungsdarstellung**

Die Benennung der Bahnhöfe in einer Abbildung bezieht sich jeweils auf die Blocknummer, die die parallelen Gleise im Bahnhof gemeinsam haben. Ein Beispiel hierfür ist, dass Zug 05 sich im Bahnhof 09 befindet.

Anhand dieser Positionsbestimmung durch das Koordinatensystem werden im Nachfolgenden alle Infrastrukturdarstellungen bei der Anwendung von Methoden und Verfahren beschrieben.

## 2.7 Definition der Aussagenbeschreibung

In Kapitel 7.1 wird für die Definition aufgestellter Regeln die Aussagenlogik verwendet. Hier werden Aussagen mit logischen Verknüpfungen (z.B. UND und ODER) zu Regeln definiert. Eine Aussage entspricht hierbei dem richtungsbezogenen Belegungszustand eines Blockes. In der beschreibenden Regel muss aus dem Syntax der Aussage also der Block mit Koordinatenangabe und der Belegungsrichtung (bezogen auf einen betrachteten Zug) hervorgehen.

Ein Beispiel, das sich auf Abbildung 4 bezieht, ist der folgende Zusammenhang. Zug 01 darf die Blöcke 12d, 11d und 10d nur befahren, wenn kein Zug in Gegenrichtung diese Blöcke befährt. Die aufzustellende Logik ergibt sich dann wie folgt:

$$12dGR \vee 11dGR \vee 10dGR$$

Ergibt der Ausdruck „true“, ist die Einfahrt auf den eingleisigen Streckenabschnitt für Zug 01 nicht freigegeben. Der Syntax einer Aussagen ist wie folgt zu deuten:

- 12d – ist der betrachtete Block
- GR – ist die relevante Richtung, bezogen auf den anfragenden Zug
  - GR – steht für belegt in Gegenrichtung
  - R – steht für belegt in gleicher Richtung
  - B – steht für belegt, unabhängig von der Richtung.

Ist, wie im diesen Fall, eine der Aussagen „true“, darf Zug 01 die Strecke nicht befahren.

---

---

## 3 Definition der Szenarien

In diesem Kapitel werden die Szenarien definiert, in denen Deadlocks auftreten können. Hierzu wird in einem ersten Schritte ein Infrastrukturnetz in RailSys erstellt, in dem die unterschiedlichen Szenarien in einzelnen Strecken abgebildet werden. Auf diese Strecken wird jeweils ein konfliktfreier Fahrplan konstruiert. Alle Szenario-Strecken werden anschließend zu dem Infrastrukturnetz zusammengeführt.

### 3.1 Annahmen zur Infrastruktur- und Fahrplankonstruktion

Da für die Deadlock-Betrachtung die Fahrdynamik und die geografischen Gegebenheiten keine Rolle spielen, wird vereinfachend angenommen, dass alle Blöcke die gleiche Länge haben und einen einheitlichen Streckenwiderstand bieten. Weiterhin ist die maximale Geschwindigkeit im gesamten Netz gleich – auch bei der Überfahrt einer Weiche im abzweigenden Ast. Somit ergeben sich keine Nachteile für einen Zug im Nebengleis. Durch diese Vereinfachungen ist die spätere Fahrplankonstruktion leichter durchzuführen und es können Variablen ausgeschlossen werden, die keinen Einfluss auf einen Deadlock haben.

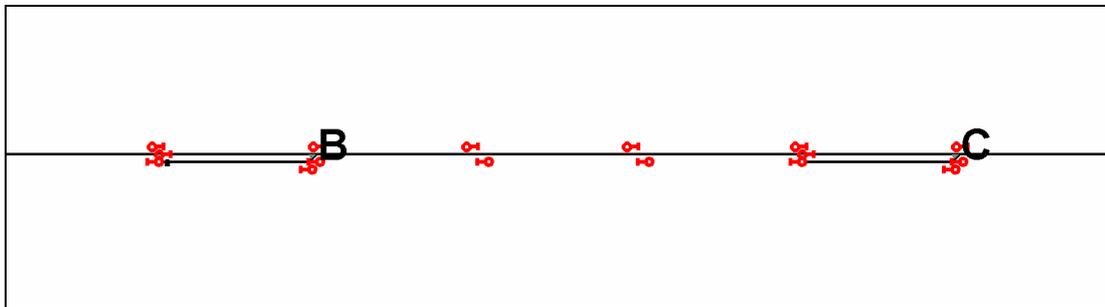
Bei der Fahrplankonstruktion werden zwei Modellzüge verwendet. Ein langsamer und ein schneller Modellzug. Diese Einteilung wird getroffen, um Überholungen zu generieren. Der schnelle Modellzug bekommt zudem eine höhere Priorität in der Simulation als der langsame Modellzug. Ist ein schneller Zug verspätet, wird er im Simulationsdurchlauf bevorzugt behandelt. Dies kann zur Folge haben, dass er einen langsamen Zug, der pünktlich ist, überholt. Bis auf den Geschwindigkeitsunterschied sind beide Züge gleich parametrisiert.

Im Nachfolgenden wird jedes einzelne Szenario mit genauer Infrastruktur und dem jeweiligen Fahrplan beschrieben und begründet.

---

### 3.2 Szenario 1

Deadlocks können nur auf Infrastrukturabschnitten entstehen, die im Zweirichtungsbetrieb den Zugverkehr abwickeln. Aus dieser Eigenschaft leitet sich das erste Szenario ab, welches den klassischen Fall einer eingleisigen Strecke betrachtet. Die Bahnhöfe sind zum Überholen und zum Kreuzen geeignet. Dies bedeutet, dass auch die Gleise im Bahnhof technisch so ausgestattet sind, dass sie in beiden Richtungen befahren werden können. Jeder Bahnhof hat zwei Gleise mit jeweils einem Block. Alle Bahnhöfe sind in ihrer Konfiguration gleich. Zwischen den Bahnhöfen existieren für jede Richtung drei Blöcke. Abbildung 5 stellt einen Teil der Strecke zwischen zwei Bahnhöfen dar.



**Abbildung 5: Teilinfrastrukturansicht in RailSys, Szenario 1**

Die gesamte Strecke führt vom Bahnhof A über B, C, D und endet im Bahnhof E. Bahnhof E wird im Szenario 5 beschrieben. In ihm werden alle Szenarien zu einem Netz zusammengeführt.

Der Fahrplan sieht einen Zugmix von 33% langsamer Züge und 66% schneller Züge vor. Abbildung 6 zeigt einen Takt des Bildfahrplans. Die Taktlänge beträgt 1,5 Stunden. Danach wiederholt sich das Betriebsprogramm.

Die Kreuzungen der gleichartigen Modellzüge finden jeweils im Bahnhof C statt. Die Kreuzungen der nicht gleichartigen Modellzüge hingegen im Bahnhof B und D. Der konstruierte Fahrplan ist konfliktfrei. Anzumerken ist noch, dass die schnellen Züge im Bahnhof B nicht dasselbe Gleis benutzen, wie die langsamen Züge. Diese Gestaltung des Fahrplans kann eine bestimmte Situation verursachen, aus der ein Deadlock entsteht.

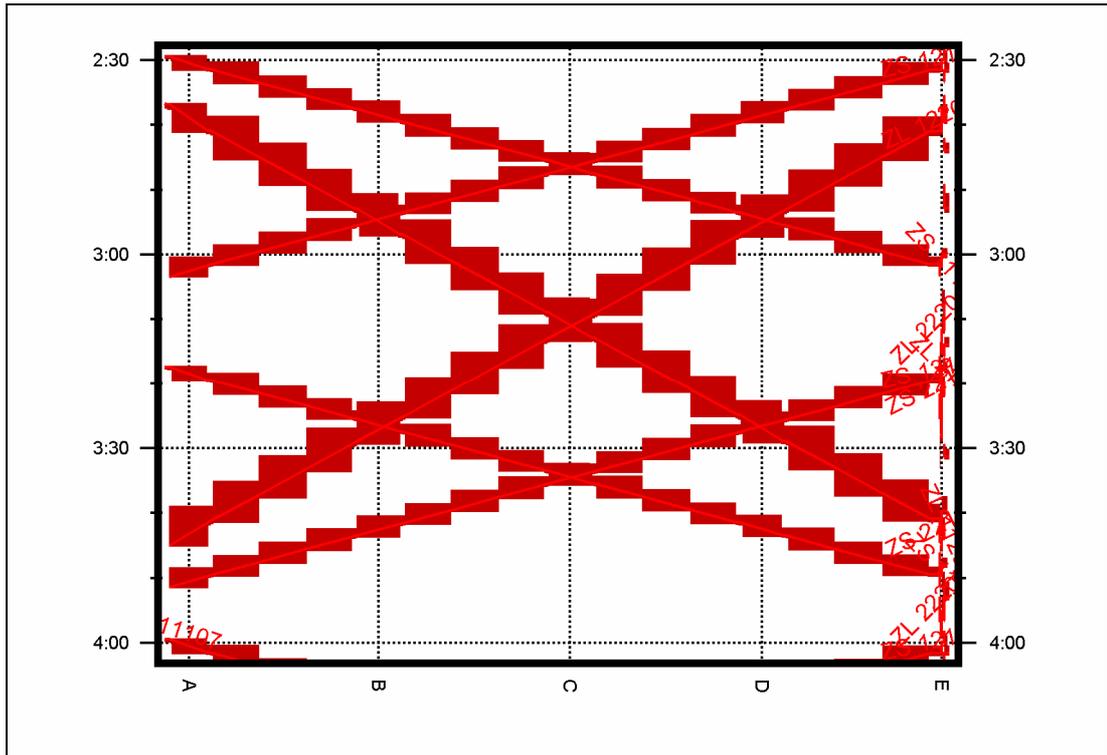


Abbildung 6: Bildfahrplanansicht in RailSys, ein Takt im Szenario 1

### 3.3 Szenario 2

In Szenario 1 wird angenommen, dass in den Bahnhöfen der Betrieb auf jedem Gleis in jeder Richtung stattfinden kann. Somit sind Überholungen möglich, aus denen Deadlock-Situationen entstehen können. In diesem Szenario soll hingegen der Zugverkehr in den Bahnhöfen im Einrichtungsbetrieb stattfinden. Das heißt, dass jedes der zwei Bahnhofsgleise nur für eine Fahrtrichtung ausgerüstet ist und das Überholen in einem Bahnhof nicht mehr möglich ist. Die Infrastruktur eines Kreuzungsbahnhofes sieht dann wie in Abbildung 7 dargestellt aus.

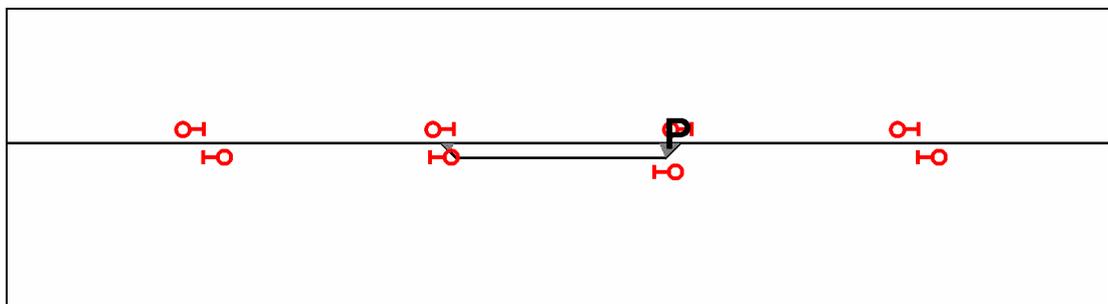
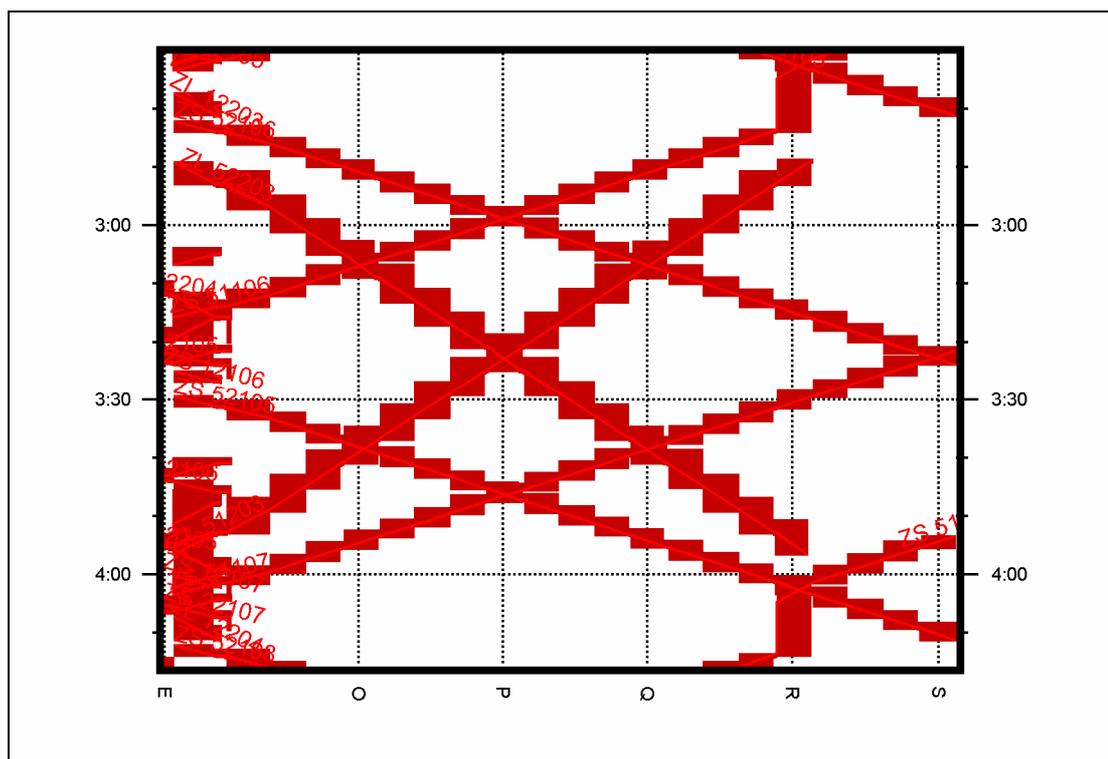


Abbildung 7: Teilinfrastrukturansicht in RailSys, Szenario 2

Die Begründung dieses Szenarios liegt genau im Einrichtungsbetrieb im Bahnhof. Es soll untersucht werden, ob hier trotz des Richtungsbetriebes in den Bahnhöfen Deadlocks entstehen können.

Der Aufbau der eingleisigen Strecke ist ähnlich der im Szenario 1. Sie unterscheiden sich in der Anzahl der Bahnhöfe. Ansonsten liegen auch hier drei Blöcke zwischen den Bahnhöfen. Die Strecke beginnt im Bahnhof S und führt über die Bahnhöfe R, Q, P, O in den Bahnhof E.

Der Aufbau des Bildfahrplans ist ebenfalls dem des Szenarios 1 ähnlich. Der Zugmix besteht aus 33% langsamen und 66% schnellen Zügen. Die langsamen Züge enden bzw. beginnen schon im Bahnhof R. Die Taktlänge beträgt auch hier 1,5 Stunden. In Abbildung 8 wird der Bildfahrplan für einen Takt des Szenarios 2 grafisch abgebildet.



**Abbildung 8: Bildfahrplanansicht in RailSys, ein Takt im Szenario 2**

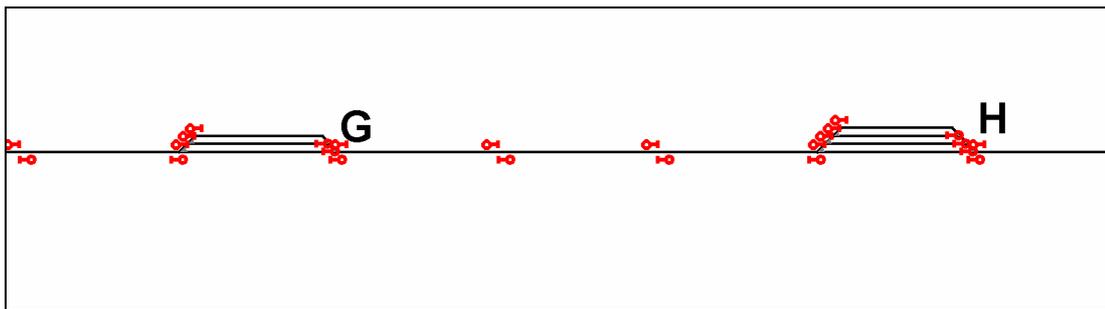
### 3.4 Szenario 3

Wie die Szenarien 1 und 2 befasst sich Szenario 3 ebenfalls mit einer eingleisigen Strecke. Auch hier wird ein Augenmerk auf die Kapazität der Bahnhöfe gelegt. In

diesem Fall jedoch eine Erhöhung der Kapazität. Die Bahnhöfe G und I besitzen jeweils drei parallele Gleise, die jeweils über einen Block verfügen. Die Kapazität im Bahnhof H wird durch vier parallele Gleise erzeugt. Bahnhof K ist zweigleisig und dient nur zum Ein- bzw. Ausfädeln der Züge beim Befahren der Strecke. Alle Gleise sind im Zweirichtungsbetrieb befahrbar.

In diesem Szenario ist zum einen die Kapazitätserhöhung in den Bahnhöfen, bezogen auf die Deadlock-Anfälligkeit, zu untersuchen. Zum anderen soll hier der Aufwand von Vermeidungsmaßnahmen von Deadlocks bei größerer Infrastruktur abgeschätzt werden.

Zwischen den Bahnhöfen existiert jeweils eine eingleisige Strecke mit drei Blöcken. Die Strecke beginnt im Bahnhof K, führt über die Bahnhöfe I, H, G und endet im Bahnhof E. Abbildung 9 stellt einen Teil der Streckeninfrastruktur dar.



**Abbildung 9: Teilinfrastrukturansicht in RailSys, Szenario 3**

Der konfliktfreie Fahrplan besteht aus einem Zugmix mit 33% langsamer Züge und 66% schneller Züge und weiterhin mit einer Taktzeit von 1,5 Stunden. Abbildung 10 zeigt einen Takt des Bildfahrplans.

Weiterhin ist in einer Richtung im Bahnhof I, wie im Szenario 1, das Stammgleis zur Durchfahrt für die langsamen Züge ein anderes, als für die schnellen Züge. Diese Fahrplangestaltung ist für die bessere Vergleichbarkeit mit dem Szenario 1 integriert.

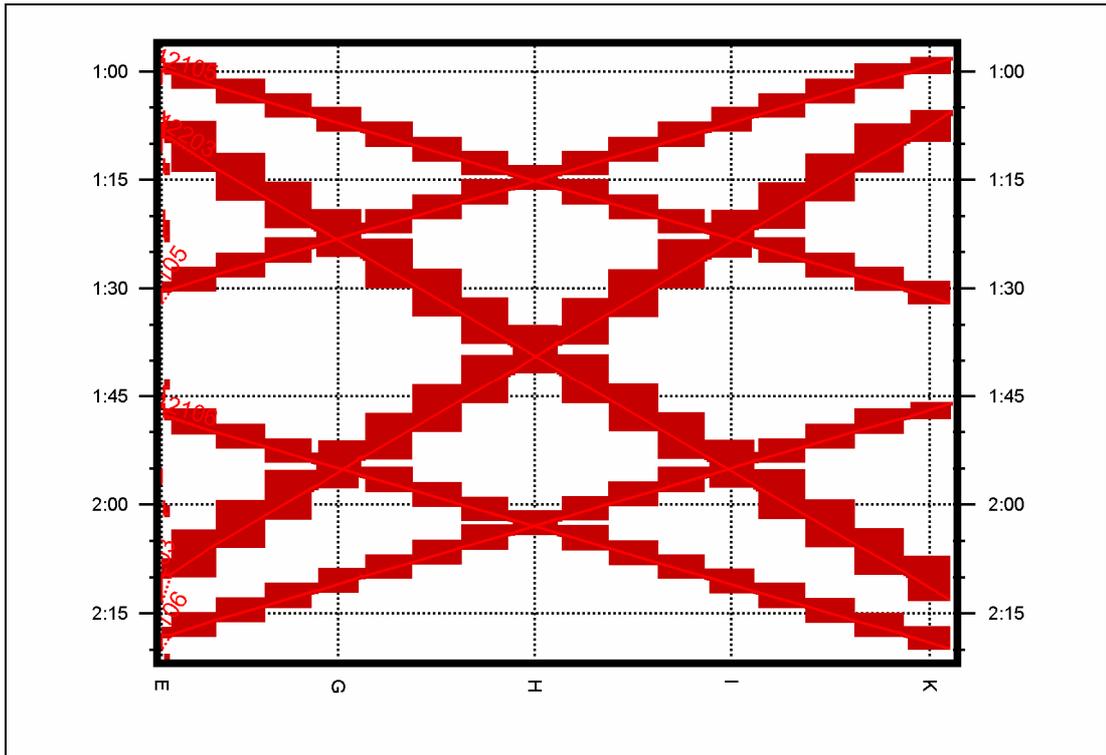


Abbildung 10: Bildfahrplanansicht in RailSys, ein Takt in Szenario 3

### 3.5 Szenario 4

Das Szenario 4 betrachtet eine zweigleisige Strecke mit zwei Überholungsmöglichkeiten in den Bahnhöfen M und N. Im Bahnhof N existieren zwei parallele Gleise mit jeweils einem Block. In ihm finden planmäßige Überholungen statt. Abbildung 11 zeigt den Bahnhof N in RailSys.

In diesem Szenario soll untersucht werden, in welcher der beiden Konstellationen Deadlocks auftreten können und ob dies einfach verhindert werden kann.

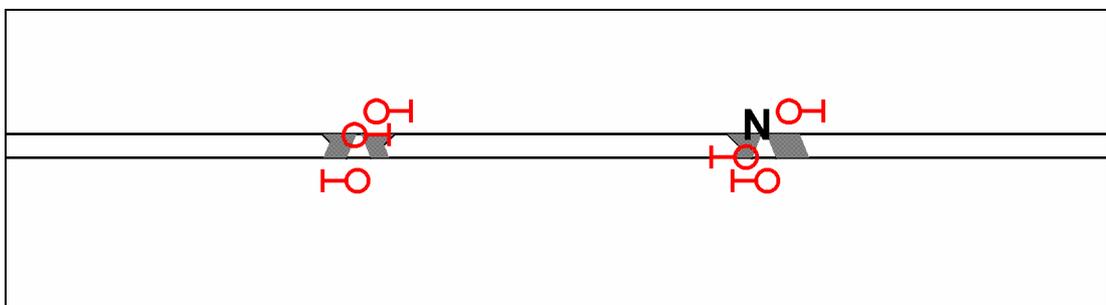
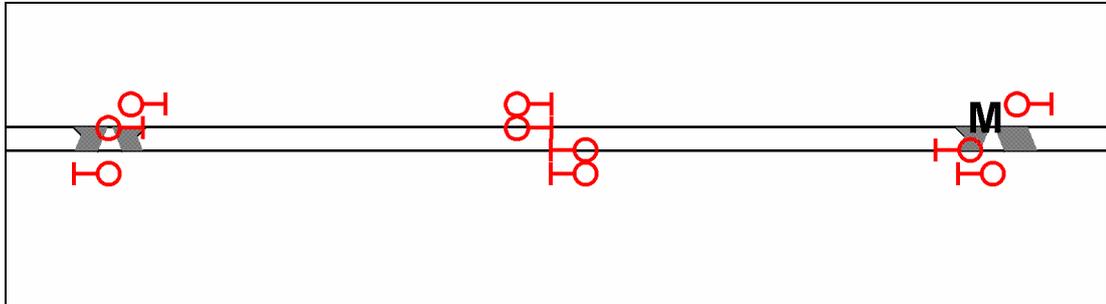


Abbildung 11: Teilinfrastrukturansicht in RailSys, Szenario 4, Bahnhof N

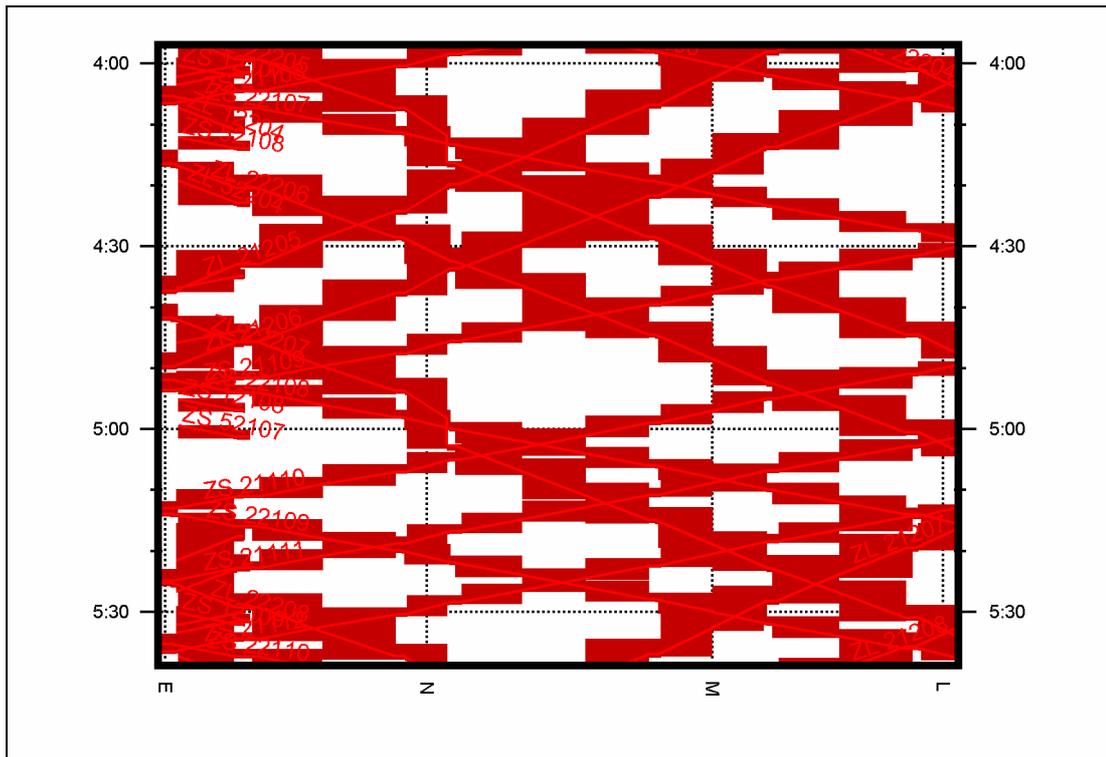
Bahnhof M bildet seine Kapazität auch aus zwei parallelen Gleisen, jedoch jeweils mit zwei aufeinanderfolgenden Blöcken. Ein Wechsel des Gleises ist hier nur bei der Einfahrt in den Bahnhof möglich und nicht nach dem ersten Block. Abbildung 12 zeigt den Bahnhof M in RailSys.



**Abbildung 12: Teilinfrastrukturansicht in RailSys, Szenario 4, Bahnhof M**

Die Strecke beginnt im Bahnhof L, führt über die Bahnhöfe M und N und mündet im Bahnhof E. Zwischen den Bahnhöfen M und N ist die Strecke im Einrichtungsbetrieb befahrbar. Für jede Richtung existieren drei Blöcke. Zwischen den Bahnhöfen E und N bzw. zwischen den Bahnhöfen M und L sind jeweils zwei Blöcke installiert.

Der Fahrplan, der diesem Szenario zugrunde liegt, fertigt doppelt so viele Züge in einem Takt ab, wie eine der eingleisigen Strecken. Die Taktdauer beträgt 1,5 Stunden. Weiterhin enden und beginnen die Züge nicht im Bahnhof E, sondern durchqueren ihn. Aus diesem Grund ist der Fahrplan nicht symmetrisch aufgebaut, wie auf den eingleisigen Strecken. Der Zugmix ist außerdem nicht gleichmäßig auf Hin- und Rückrichtung verteilt. So fahren 50% langsamer Züge vom Bahnhof E nach L aber nur 33% in die Gegenrichtung. Die Zuganzahl ist für beide Richtungen gleich. Abbildung 13 stellt den konfliktfreien Bildfahrplan für einen Takt des Szenarios 4 dar.

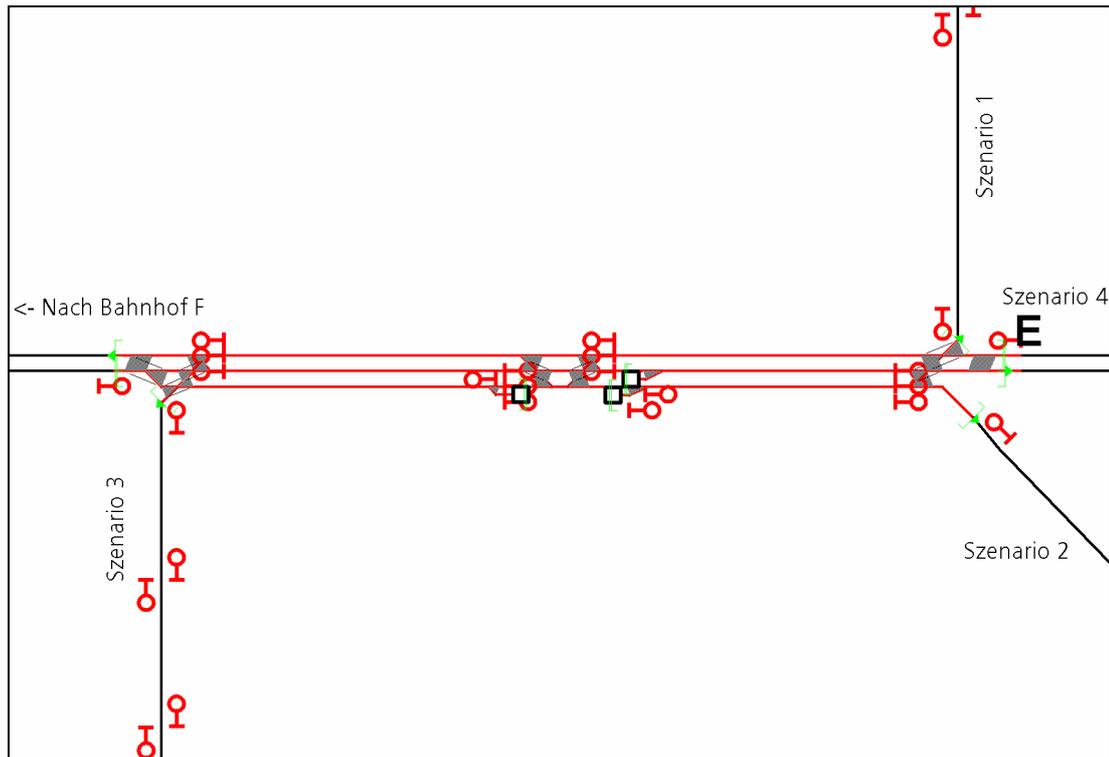


**Abbildung 13: Bildfahrplanansicht in RailSys, ein Takt in Szenario 4**

### 3.6 Szenario 5

In Szenario 5 werden ein kurzer zweigleisiger Streckenabschnitt und ein Bahnhof betrachtet. Der Bahnhof E vereinigt alle Verkehrsströme, der zuvor beschriebenen Szenarien in sich. Abbildung 14 zeigt die Darstellung des Bahnhofs E in RailSys. Die Züge der Szenaren 1, 2 und 3 enden bzw. beginnen alle im Bahnhof E. Die Züge des Szenarios 4 fahren durch den Bahnhof E durch. Sie fahren nach bzw. kommen von der westlich angeordneten zweigleisigen Strecke zum Bahnhof F. Diese Strecke hat die Aufgabe den Bahnhof E mit Zügen zu belasten, bzw. das Zugaufkommen abzuführen. Im Bahnhof F am Netzrand ist das Befahren aller Gleise in zwei Richtungen möglich.

Der infrastrukturelle Aufbau des Bahnhofs E besteht aus 6 Blöcken für jede Richtung, womit sich drei parallele Gleise ergeben, die jeweils zwei Blöcke besitzen. Vor bzw. nach jedem Block besteht ein Weichenbereich, der eine Vielzahl von Fahrwegen durch den Bahnhof E ermöglicht. Alle Gleise sind im Zweirichtungsbetrieb befahrbar.



**Abbildung 14: Teilinfrastrukturansicht in RailSys, Szenario 5**

Die Infrastruktur im Bahnhof E bietet eine Besonderheit. Für die Züge, die im Bahnhof E beginnen, wurden Einfahrgleise installiert. Diese Einfahrgleise haben die Aufgabe die korrekte Blockbelegung im Bahnhof, bei der Entstehung eines Zuges, darzustellen. Ein Problem in der Simulation ist, dass beginnende und endende Züge aus dem „Nichts“ entstehen bzw. „sich in Luft auflösen“. Beim Entstehen eines Zuges auf einem der rechten Blöcke Richtung Szenario 1 würde der Zug dann den gesamten Kreuzungsbereich zwischen den Blöcken belegen. Durch die Einfahrgleise wird das Problem umgangen und die falsche Abbildung in der Simulation berichtigt. Der Zug belegt nun nur noch den relevanten Block.

Endet ein Zug im Bahnhof E wird er beim Erreichen seines Zielgleises sofort aus dem System genommen. Hierdurch wird eine unbegrenzte Infrastruktur im Bahnhof angenommen. Diese Annahme trägt zur Vermeidung von Deadlocks in Bahnhöfen bei, da das Konfliktpotential aus dem System gelöscht wird.

Um dennoch Deadlocks in diesem Bahnhof zu betrachten, wird eine weitere Fahrplanvariante für dieses Szenario angenommen. In einem ersten Schritt wird ein Teilnetz, das nur aus den Bahnhöfen E und F besteht, erzeugt. Das Konfliktpotenzial bilden Züge, die den Bahnhof E durchquerenden und zwar in

den Relationen  $F - E - N$ ,  $N - E - F$  und  $D - E - F$ . Hierdurch ergeben sich kreuzende Verkehrsströme.

### **3.7 Infrastrukturnetz der Simulation**

Wie aus der Beschreibung des Szenarios 5 hervorgeht, werden alle Strecken zu einem Infrastrukturnetz zusammengefasst. Im Anhang A wird das Netz in seiner Gesamtheit dargestellt.

#### **3.7.1 Erzeugen von gestörten Fahrplänen**

Der Fahrplan, der der Simulation zu Grunde liegt, ist, wie bereits erwähnt, konfliktfrei. Dieses kann durch den störungsfreien Ablauf der Fahrplansimulation in RailSys bewiesen werden. D.h. es entstehen keine Folgeverspätungen während der Simulation.

Um Deadlocks zu erzeugen, wird der konfliktfreie Fahrplan mit Störungen überlagert. In RailSys besteht die Möglichkeit für jeden Zug bzw. für jede Zuggruppe zu definieren, an welcher Station (Bahnhof) eine Verspätung auftreten soll. Die Verspätungen werden in RailSys entweder durch eine negative Exponentialverteilung oder eine empirische Verteilung abgebildet. Durch die Definition dieser Verteilungen wird entschieden, welcher Zug wie viele Verspätungsminuten erhält.

Hier wird eine negative Exponentialverteilung gewählt, mit den Angaben über Anteil der Züge, die verspätet sein sollen, die mittlere Verspätung dieser Züge und die maximale Verspätung der Züge. Diese Verteilung wurde gewählt, weil mit ihr eine gröbere Abschätzung möglich ist. Bei der empirischen Verteilung hätten die Verspätungen in Klassen eingeteilt und jeder Klasse eine Zugmenge zugewiesen werden müssen. Davon abgesehen ist die Entscheidung über die Verteilung egal, da hier kein reelles Szenario abgebildet werden soll. Wichtig ist, dass Deadlocks entstehen.

An welchem Bahnhof die Verspätungen entstehen, kann entweder vom Benutzer oder gleich verteilt über alle Bahnhöfe definiert werden. Da die Funktion der Gleichverteilung in RailSys nicht fehlerfrei funktioniert, werden die Verspätungen so über das Netz verteilt, dass es der Methode der Gleichverteilung entspricht.

---

Die Verteilung sollte zunächst gleichmäßig sein, da keine anderen Daten zur Verfügung stehen.

Bleibt nun noch zu bestimmen, mit welcher Art von Verspätungen die Züge belastet werden. Es stehen drei zur Auswahl:

- Abfahrtverspätung
- Einbruchsverspätung
- Haltezeitverlängerung

Da der hier betrachtete Fahrplan so konstruiert ist, dass die Züge in den Bahnhöfen nur in seltenen Fällen halten, ist es sinnvoll, sich auf die Einbruchsverspätungen zu beschränken. Da jeder Zug in das System einbricht, kann somit der gesamte Fahrplan gleichmäßig mit Störungen überlagert werden.

Nun muss noch definiert werden, welche Züge mit Einbruchsverspätungen belastet werden. Es existieren im Fahrplan zwei Modellzüge. Ein langsamer und ein schneller Modellzug. Um Deadlocks zu erzeugen, ist es sinnvoll die schnellen Modellzüge zu verspäten, da sie höhere Priorität als die langsamen Modellzüge besitzen und diese Priorität bei Verspätungen schneller wächst. Somit beginnen die schnellen Züge bei einer Verspätung die langsamen zu überholen, was vermutlich zu einem Deadlock führen kann.

Da jetzt alle Parameter für den gestörten Fahrplan definiert sind, kann er nun erzeugt werden. Es werden 100 gestörte Fahrpläne erzeugt, die alle unterschiedlich sind, jedoch auf den gleichen Gegebenheiten beruhen. Jeder dieser Fahrpläne stellt einen Betriebstag dar.

### **3.7.2 Erzeugen von Teilnetzen**

RailSys bietet die Möglichkeit zur Bildung sogenannter Teilnetze. Für die Untersuchung von Deadlocks kann es zweckmäßig sein, nicht nur das gesamte Netz zu betrachten. Denn hier kann das Problem auftreten, dass immer nur eine Variante von Deadlocks auftritt, nämlich die, die sehr schnell und einfach entsteht.

---

Um dies zu verhindern, sind die einzelnen Szenarien so gewählt, dass sie voraussichtlich nur für einen Typ von Deadlocks eine große Anfälligkeit aufweisen. Damit ein Deadlock, der zuvor in einem anderen Abschnitt des Infrastrukturnetzes aufgetreten ist, nicht einen anderen Deadlock überdeckt, der im relevanten Abschnitt entstehen soll, wird für jedes Szenario ein Teilnetz erstellt.

Jedes Teilnetz besteht, bezogen auf die Infrastruktur, aus den relevanten Bahnhöfen für dieses Szenario. Für Szenario 1 bedeutet dies z.B., dass die Bahnhöfe A, B, C, D und E betrachtet werden. In Bahnhof E verkehren dann jedoch nur die Züge, die auch jeweils die anderen eben erwähnten Bahnhöfe befahren.

Die Teilnetze bedienen sich derselben gestörten Fahrpläne wie das Gesamtnetz. Jedoch sind die Zugabfolgen der jeweils betrachteten Teilstrecke und des Gesamtnetzes nicht identisch. Dies liegt an den Beeinflussungen der Züge, die im Teilnetz nicht mehr betrachtet werden, weil sie zu einem anderen Teil des Infrastrukturnetzes gehören. Somit ist es egal, ob dieselben gestörten Fahrpläne Anwendung finden oder nicht. Es beschert nur weitere Situationen, in denen Deadlocks auftreten können.

---

---

## 4 Deadlock-Betrachtung

In diesem Kapitel soll dargestellt werden, in welchen Situationen Deadlocks entstehen und welche Bedingungen dazu beitragen. Es werden hierzu die definierten Szenarien herangezogen und mit ihnen Deadlocks produziert. Vorausgehend dazu wird auf Voraussetzungen für einen Deadlock eingegangen. Hierzu werden Regeln definiert, nach welchen ein System Deadlock fähig ist.

### 4.1 Voraussetzungen für einen Deadlock

Damit ein Deadlock in einem System entstehen kann, muss gewährleistet sein, dass das System Deadlock fähig ist. Hierzu werden nach Bruns [BRU83] vier Voraussetzungen definiert, die das System erfüllen muss, um einen Deadlock zu produzieren. Im Nachfolgenden sind diese Voraussetzungen, bezogen auf das System Bahn nach Pachtl, aufgeführt:

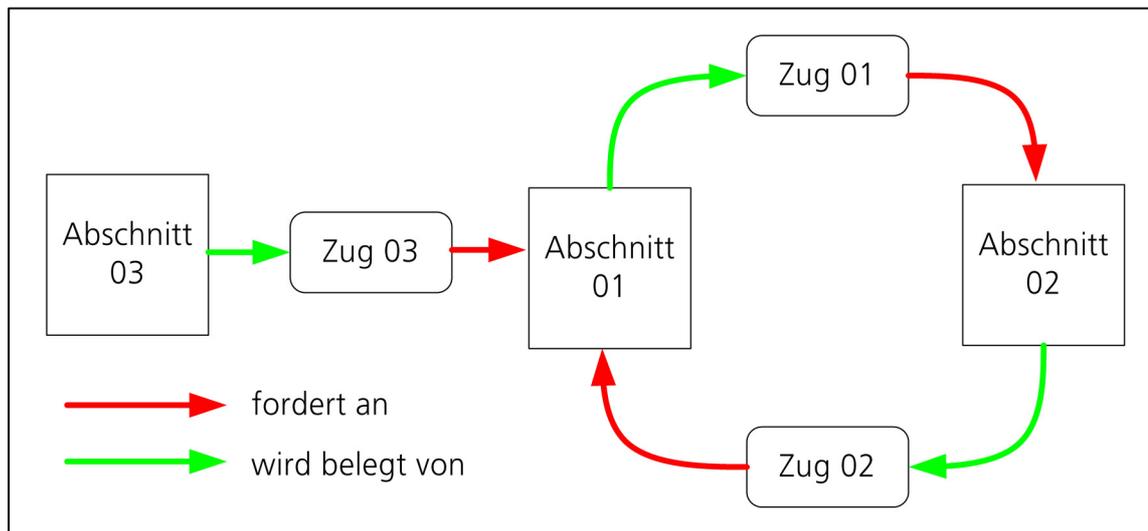
1. Ein zu belegender Gleisabschnitt (Block) kann nur von einem Zug belegt werden. Nach der Belegung durch den ersten Zug kann kein zweiter Zug in den Block einfahren.
2. Ein Zug kann, während er einen Block belegt, weitere Blöcke anfordern, die er belegen möchte und darauf warten.
3. Ein Zug gibt einen Block erst wieder frei, wenn er einen anderen belegt hat.
4. Die Bildung geschlossener Warteketten ist möglich, d.h. mehrere Züge warten auf die Zuteilung eines angeforderten Blockes, die jeweils von anderen Zügen belegt sind.

[PAC93]

Die Voraussetzungen eins bis drei ergeben sich aus dem Zugsicherungsverfahren und aus den Systemeigenschaften der Eisenbahn. Das Bilden von Warteketten aus der vierten Voraussetzung resultiert aus den vorangehenden Voraussetzungen.

---

Den Beweis zeigt das Beispiel in Abbildung 15. Hier wird die Wartekette für die in Abbildung 2 dargestellte Situation visualisiert. Die Züge 01 und 02 warten gegenseitig auf den Abschnitt, den jeweils der andere Zug belegt. Somit erfüllt das System Bahn alle Voraussetzungen eines Deadlock-fähigen Systems.



**Abbildung 15: Wartekette im System Bahn – Deadlock-Situation**

Die in dieser Arbeit betrachtete synchrone Simulation erfüllt ebenfalls die Anforderungen eines Deadlock-fähigen Systems. Besonderes Augenmerk liegt hierbei auf der vierten Voraussetzung. Da eine synchrone Simulation immer auch ereignisorientiert ist, wird diese Bedingung erfüllt. Züge in der Simulation warten auf das Ereignis, den nächsten Block zu belegen. Somit können Warteketten entstehen.

Bei einer asynchronen Simulation ist dies nicht der Fall. Hier wird die vierte Voraussetzung nicht erfüllt. Warteketten können nicht entstehen, weil jeder Zuglauf als ganzes auf die Infrastruktur umgelegt wird. Somit steht ein einzulegender Zug nur im Konflikt mit den bereits umgelegten Zügen. Dieser Konflikt kann nur von dem aktuellen Zug beseitigt werden.

Nur synchrone Simulationen sind in der Lage Deadlocks zu produzieren. Asynchrone Simulationen sind dagegen nicht in der Lage, weil sie nicht alle Voraussetzungen erfüllen.

---

## 4.2 Randbedingungen für einen Deadlock

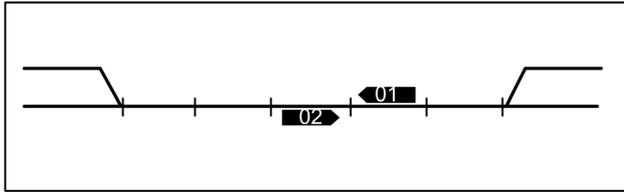
Ein Deadlock entsteht immer dann, wenn sich mindestens zwei Züge gegenüberstehen. Jeder Zug steht dabei in seinem Block und wartet auf die Weiterfahrt, die an Bedingungen geknüpft ist. Das Gegenüberstehen zweier Züge beschreibt den Kern einer Deadlock-Situation. Damit diese Situation entsteht, muss die Strecke, in der sich die beiden belegten Blöcke befinden, in beiden Richtungen befahrbar sein. Als erste Randbedingung gilt also, dass die betroffenen Streckenabschnitte im Zweirichtungsbetrieb betrieben werden müssen. Somit fallen zweigleisige Strecken, die im Einrichtungsbetrieb betrieben werden, aus der Betrachtung raus. Infrastrukturelemente im Zweirichtungsbetrieb können sein:

- Eingleisige Strecken
- Bahnhöfe
- Zweigleisige Strecken im Zweirichtungsbetrieb

Eine zweite Randbedingung ist, dass der durchzuführende Zugbetrieb eine stochastische Komponente beinhaltet. Je stochastischer, d.h. zufallsabhängiger der Betriebsablauf stattfindet, desto größer ist die Wahrscheinlichkeit eines Deadlocks im System. In dieser Arbeit wird dies durch die gestörten Fahrpläne realisiert. [LAD01]

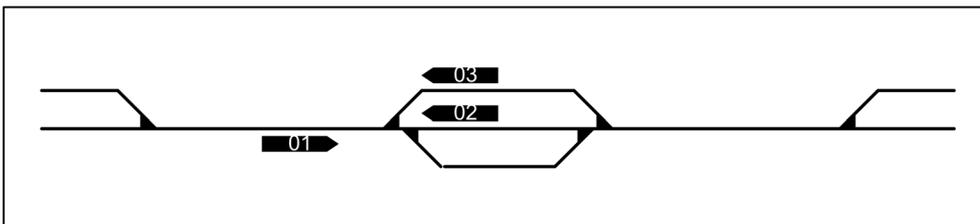
Als weitere Randbedingung kann die Kapazität, in Abhängigkeit vom Verkehrsaufkommen, der betrachteten Infrastruktur angegeben werden. Bei der Annahme einer unbegrenzten Kapazität ist ein Entstehen von Deadlocks allerdings nicht ausgeschlossen. Ein einfacher Algorithmus zur Vermeidung von trivialen Deadlocks, die einem kleinen „Deadlock gefährdeten Bereich“ zugrunde liegen, ist auch hier die Voraussetzung. Abbildung 16 stellt einen trivialen Deadlock dar. In diesem Beispiel sind zwei Züge auf einen eingleisigen Streckenabschnitt eingefahren. Dieses ist die einfachste Form eines Deadlocks, die durch eine einfache Vorschrift verhindert werden kann.

---



**Abbildung 16: Trivialer Deadlock**

Mit Hilfe der Kapazität können jedoch nichttriviale Deadlocks vermieden werden. Ist die Kapazität z.B. in einem Knoten (Bahnhof) größer als das Verkehrsaufkommen, das diesen Knoten befahren möchte oder befährt, kann durch dispositive Maßnahmen ein Deadlock verhindert werden, indem z.B. ein einfahrender Zug in einen freien Abschnitt umgeleitet wird. Die Situation aus Abbildung 2 hätte nicht zu einem Deadlock geführt, wenn die Kapazität des Infrastrukturknotens größer gewesen wäre. In Abbildung 17 wird dies umgesetzt. Zug 01 kann nun in den Knoten einfahren.



**Abbildung 17: Deadlock-Vermeidung durch Kapazitätsausgleich**

Da die Kapazität jedoch nicht in unbegrenzter Höhe zur Verfügung steht, ist die Vermeidung durch Kapazitätserhöhung nicht realisierbar. Aufgrund der hohen Kosten im Infrastrukturbau im System Bahn, kann z.B. die Kapazität eines Knotens nicht an ein mögliches maximales Verkehrsaufkommen, wie im vorangegangenen Beispiel beschrieben, angepasst werden, dass möglicherweise nie eintritt. Dieses wäre eine unwirtschaftliche Methode Deadlocks zu vermeiden. Gleiches gilt für den Einrichtungsbetrieb, der auch eine höhere Kapazität zur Verfügung stellt, die möglicherweise nicht benötigt wird.

Es kann festgehalten werden, dass das Vermeiden von Deadlocks durch einen Kapazitätsausgleich nicht unbedingt sinnvoll ist. Es sollte aber an dieser Stelle vermerkt werden, dass die Kapazität eine wesentliche Rolle in der Deadlock-Behandlung spielt.

---

## 4.3 Analyse der Deadlock-Situationen

Um ein Verfahren zu entwickeln, das Deadlocks vermeidet, muss zuvor analysiert werden, in welchen Situationen Deadlocks entstehen. In den vorangegangenen Kapiteln wurde schon auf die Voraussetzungen und Randbedingungen eingegangen.

Weiterhin ist zu definieren, ob es unterschiedliche Arten von Deadlocks gibt. Hierzu kommt an dieser Stelle der Arbeit das erste Mal die Simulation in RailSys zum Einsatz. Die Simulation der erstellten Infrastruktur wird nun mit den erzeugten gestörten Fahrplänen gestartet und anhand dieser beobachtet, in welchen Situationen und an welchen Stellen Deadlocks entstehen.

Die unterschiedlichen Arten von Deadlocks, die dabei beobachtet wurden, sind im Nachfolgenden beschrieben.

### 4.3.1 Trivialer Deadlock

Durch das Fahren im festen Raumabstand ist gewährleistet, dass sich immer nur ein Zug in einem Block befindet. Das ist die Grundregel nach der das Simulationstool RailSys verfährt. Sind keine weiteren Regeln aktiv, kann es zu trivialem Deadlock kommen. Eine solche Situation ist in Abbildung 16 dargestellt. RailSys überprüft für jeden Zug nur, ob der nächste Block frei ist. Ist dies der Fall, fährt der Zug in den Block ein und belegt ihn. Kommt es zu der Situation, dass zwei Züge einen eingleisigen Streckenabschnitt in unterschiedlicher Richtung befahren, kommt es zur Deadlock-Situation. Dieser folgt zwangsläufig ein Deadlock.

Die Bedingung für einen trivialen Deadlock ist, dass der eingleisige Streckenabschnitt aus mindestens zwei Blöcken bestehen muss. Ist dies nicht der Fall, entsteht kein trivialer Deadlock, da nur ein Zug diesen einen Block belegen kann.

Die Szenarien 1, 2 und 3 sind typische Strecken auf denen triviale Deadlocks entstehen können, weil zwischen jedem Bahnhof ein eingleisiger Streckenabschnitt mit jeweils drei Blöcken liegt. Sobald vom konfliktfreien Fahrplan abgewichen wird, steigt die Wahrscheinlichkeit für einen solchen

---

Deadlock stark an, weil das konfliktfreie Kreuzen in den Bahnhöfen nicht mehr gewährleistet ist. Das Überholen von Zügen, was in dieser Arbeit bereits als kritisch eingestuft wurde, trägt bei dieser Art von Deadlock jedoch nicht zu einem Ansteigen der Wahrscheinlichkeit bei.

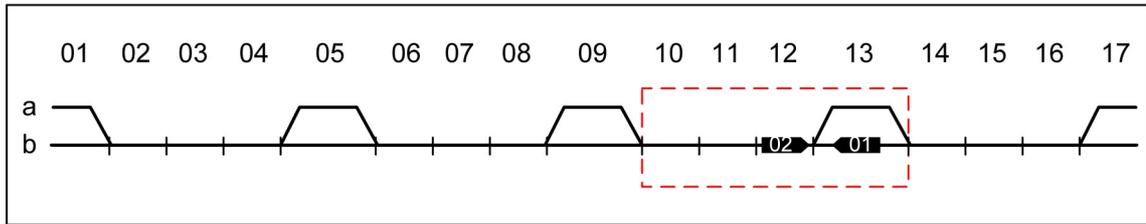
Die Bezeichnung „trivialer Deadlock“ leitet sich einerseits aus der einfach zu produzierenden Deadlock-Situation ab und zum anderen aus der einfachen Lösung dieses Problems. Um einen trivialen Deadlock zu vermeiden, kann eine einfache Regel aufgestellt werden. Diese Regel ist eine Voraussetzung für die weitere Betrachtung in dieser Arbeit, weil ohne die Vermeidung trivialer Deadlocks die Simulation zu schnell, nach einem solchen, abgebrochen wird. Dadurch entstehen nur wenige Situationen mit nichttrivialen Deadlocks, die es auch noch zu lokalisieren gilt. Diese Regel ist in RailSys implementiert und lautet wie folgt (die tatsächliche Umsetzung der Regel wird in Kapitel 5.1.1 beschrieben):

- Ein eingleisiger Streckenabschnitt darf von einem Zug nur befahren werden, wenn sich kein Zug auf diesem Streckenabschnitt befindet, der nicht der gleichen Richtung folgt.

Ein Streckenabschnitt ist ein Block oder eine Aneinanderreihung von Blöcken, die nur aufeinander folgen und nicht die Möglichkeit besteht innerhalb des Abschnittes ab- oder einzubiegen. Siehe Abbildung 16. Diese Regel ist identisch mit der Entscheidungsweise eines Disponenten in der Realität und keine vereinfachende Annahme.

Ein weiterer Deadlock mit zwei Zügen kann auch bei der Einfahrt in einen Bahnhof entstehen. Hierbei stehen sich der ausfahrende und der einfahrende Zug gegenüber. Dabei ist ein weiteres Bahnhofsgleis nicht belegt, sondern frei zur Einfahrt. Diese Situation kommt zustande, weil die Regelfahrwege der Züge im Bahnhof über dasselbe Gleis führen. Sind keine Methoden im Simulationsprogramm aktiv, kann der einfahrende Zug auf das freie Nebengleis nicht ausweichen. Es entsteht ein Deadlock, der in der Realität zu diesem Zeitpunkt nicht als Deadlock gewertet wird. Abbildung 18 stellt die Situation dar.

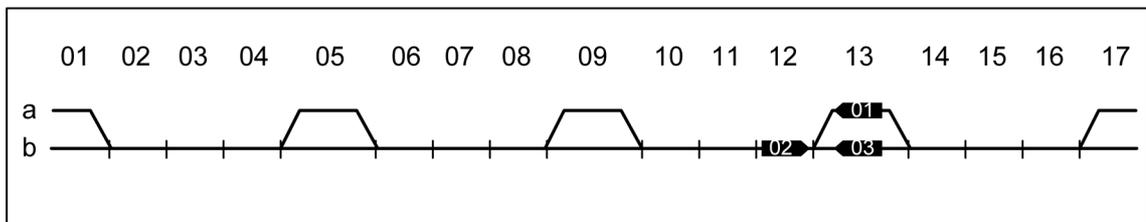
---



**Abbildung 18: Deadlock in der Simulation durch nicht aktivierte Methoden**

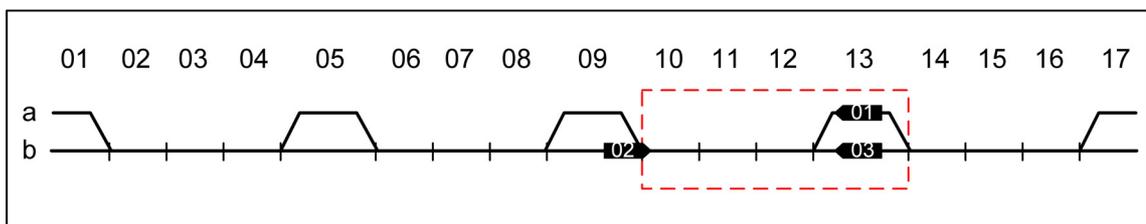
### 4.3.2 Deadlock mit drei Zügen (aus Parallelfahrt)

Die einfachste Form eines nichttrivialen Deadlocks, ist die Situation, in der sich drei Züge nach einem eingleisigen Abschnitt gegenüberstehen. Abbildung 19 stellt dies dar. Die Züge 01 und 03 belegen den Bahnhof in den Blöcken 13a und 13b. Zug 02 belegt Block 12b.



**Abbildung 19: Einfacher Deadlock mit drei Zügen**

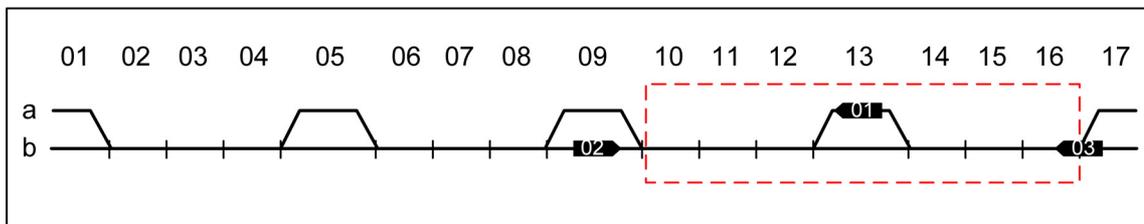
Dieser Deadlock kann aus unterschiedlichen Deadlock-Situationen entstanden sein. Die erste Deadlock-Situation wird in Abbildung 20 gezeigt. Die Züge 01 und 03 stehen im Bahnhof. Zug 02 fährt aus Block 09b in 10b in den „Deadlock gefährdeten Bereich“ ein und stellt somit die Deadlock-Situation her. Die 1. Regel aus Kapitel 4.3.1 tritt hier auch in Kraft, verhindert den Deadlock aber nicht.



**Abbildung 20: Deadlock-Situation mit drei Zügen, Parallelfahrt im Bahnhof 13**

In der zweiten Situation ist der „Deadlock gefährdete Bereich“ größer. Zug 03 fährt von Block 17a in 16b ein. Zu diesem Zeitpunkt existiert noch keine Deadlock-Situation. Geht man davon aus, dass der Deadlock, wie in Abbildung 19 dargestellt ist, eintreten wird, existiert die Deadlock-Situation in dem

Augenblick, in dem Zug 02 in den Block 10b einfährt. Dies ist aber an die Bedingung geknüpft, dass Zug 03 den Block 13b schneller erreicht als Zug 02. Würde Zug 02 diesen Block zuerst belegen, wäre der Deadlock zufällig vermieden.



**Abbildung 21: Deadlock-Situation mit drei Zügen, Überholung durch Zug 03**

Die Vermeidung des Deadlocks resultiert aus der wartenden Position des Zuges 01. Es stellt sich die Frage, auf was Zug 01 wartet. Auch hier gibt es mehrere Möglichkeiten:

1. Er wartet auf die Kreuzung des Zuges 02 und möchte anschließend weiterfahren.
2. Er wartet auf die Überholung durch Zug 03.
3. Zug 01 wartet auf die Überholung von Zug 03 und auf die Kreuzung von Zug 02.

Ein Fahrdienstleiter oder Disponent weiß, welche Züge er überholen lässt, und welche kreuzen. Weiterhin ist ihm bekannt, welche Züge er für ein bestimmtes Fahrmanöver warten lassen muss, damit keine Deadlock-Situation entsteht.

Die Information über die Fahrmanöver ist also wichtig, denn nicht alle drei Gründe des Wartens von Zug 01 führen zu einem Deadlock. Je nachdem, aus welchem Grund er wartet, leitet sich eine andere Vermeidung des Deadlocks ab.

Diese Art von Deadlock entsteht in der Simulation z.B. auf der Strecke in Szenario 1 im Bahnhof B. Hier ist der Fahrplan so konstruiert, dass ein schneller Zug einen langsamen überholen kann, wenn er verspätet ist. Der Deadlock entsteht dann aus der Situation, wie sie in Abbildung 21 dargestellt wird. In Szenario 2 kann ein solcher Deadlock nicht entstehen, weil die Bahnhöfe im Richtungsbetrieb den Verkehr abwickeln und somit das Einfahren zweier Züge, die in die gleiche Richtung fahren, nicht möglich ist. In Szenario 3 wäre es möglich, einen solchen

Deadlock zu produzieren. Jedoch ist hier die Kapazität der Bahnhöfe größer als in Szenario 1, so dass der Regelfahrweg keine Kreuzungen vorsieht.

Es bleibt festzuhalten, dass ein Deadlock mit drei Zügen nur dann entsteht, wenn zwei Zügen der gleichen Richtung auf unterschiedlichen Gleisen den Bahnhof belegen. Dies ist z.B. bei einer Überholung der Fall.

Zu diesem Zeitpunkt, in dem noch keine Dispositionsmaßnahmen in RailSys eingeschaltet sind, wird die Möglichkeit einer Überholung im Fahrplan konstruiert. Überholungen können jedoch auch vom Disponenten in RailSys vorgenommen werden. Diese Funktion wird in Kapitel 5.1.2 beschrieben.

### **4.3.3 Deadlock mit vier Zügen (aus Folgefahrt)**

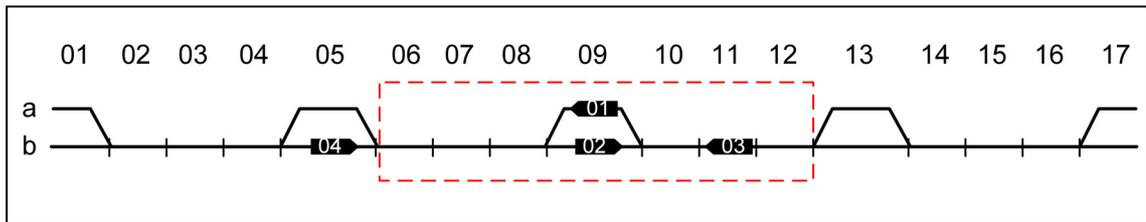
Wie in Kapitel 4.3.2 beschrieben wird, hätte der Deadlock zufällig vermieden werden können, wenn Zug 02 vor Zug 03 in den Block 13b eingefahren wäre. Zug 01 hätte dann aus dem Block 13a ausfahren und in 12b einfahren können.

Abbildung 22 stellt, aufbauend auf diese Situation (anderer Bahnhof), eine neue Situation dar, bei der vier Züge beteiligt sind. Dieser steht in Block 05b, kurz vor der Einfahrt in den „Deadlock gefährdeten Bereich“. Fährt er in den Block 06b ein, ist eine Deadlock-Situation entstanden.

In der hier betrachteten Simulation in RailSys kommt diese Art von Deadlocks in allen eingleisigen Szenarien vor. Besonders prädestiniert ist allerdings Szenario 2 für diese Deadlocks. Aufgrund des Einrichtungsbetriebes in den Bahnhöfen, welches das Ausweichen auf andere Gleise von vornherein ausschließt, ist ein Überholen der Züge untereinander nicht möglich. Ist z.B. ein schneller Zug verspätet und hat einen langsamen Zug vor sich, besteht nur die Möglichkeit dem vorausfahrenden Zug zu folgen. Weil der folgende Zug schneller ist, wird er nach bestimmter Zeit auf den langsamen Zug auffahren und ihm dann dicht folgen. Entsteht in der Gegenrichtung eine ähnliche Situation, fahren vier Züge auf einen Bahnhof zu, der nur eine Kapazität von zwei Zügen hat. So entsteht der Deadlock aufgrund des Fahrmanövers „Folgen“.

---

Da in den Szenarien 1 und 3 das Ausweichen auf andere Gleise noch nicht möglich ist, entstehen auch hier diese Art von Deadlocks. (Zu diesem Zeitpunkt der Arbeit verfahren die Züge nur auf ihren im Fahrplan angegebenen Gleisen.)



**Abbildung 22: Deadlock-Situation mit vier Zügen aus Folgefahrt**

Das Verhalten der Züge untereinander, „Kreuzen“ oder „Folgefahrt“, scheint bedeutend für das Entstehen von Deadlocks zu sein. Nimmt man den Schluss der Kapazität mit hinzu, lässt sich folgende Aussage treffen: Ein Zug darf einem anderen Zug über einen eingleisigen Abschnitt nur folgen, wenn die Kapazität der Infrastruktur dies zulässt.

Auch in diesem Beispiel kann ein Deadlock vermieden werden, wenn nicht Zug 04 in den Block 06b einfährt, sondern Zug 01 zuerst aus dem Block 09a ausfährt. Dieses Verhalten muss aber durch Regeln definiert werden.

#### 4.3.4 Deadlocks bei höherer Kapazität der Infrastruktur

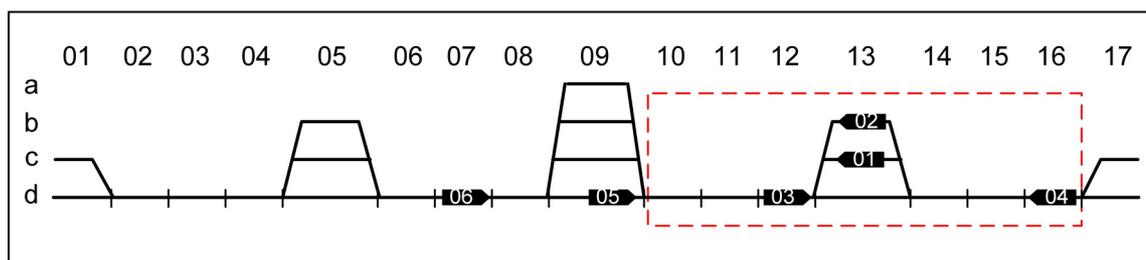
Bei dem jetzigen Stand der Arbeit ist es noch nicht möglich Deadlocks mit Hilfe der Simulation zu produzieren, die auf Infrastrukturen mit größerer Kapazität im Bahnhof entstehen. Dies ist der Fall, weil für die konfliktfreie Fahrplankonstruktion in einem Bahnhof auf eingleisigen Strecken meistens zwei Gleise ausreichen. Die übrigen Gleise werden, unter der Annahme der Konfliktfreiheit, nicht benötigt.

Sobald alle Gleise der Infrastruktur im Bahnhof genutzt werden, kann eine Auswertung dieser Deadlocks stattfinden. Hierzu ist es wichtig, eine weitere Dispositionsmethode einzuführen. Es handelt sich hierbei um die sogenannten Ausweichgleise. Ausweichgleise werden für jedes Gleis, das einen Regelfahrweg darstellt, definiert. Ist das Gleis des Regelfahrweges besetzt, wird ein Zug, der dieses befahren möchte, auf ein Ausweichgleis umgeleitet, das frei ist.

Durch das Definieren von Ausweichgleisen wird die Deadlock-Wahrscheinlichkeit voraussichtlich reduziert. Bei hohem Verkehrsaufkommen, als Folge von starken Verspätungen, können jedoch immer noch Deadlocks entstehen.

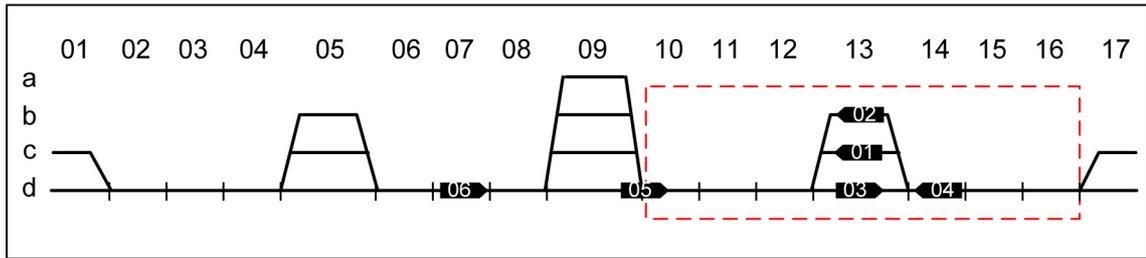
Bei der Definition von Ausweichgleisen in Bahnhöfen mit nur zwei Gleisen kann es jedoch auch zu einer Erhöhung der Wahrscheinlichkeit für Deadlocks führen. Denn sobald Züge der gleichen Richtung in einen Bahnhof einfahren können, besteht auch immer die Möglichkeit der Überholung. Im Nachfolgenden wird auf Beispiele eingegangen, die z. B. auf der Szenariostrecke 3 entstanden sind, beim Aktivieren der Ausweichgleise.

Wie oben erwähnt, wird die Anzahl der Deadlocks bei ausreichender Infrastruktur wahrscheinlich abnehmen. Die Deadlocks, die jetzt noch auftreten, sind auf den ersten Blick umfangreicher als die bislang betrachteten. Abbildung 23 zeigt ein erstes Beispiel für einen Deadlock bei größerer Kapazität der Infrastruktur.



**Abbildung 23: Bevorstehende Deadlock-Situation bei größerer Kapazität**

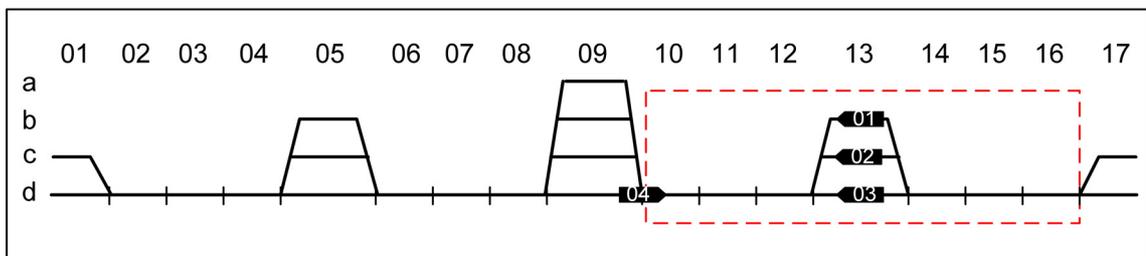
Zug 01 und Zug 02 sind zwei langsame Züge mit geringer Priorität. Beide stehen im Abschnitt 13, der jetzt nur noch Kapazität für einen zusätzlichen Zug hat. Sie warten jeweils auf die Überholung bzw. Kreuzung durch die Züge 04 und 03, die beide in den freien Abschnitt 13d einfahren möchten, um ihr Fahrmanöver durchzuführen. In diesem Fall wird Zug 03 den Abschnitt schneller erreichen. Abbildung 24 zeigt das Entstehen der Deadlock-Situation. Mit Einfahrt von Zug 05 auf den eingleisigen Streckenabschnitt in Block 10d wird diese Deadlock-Situation hergestellt.



**Abbildung 24: Deadlock-Situation bei größerer Kapazität durch Folgefahrt**

Prinzipiell ähnelt sich dieser Deadlock mit den voranbeschriebenen, mit dem Unterschied, dass mehr Züge benötigt werden, um den Deadlock zu erzeugen. Lässt man Zug 02 weg und das Gleis 13b außer Acht, ergibt sich ein Deadlock mit vier Zügen, wie er in Kapitel 4.3.3 beschrieben wurde.

Ein weiteres Beispiel zeigt Abbildung 25. Hier wird auf der Infrastruktur in Szenario 3 ein Deadlock mit nur vier Zügen im Gegensatz zu Abbildung 24 mit fünf Zügen erzeugt. Es werden nur vier Züge benötigt, weil sie alle den Bahnhof in Abschnitt 13 in die gleiche Richtung befahren. In diesem Fall ist Zug 01 ein langsamer Zug, der von Zug 02 (schneller Zug) überholt wird. Zug 03 hat soviel Verspätung, dass er beide Züge überholen darf. Die Deadlock-Situation wird erzeugt, indem Zug 04 in den Block 10d einfährt.



**Abbildung 25: Deadlock-Situation bei größerer Kapazität durch Parallelfahrt**

Auch bei diesem Beispiel gilt, würde man einen Zug, z.B. Zug 01, weglassen und Gleis 13b nicht betrachten, entstünde ein Deadlock, wie in Kapitel 4.3.2 beschrieben ist. Hieraus lässt sich ableiten, wie viele Züge, unter Berücksichtigung der Kapazität der Infrastruktur, benötigt werden, um einen Deadlock zu erzeugen. Es gilt:

Die Anzahl der benötigten Züge für einen Deadlock in einem Bahnhof ist die Anzahl der parallelen Gleise plus 1, wenn alle Züge im Bahnhof dieselbe Fahrtrichtung haben. Oder die Anzahl der Gleise plus 2, wenn die Züge im

Bahnhof nicht dieselbe Fahrtrichtung haben. Die Fahrtrichtung wird durch das Fahrmanöver der Züge im Bahnhof bestimmt.

Dieser Zusammenhang kann voraussichtlich für die Entwicklung einer Methode zur Vermeidung von Deadlocks verwendet werden.

Nochmals zu erwähnen ist, dass die Definition von Ausweichgleisen in Szenario 3 zu einer Verringerung der Deadlock-Wahrscheinlichkeit führt. Hierbei trat die Deadlock-Situation mit aus Abbildung 24 wesentlich häufiger auf, als die Deadlock-Situation in Abbildung 25. Dies liegt daran, dass die Verspätungen der Züge so verteilt werden müssen, dass eine Überholung mit drei Zügen in derselben Fahrtrichtung entsteht. Ein solcher Deadlock ist theoretisch auch im Bahnhof H möglich, jedoch in der Simulation nicht vorgekommen.

#### **4.3.5 Deadlocks in Bahnhöfen mit Doppelblock**

Deadlocks entstehen nicht nur auf Grund der Eingleisigkeit von Strecken, sondern auch in Bahnhöfen, in denen die Gleise in beiden Richtungen befahrbar sind. In den Szenarien 5 und 4 werden solche Bahnhöfe dargestellt. Bahnhof E aus Szenario 5 bietet hierbei unterschiedlichste Fahrtmöglichkeiten für die ihn durchquerenden Züge. Bahnhof M bietet dagegen nur vier Fahrwege.

Der konfliktfreie Fahrplan, der bisher in dieser Arbeit angenommen wurde, reicht jedoch nicht aus, um hier einen Deadlock unmittelbar im Knoten zu erzeugen. Die Begründung liegt in dem Wesen der Simulation und in der Methode, wie hier der Fahrplan gestaltet wurde. Bei den eingleisigen Strecken enden und beginnen die Züge alle im Bahnhof E. D.h. wenn ein Zug im Bahnhof E ankommt, wird er aus dem System herausgenommen und nicht weiter betrachtet. Er „löst sich so zu sagen in Luft auf“ und stellt somit kein Konfliktpotential für einen Deadlock dar. Die einzigen Züge, die im Bahnhof E somit Deadlock relevant sind, sind diese, die ihn durchqueren.

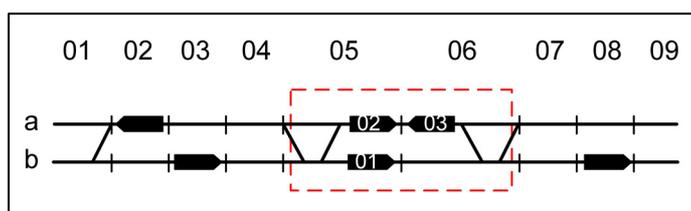
Eine ähnliche Situation entsteht im Szenario 4 – die zweigleisige Strecke. Hier scheint das Verkehrsaufkommen des Fahrplans zu gering zu sein, um einen Deadlock zu erzeugen.

---

Wie in der Szenariobeschreibung erwähnt, scheint es nur die Möglichkeit zu geben einen gesonderten Fahrplan für diese beiden Szenarien zu konstruieren, um auch hier Deadlocks zu erzeugen. Es könnte natürlich auch angenommen werden, dass RailSys in der Lage ist, diese Art von Deadlocks zu vermeiden. Jedoch sollte bis dahin alles Mögliche getan werden, um Deadlocks zu erzeugen und eventuelle Verfahren und Methoden auf einem breiten Spektrum zu testen.

Der neu konstruierte Fahrplan für die Szenarien 4 und 5 wird im Folgenden noch einmal kurz beschrieben. Die durchquerenden Züge des Bahnhofs E von Bahnhof F nach Bahnhof L wurden erhöht. Weiterhin wurden planmäßige Überholungen zwischen langsamen und schnellen Zügen in den Bahnhöfen E und M realisiert. Um einen Konflikt bei Verspätungen herbeizuführen, wurde ein weiterer Zugstrom durch den Bahnhof E erzeugt. Dieser führt von Bahnhof D über E nach Bahnhof G und dies nur in eine Richtung. Alle Züge halten jetzt in jeder Station für mindestens eine Sekunde.

Durch diese Veränderungen im Fahrplan entstehen nun auch Deadlocks im Bahnhof M. Dieser wird in Abbildung 12 in der mikroskopischen Netzansicht in RailSys dargestellt. Ein Deadlock, der hier nun entsteht, stellt sich, wie in Abbildung 26 gezeigt wird, dar.



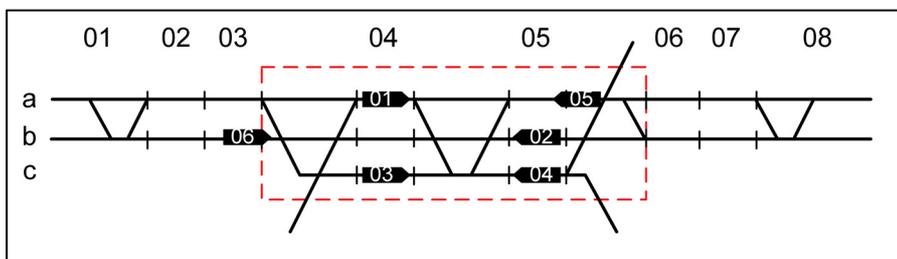
**Abbildung 26: Deadlock im Bahnhof M**

Zug 01 ist ein langsamer Zug mit geringer Priorität. Er wird von Zug 02 planmäßig überholt, wobei er auch Verspätung hat. Zug 03 hat ebenfalls Verspätung und hätte Zug 01 und Zug 02 schon früher konfliktfrei kreuzen sollen. Durch seine Verspätung fahren Zug 02 und Zug 03 jedoch auf einem Gleis (Block 05a und Block 06a) aufeinander zu. Der „Deadlock gefährdete Bereich“ ist in dieser Situation eher klein. Weiterhin ähnelt dieser Deadlock einem trivialen Deadlock, da zwischen den Zügen keine Abzweigmöglichkeit vorhanden ist.

Im Bahnhof E entstehen nun auch Deadlocks. Diese lassen sich in vier Kategorien einteilen, die nachfolgend an jeweils einem Beispiel dargestellt werden. Es sind:

- Deadlock durch Konflikte im Bahnhof
- Simulationsbedingter Deadlock
- Deadlock durch Erweiterung der eingleisigen Strecken
- Deadlock durch Rückstau

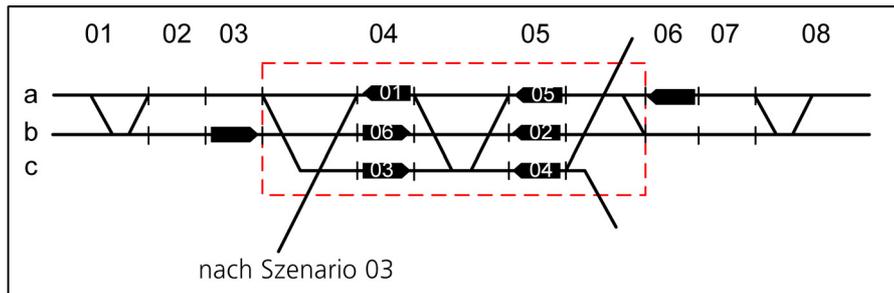
Der klassische Deadlock im Bahnhof entsteht durch Konflikte. Hier gibt es nur eine Zugkonfiguration, die zu einem solchen Deadlock führt. Abbildung 27 stellt diese Situation dar, bei der sich jeweils drei Züge gegenüberstehen.



**Abbildung 27: Deadlock-Situation im Bahnhof E**

Auch hier ist der „Deadlock gefährdete Bereich“ auf die Bahnhofinfrastruktur begrenzt. Vier, der sechs Blöcke, sind schon belegt und zwar mit Zügen geringerer Priorität. Die Züge 05 und 06 fahren jeweils in einen freien Block im Bahnhof ein, um andere Züge zu kreuzen oder zu überholen. In dem Augenblick, indem Zug 06 in den „Deadlock gefährdeten Bereich“ einfährt, ist die Deadlock-Situation hergestellt und der Deadlock unvermeidbar. Weiterhin ist anzumerken, dass dieser Deadlock in seiner speziellen Art nur sehr selten auftritt bzw. schwer aus einen konfliktfreien Fahrplan zu konstruieren ist.

Eine ähnliche Darstellung bietet der simulationsbedingte Deadlock. Auch hier sind alle Blöcke im Bahnhof E belegt, mit dem Unterschied, dass sich nicht alle Züge gegenüberstehen. Die Situation wird in Abbildung 28 dargestellt.

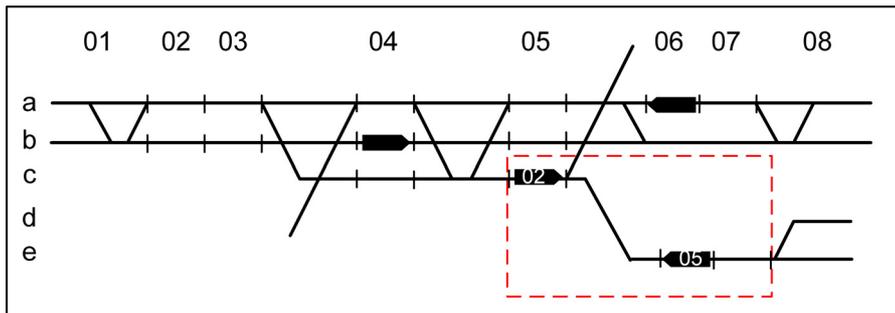


**Abbildung 28: Deadlock in Szenario 5 durch Simulationsfehler**

Zug 01 hat eigentlich die Möglichkeit den Bahnhof E aus Block 04a in Block 03a zu verlassen (Dies ist seine Wunschrichtung). Das System bricht jedoch die Simulation mit der Meldung Deadlock ab. Ein Grund könnte in der Einstellung der Deadlock-Erkennung liegen. Zug 01 wartet zulange und blockiert somit den gesamten Bahnhof. Dieser Stillstand wird vermutlich vom System als Deadlock interpretiert. Ein Indiz dafür ist, dass der Zug, der den Deadlock ausgelöst hat, nicht Zug 01, sondern Zug 04 ist. Somit wäre Zug 01 nach einer bestimmten Zeit weiter gefahren. Die gleiche Situation tritt ebenfalls auf, wenn Zug 01 Richtung Szenario 03 fahren möchte.

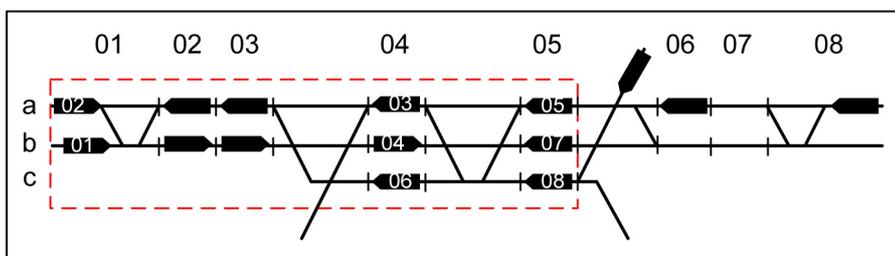
Eine weitere Deadlock-Art, die im Bahnhof E auftritt, ist durch die spezielle Konstruktion der Infrastruktur im Bahnhof E, zu begründen. Der Deadlock durch die Erweiterung der eingleisigen Strecke entsteht bei der Ein- bzw. Ausfahrt des Szenarios 02, bezogen auf den Bahnhof E. Züge, die Richtung Szenario 02 (Bahnhof O) fahren möchten, haben nur die Möglichkeit im Block 05c zu beginnen. Block 05a und 05b sind aufgrund der Gleislage nicht direkt erreichbar. Gleiches gilt für die Züge, die von Bahnhof O kommen. Sie haben nur die Möglichkeit in Block 05c einzufahren. Somit wird die eingleisige Strecke um einen Bahnhofsblock erweitert. RailSys zählt dieses jedoch nicht zur Disposition auf eingleisigen Strecken (siehe Kapitel 5.1.1). Vermutlich, weil eine Weiche dazwischen liegt. So wird ein Deadlock trivialer Art produziert.

Abbildung 29 stellt die Situation dar. Zug 02 steht im Bahnhof E im Block 05a und möchte auf die eingleisige Strecke einfahren. Zug 05 hat mit der Einfahrt in den „Deadlock gefährdeten Bereich“ die Deadlock-Situation jedoch schon hergestellt.



**Abbildung 29: Deadlock-Situation durch Versagen der eingleisigen Disposition**

Die vierte Art von Deadlocks, die im Bahnhof E auftritt, entsteht durch einen Rückstau von Bahnhof F. Durch die definierten Einbruchsverspätungen und Ausweichgleise kommt es im Bahnhof F während des Einbrechens von Zügen in das System zu einer Überholung. Hierbei wird ein verspäteter Zug einem nicht verspäteten Zug vorgezogen und nicht auf sein bereits belegtes Stammgleis, sondern auf ein Ausweichgleis eingelegt. Somit können vorerst keine Züge, die von Bahnhof E kommen, Richtung Bahnhof F abfließen, weil beide Gleise in Bahnhof F belegt sind. Solange die Züge nur noch zufließen und das Abfließen verhindert wird, steigt die Wahrscheinlichkeit eines Deadlocks an. Abbildung 30 stellt beispielhaft eine solche Situation dar.



**Abbildung 30: Deadlock in Bahnhof E durch mangelnde Kapazität**

Hier ist der Deadlock bereits entstanden. Die Züge 03 und 06 möchten in Block 03a einfahren. Zug 08 ist als letzter in den „Deadlock gefährdeten Bereich“ aus Block 06a eingefahren und hat somit den Deadlock erzeugt. Dadurch, dass in den Blöcken 01a und 01b bzw. 05a, 05b und 05c die Züge in der gleichen Richtung stehen und zwischen diesen beiden Blockabschnitten kein weiterer Block frei ist, der den Konflikt lösen könnte, ist durch einen Rückstau ein Deadlock mit großer Komplexität entstanden.

---

## 5 Implementierte Deadlock-Vermeidung in RailSys

In diesem Kapitel sollen alle Methoden und Verfahren beschrieben werden, die in RailSys 4 zur Verfügung stehen, um Deadlocks zu vermeiden. Die optimale Parametrierung ist ein Teil dieser Beschreibung. Hierzu werden die Szenarien herangezogen und an ihnen die Beschreibungs- und Optimierungsprozesse durchgeführt.

Grundsätzlich lassen sich alle Möglichkeiten zur Vermeidung von Deadlocks in RailSys in zwei Gruppen einteilen:

- Methoden durch den Disponenten
- Andere Methoden

Die erste Gruppe wird von einem sogenannten Disponenten aufgerufen und gesteuert. Dieser Disponent versucht aber nicht nur Deadlocks zu vermeiden, sondern hat vor allen die Aufgabe das Störungsmanagement zu händeln und somit Verspätungen auf geringem Niveau zu halten. Alle Aktionen, die vom Fahrplan abweichen, müssen vom Disponenten erkannt, bewertet und entschärft werden. Jede Methode kann einzeln zugeschaltet und der Einflussbereich durch Parameter bestimmt werden.

In der zweiten Gruppe sind alle Methoden enthalten, die unabhängig vom Disponenten eine Auswirkung auf das Deadlock-Verhalten bzw. auf die Vermeidung haben. Dies können auch Methoden sein, die für einen anderen Zweck bestimmt sind, jedoch auch zur Lösung der Deadlock-Problematik beitragen.

Die folgenden Inhalte dieses Kapitels werden zum Teil aus der Quelle [RMC05] entnommen und mit den eigenen Erkenntnissen aus der Simulation ergänzt.

### 5.1 Methoden durch den Disponenten

Konflikte, die durch den Disponenten gelöst werden sollen, müssen frühzeitig erkannt werden. Hierfür stehen die Werte des Vorschaubereiches zur Parametrierung offen. Der Vorschaubereich ist die Zeit, die der Disponent in die

---

Zukunft schaut, um Konflikte zu erkennen. Je größer diese Zeit ist, desto eher wird ein Konflikt erkannt und kann eventuell gelöst werden. Der Vorschaubereich ist nur für alle Disponentenmethoden gleichzeitig einstellbar.

### **5.1.1 Disposition auf eingleisigen Strecken**

Um triviale Deadlocks (siehe Kapitel 4.3.1) auf eingleisigen Strecken zu vermeiden, wird die Dispositionsmethode „Disposition auf eingleisigen Strecken“ eingesetzt. Sie ist eine der wichtigsten Methoden, weil ohne sie das Abbilden von eingleisigen Strecken nicht möglich wäre.

Damit diese Methode angewandt werden kann, muss zuvor durch den Benutzer definiert werden, welche Strecken im Zweirichtungsbetrieb betrieben werden (Dieses kann auch automatisch erfolgen). Hierzu gibt es ein Kantenattribut „Zweirichtungsbetrieb“. Es ist jedoch zu unterscheiden, ob eine Strecke im Zweirichtungsbetrieb betrieben wird oder in beiden Richtungen befahrbar ist. So werden Kanten im Bahnhof nicht mit diesem Attribut versehen. Der Disponent kann aufgrund dieser Unterscheidung die eingleisigen Strecken identifizieren.

Bei dieser Methode kann außerdem auch der Handlungsbereich eingestellt werden. Der Handlungsbereich ist die Zeit, in der der Disponent den Konflikt lösen kann. Die Erkennung und die Lösung eines Konfliktes sind also nicht zur selben Zeit durchzuführen. Der Zeitwert des Handlungsbereiches sollte so eingestellt werden, dass er mit der größten Fahrzeit über eine eingleisige Strecke übereinstimmt. RailSys bietet ein Tool, um diese Zeit zu bestimmen.

Die Methode arbeitet in der Simulation zuverlässig, bis auf einige Ausnahmen, bei denen eine eingleisige Strecke nicht erkannt wird. Dieses ist z.B. der Fall im Szenario 1 auf dem Streckenabschnitt zwischen Bahnhof E und Bahnhof D. Hier kann es trotz definiertem Zweirichtungsbetrieb dazu kommen, dass auf dem eingleisigen Abschnitt triviale Deadlocks entstehen.

Ein weiteres Problem ist, dass jeweils nur eine eingleisige Strecke betrachtet wird. Liegt ein Bahnhof dazwischen, werden beide Strecken unabhängig voneinander betrachtet.

---

---

### 5.1.2 Ausweichgleise

Das Definieren von Ausweichgleisen in Bahnhöfen ist eine der Grundmethoden der Disposition allgemein. Durch diese Möglichkeit hat der Disponent den größten Einfluss auf das Störungsmanagement. Es beziehen sich nahezu alle Dispositionshandlungen auf Ausweichgleise.

Die Ausweichgleise können für jedes Gleis definiert werden. Ist dann ein solches Gleis als Regelfahrweg durch einen Zug A belegt, kann der Disponent einen Zug B, der dieses Gleis befahren möchte, auf ein Ausweichgleis umleiten. Somit kann die gesamte Kapazität der Infrastruktur genutzt werden, um Deadlock zu vermeiden. Der Begriff Disponent ist hier jedoch etwas irreführend. Vielmehr sind die Ausweichgleise als Entscheidung eines Fahrdienstleiters zu sehen, der für seinen Bahnhof kurzfristig Entscheidungen trifft aber nicht das gesamte System, wie ein Disponent, in Betrachtung zieht.

In Bahnhöfen mit geringerer Kapazität kann es daher auch zu einer Erhöhung der Wahrscheinlichkeit von Deadlocks führen. Durch die Definition von Ausweichgleisen kommt es bei Zügen mit Verspätung und hoher Priorität zu Überholungen von niederwertigen Zügen. Hierbei ist das parallele Einfahren in den Bahnhof kritisch zu betrachten. Sobald alle Parallelgleise mit Zügen belegt sind, die alle in die gleiche Richtung fahren, entsteht eine Deadlock kritische Situation. Findet in diesem Augenblick eine Gegenfahrt auf einer angrenzenden eingleisigen Strecke statt, entsteht ein Deadlock.

Die Definition von Ausweichgleisen ist in jedem Fall sinnvoll. Laut der Quelle [RMC05] sollte das Anlegen von Ausweichgleisen jedoch vermieden werden, wenn Züge einen Bahnhof nur durchqueren und in ihm nicht halten. Es hat jedoch keine Auswirkung auf das Deadlockverhalten, wenn Ausweichgleise definiert wurden. Die oben beschriebenen Szenarien sind so angelegt, dass beim Ausweichen auf ein anderes Gleis keine Fahrzeiteinbußen entstehen.

### 5.1.3 Zielgleisänderung

Die Zielgleisänderung betrachtet, im Gegensatz zu den Ausweichgleisen, das ganze System. Natürlich ist die Betrachtung durch die Einstellung der Dispositionsvoraussicht begrenzt. Eine Zielgleisänderung wird immer schon im

---

Voraus durchgeführt, auf der Entscheidungsbasis zukünftiger Konflikte im Bahnhof.

Stellt der Disponent im Rahmen seines Vorschaubereiches fest, dass voraussichtlich zwei Züge das gleiche Zielgleis in einen Bahnhof befahren werden, ändert er für einen der beiden Züge das Zielgleis im Konfliktbahnhof. Ist dies nicht möglich, muss eine andere Dispositionsmethode eingreifen.

Der Unterschied zu den Ausweichgleisen ist, dass die Entscheidung zur Änderung des Gleises nicht unmittelbar vor der Einfahrt in den Bahnhof, sondern bei der Identifikation eines Konfliktes bzw. in der Zeit dazwischen stattfinden kann. Dies kann ein Zeitpunkt sein, der einige Zeit vor dem eigentlichen Konflikt liegt. Der Vorteil besteht in dem Wissen um den Konflikt und in der alternativen Anwendung einer Dispositionsmethode, wenn eine Zielgleisänderung nicht möglich ist.

Als Basis der Zielgleisänderung dienen die Ausweichgleise. Sind keine Ausweichgleise definiert, können auch keine Zielgleisänderungen vorgenommen werden.

Für die Zielgleisänderung kann der Wert der Mindestverspätung und des Handlungsbereiches definiert werden. Die Mindestverspätung gibt an, ab welchem Zeitwert der Verspätung eines Zuges, die durch einen Konflikt voraussichtlich entsteht, die Methode eingesetzt werden soll. Wird dieser Zeitwert überschritten, ist die Methode aktiv. Der Handlungsbereich ist die Zeit, in der das System in der Lage ist, einen Konflikt im Voraus zu lösen.

#### **5.1.4 Disposition für Alternativfahrwege**

Die Alternativfahrwege sind eine Erneuerung von RailSys 4. Sie stehen im Zusammenhang mit dem Definieren von Betriebsstellengrenzen. Die Grenzen einer Betriebsstelle werden durch sogenannte Betriebsstellengrenzen definiert. Zwischen diesen Grenzen können dann Betriebsstellenfahrwege definiert werden. Mit Hilfe dieser, kann in der Fahrplanerstellung einem Zug der exakte Fahrweg durch eine Betriebsstelle zugewiesen werden.

---

Weiterhin werden die definierten Fahrwege auch zur Disposition in Betriebsstellen herangezogen. Ist ein Teil des Regelfahrweges durch die Betriebsstelle von einem Zug belegt, wird unter bestimmten Voraussetzungen ein Alternativfahrweg bestimmt. Diese Voraussetzungen sind:

- Das der Alternativfahrweg an denselben Betriebsstellengrenzen, wie der Regelfahrweg beginnt und endet.
- Das der Alternativfahrweg vollständig konfliktfrei ist.
- Der Alternativfahrweg über den gleichen betrieblichen Punkt führen muss, wie der Regelfahrweg.

Ein betrieblicher Punkt ist z.B. eine Haltetafel oder ein Stationssignal. Der betriebliche Punkt kann über die Definition von Ausweichgleisen und die Änderung des Zielgleises verändert werden. Dadurch wird ein Scheitern dieser Methode durch die dritte Voraussetzung bedingt verhindert.

Die Konfliktfreiheit des Alternativfahrweges wird nach drei unterschiedlichen Möglichkeiten geprüft:

- Bei der Einfahrt mit „Halt“ an einem betrieblichen Punkt wird geprüft, ob der Fahrweg bis zum betrieblichen Punkt frei ist.
- Bei der Ausfahrt von einem betrieblichen Punkt wird geprüft, ob der Fahrweg zwischen dem betrieblichen Punkt und der Betriebsstellengrenze frei ist.
- Bei der Durchfahrt eines Bahnhofes wird geprüft, ob der Fahrweg zwischen den Betriebsstellengrenzen frei ist.

Durch diese Unterscheidungen wird gewährleistet, dass nicht mehr vom Fahrweg geprüft wird, als für eine Fahrt nötig ist. Hierdurch kann die Kapazität der Infrastruktur erhöht werden.

Auch bei der Alternativfahrwegdisposition können jeweils Zeitwerte für die Mindestverspätung und den Handlungsbereich definiert werden. Nach Quelle [RMC05] wird empfohlen für die Mindestverspätung einen Wert von 15s bis 30s anzusetzen. Der Wert ist so klein gewählt, weil die Konflikte innerhalb einer Station meist keinen großen Umfang haben und dadurch auch schon bei

---

geringen Verspätungen gelöst werden sollten. Gleiches gilt für den Handlungsbereich. Aufgrund der geringen Mindestverspätung reicht es diesen Wert auf 1min bis 3min zu setzen.

## **5.2 Weitere Methoden in RailSys**

### **5.2.1 Fahrwegsreservierung**

Die Fahrwegsreservierung ist eine Methode zur Verlängerung des Reservierungsbereiches vor einem Zug. Im Normalfall reserviert ein Zug immer nur einen Block im Voraus bzw. nur den Block, den er gerade befährt. Durch die Parametrierung der Zuggattungen kann eine solche Reservierung des Laufweges eines Zuges vergrößert werden.

Ist ein Abschnitt, dessen Länge beliebig ist, von einem Zug reserviert, kann kein anderer Zug diesen Abschnitt befahren bzw. reservieren. Dadurch kann vermieden werden, dass z.B. sich kein langsamer Zug mit geringer Priorität vor einen schnellen Zug setzt und diesen behindert.

Die Reservierungslänge kann auf zwei Wege bestimmt werden. Entweder über einen Zeitwert oder über die Anzahl der Blöcke. Werden beide Werte definiert, gilt jeweils das Maximum.

Auf die Deadlock-Vermeidung hat die Fahrwegsreservierung ebenfalls einen Einfluss. Durch die Reservierung mehrerer Blöcke im Voraus können sowohl Konflikte in Bahnhöfen als auch im Zusammenhang mit eingleisigen Streckenabschnitten vermieden werden. Jedoch ist dann die Reservierungslänge so groß, dass dies negative Auswirkungen auf die Kapazität der Infrastruktur hat. Die Kapazität wird aus dem Grund verringert, weil die zu reservierenden Blöcke vor dem Zug zu einem relevanten Block zusammengefasst werden. Somit können Züge bei Folgefahrten nicht so dicht aufeinander folgen, wie ohne eine Fahrwegsreservierung.

Aufgrund dieser Eigenschaft sollte die Fahrwegsreservierung, unter Berücksichtigung der Infrastrukturkapazität und des darauf abzufertigende Betriebsprogramms, eher klein angesetzt werden, weil es eine Methode ist, die keinem realen Vorbild entspricht.

---

---

### 5.2.2 Signalvorschaubereich

Ähnlich, wie die Fahrwegsreservierung, ist die Methode des Signalvorschaubereiches zu bewerten. Auch hier wird ein zu definierender Streckenabschnitt vor der Fahrt des Zuges reserviert, mit dem Unterschied, dass dieses für jeden Zug und für jedes Signal einzeln geschehen kann. Außerdem werden die Blöcke, die auf „Fahrt“ zeigende Signale geprüft wurden, nicht reserviert. D.h. nach Fahrtbeginn kann trotzdem ein entgegenkommender Zug in den zuvor geprüften Bereich einfahren.

Mit dieser Methode kann z.B. simuliert werden, dass ein schwerer Güterzug, der eine Steigung befährt, auf dieser Steigung nicht anhalten muss, weil sichergestellt wird, dass alle Signale im betroffenen Abschnitt „Fahrt“ zeigen. Ein schwerer Zug kann Steigungen unter bestimmten Randbedingungen nur ohne anzuhalten befahren, weil seine Zugkraft zum Anfahren an der Steigung nicht ausreicht.

Der Signalvorschaubereich kann für jedes Signal in beliebiger Anzahl von Blöcken definiert werden. Der Vorteil gegenüber der Fahrwegsreservierung liegt darin, dass der Signalvorschaubereich an bestimmten Stellen und nicht im gesamten Laufweg definiert werden kann.

Dieser Vorteil äußert sich auch positiv in der Kapazitätsbetrachtung. Wird diese Methode zur Deadlock-Vermeidung eingesetzt, kann eine gezielte Definition stattfinden, dort wo eine Risikowahrscheinlichkeit für Deadlocks besteht.

Die Methode bietet weiterhin die Möglichkeit nach einfacher Belegung und Belegung durch eine Gegenfahrt zu unterscheiden. Bei einfacher Belegung ist es egal ob eine Gegen- oder eine Folgefahrt stattfindet. Bei der Prüfung auf Belegung durch Gegenfahrt werden nur die Konflikte betrachtet, die mit einem entgegenkommenden Zug entstehen. Der erste Block wird jedoch immer auf einfache Belegung geprüft.

### 5.2.3 Abhängige Kanten

RailSys bietet die Möglichkeit sogenannte abhängige Kanten zu definieren. Hierbei werden einzelne Kanten untereinander in Verbindung gesetzt. Im

---

einfachsten Fall bedeutet dies, dass wenn eine Kante A mit einer anderen Kante B verbunden ist und Kante B durch einen Zug belegt oder reserviert wird, dann ist Kante A, auch wenn sie nicht direkt belegt wird, als reserviert im System angesehen. Ein Zug, der den Block befahren möchte, indem Kante A liegt, muss solange warten, bis Kante B, die in einem anderen Block liegen kann, wieder frei ist.

Abhängige Kanten können auf zwei Arten miteinander verknüpft werden. Es stehen eine logische UND- und eine logische ODER-Verknüpfung zur Verfügung. Bei einer UND-Verknüpfung müssen alle verknüpften Kanten frei sein, damit die Ausgangskante nicht als belegt angezeigt wird. Bei einer ODER-Verknüpfung muss mindestens eine verknüpfte Kante frei sein, damit die Ausgangskante als nicht belegt angezeigt wird.

Diese Methode bietet unterschiedlichste Möglichkeiten, aufgrund von logischen Zusammenhängen, Deadlocks zu vermeiden. Es ist jedoch sehr aufwendig die Verknüpfungen im großen Umfang anzulegen.

### **5.3 Test der Methoden in RailSys**

In diesem Kapitel werden die oben beschriebenen Methoden in RailSys auf ihre Wirksamkeit hin getestet. Weiterhin soll über die zu definierenden Parameter der Disponentenmethoden eine möglichst geringe Wahrscheinlichkeit für Deadlocks erzielt werden.

#### **5.3.1 Vorstellung des Testverfahrens**

Es wird im Nachfolgenden jede Disponentenmethode, auf seine Möglichkeiten zur Deadlock-Vermeidung hin, in jedem Szenario einzeln getestet. Dazu kommt ein Test des Gesamtnetzes. Hier wird versucht für die optimale Einstellung der Methoden, unabhängig von den anderen Methoden, zu deklarieren. Dafür wird immer nur die aktuelle Methode aktiv geschaltet.

Ist die optimale Einstellung für jede Methode gefunden, werden die Methoden zusammen getestet und die Parameter nochmals auf eine möglichst optimale Einstellung hin getestet. Auch dieses geschieht für jedes Szenario und für das Gesamtnetz.

---

Um zu ermitteln, ob eine Verringerung der Wahrscheinlichkeit eines Deadlocks durch eine Methode herbeigeführt wird, werden 100 mit Störungen überlagerte Fahrpläne erzeugt und die Betriebsimulation gestartet. Dann wird der Anteil der Fahrpläne ermittelt, die durch einen Deadlock abgebrochen wurden. Dieser Anteil ist die Wahrscheinlichkeit, dass ein Deadlock auftritt. Beim Zuschalten der Methoden sollte sich die Wahrscheinlichkeit für Deadlocks verringern.

Als Ausgangswert für die Methodenwerte werden die voreingestellten Werte von RailSys angenommen und durch die Vorgaben aus der Quelle [RMC05] ergänzt. Zur Ermittlung des optimalen Wertes wird dann in sinnvollen Intervallen der jeweilige Wert verändert und die Reaktion auf die Simulation protokolliert.

Zuerst muss jedoch ein Wert für den Vorschaubereich ermittelt werden. Der Vorschaubereich sollte möglichst groß sein, um alle Konflikte möglichst frühzeitig zu erkennen. Zu welchem Zeitpunkt ein Konflikt dann gelöst wird, ist vom Handlungsbereich der jeweiligen Methode abhängig. In dieser Betrachtung wird der Vorschaubereich auf die Voraussicht auf mindestens den übernächsten Bahnhof gesetzt. Hierfür ist der langsamste Zug maßgebend.

### **5.3.2 Test der Zielgleisänderung**

Die im Programm voreingestellten Parameter für die Zielgleisänderung sind:

- Mindestverspätung, 120s
- Handlungsbereich, 3min

Für die Untersuchung sollten geeignete Parameter definiert werden. Zunächst muss untersucht werden, welchen Einfluss die beiden Parameter untereinander haben. Der Handlungsbereich wird aufgrund der zugrundeliegenden Infrastruktur und der Fahrgeschwindigkeit des langsamsten Zuges berechnet. Es wird zum einen der Handlungsbereich auf einen und auf zwei Bahnhöfe im Voraus betrachtet. D.h. der Disponent kann einen Konflikt zweier Züge lösen, bevor der Zug aus dem voraus liegenden Bahnhof ausfährt. Die Standardeinstellung des Handlungsbereiches von 3min symbolisiert eine Entscheidung des Disponenten kurz vor der Einfahrt in den Bahnhof.

---

Die Mindestverspätung wird mit dem Standardwert und einem wesentlich kleinerem Wert (30s) getestet. Der Wert von 30s stellt eine Sensibilisierung des Disponenten dar. D.h., dass er auch schon bei voraussichtlich geringen Verspätungen in Aktion tritt.

Wie der Tabelle 1 zu entnehmen ist, hat die Anwendung der Methode zur Zielgleisänderung nicht auf alle Szenarien einen Einfluss. In den Szenarien 1 und 2 erzielt die Methode gar keine Wirkung. Szenario 1 hat mit und ohne diese Einstellung 100% Deadlocks – dies ist jedoch kein Beweis für die Untauglichkeit der Methode in diesem Szenario, weil der Wert von 100% extrem ist und somit andere Gründe haben könnte. Den Beweis, dass diese Methode der Zielgleisänderung keinen Einfluss auf diese Art von Strecke hat, liefert Szenario 2. Hier treten in 42% der Fahrplandurchläufe Deadlocks auf, egal ob mit oder ohne die Methode. Auch die Veränderungen der Parameter haben keinen Einfluss darauf. Die Methode wird vom Disponenten nicht aufgerufen, weil keine Situation eintritt, in der sie eingreifen könnte. Vermutlich liegt dies an der Zweigleisigkeit der Bahnhöfe, die somit eine geringe Kapazität aufweisen und eine Zielgleisänderung nicht möglich ist, weil keine Ressourcen zur Verfügung stehen. (In Szenario 2 können keine Zielgleisänderungen stattfinden, weil die Bahnhöfe im Einrichtungsbetrieb betrieben werden.)

Anders sieht die Situation in Szenario 3 aus. Hier hat die Methode einen erheblichen Einfluss auf die Wahrscheinlichkeit von Deadlocks. Ohne die Zielgleisänderung liegt diese bei 82%. Schon in der Standard RailSys Einstellung wird dieser Wert auf 52% reduziert. Wird der Handlungsbereich auf 20min vergrößert, liegt die Wahrscheinlichkeit für einen Deadlock nur noch bei 34%. Der Wert der Mindestverspätung wurde zwar auch verringert, jedoch hat er keinen nennenswerten Einfluss, weil die vorhandenen Verspätungen den Wert von 120s in den meisten Fällen überschreiten. Die Auswirkungen in Szenario 3 sind aus dem Grund so groß, weil hier ausreichend Kapazität in den Bahnhöfen bereit steht, mit der der Disponent arbeiten kann.

---

	Min. Versp.	Hand. Ber.	Szen. 1	Szen. 2	Szen. 3	Szen. 4	Szen. 5	Netz
	[s]	[min]	[%]	[%]	[%]	[%]	[%]	[%]
ohne	-	-	100	42	82	85	2	100
Standard	120	3	100	42	52	74	24	100
Test 01	120	20	100	42	39	72	-	100
Test 02	30	20	100	42	38	73	-	100
Test 03	30	35	100	42	38	72	-	100

**Tabelle 1: Deadlock-Wahrscheinlichkeiten durch Zielgleisänderung**

Szenario 4 wird ebenfalls von der Methode beeinflusst. Jedoch sind die Auswirkungen nicht so relevant, wie in Szenario 3. Bei weiteren Veränderungen der Parameter ergeben sich keine relevanten Änderungen der Wahrscheinlichkeit.

In Szenario 5 kommt es zu einem Phänomen. Ohne die Methode der Zielgleisänderung kommt es nur zu 2% Deadlocks. Erst nach dem Einschalten der Methode erhöht sich die Wahrscheinlichkeit für einen Deadlock auf 24%. Dieses legt nahe, dass der besondere Aufbau des Bahnhofes E mit der Methode der Zielgleisänderung nicht kompatibel ist. Ein Großteil der Deadlocks stellt sich, wie in Abbildung 28 visualisiert wird, dar. Bei dieser Art von Deadlock ist bis jetzt unklar, warum er entsteht, da die Ausfahrt des Zuges 03 gewährleistet ist. Es könnte sein, dass durch die Änderung des Zielgleises ein Gleis ausgewählt wird, das für den Zug nicht mehr erreichbar ist. Der Deadlock entsteht nun durch die verminderte Kapazität, und diese, weil Zug 03 nicht ausfährt. Die Besonderheit des Bahnhofes E liegt darin, dass jeweils zwei Zielgleise hintereinander liegen. Eine weitere Betrachtung der Methode in diesem Szenario wird nicht durchgeführt, weil das Optimum schon ohne die Methode erreicht ist. Für dieses Szenario muss eine andere Methode Anwendung finden.

Die Betrachtung des Gesamtnetzes führt hier zu keinem Erfolg, weil die Deadlocks aus dem Szenario 1 die Simulation überlagern.

### 5.3.3 Test der Disposition für Alternativfahrwege

Die Standard Parameter für die Alternativfahrwegsdisposition sind vom Programm wie folgt voreingestellt:

- Mindestverspätung, 30s
- Handlungsbereich, 3min

Die Parameter sollten so klein gehalten werden, weil Konflikte in Betriebsstellen immer kurzfristig gelöst werden.

Jedoch werden die Werte in verschiedenen Einstellungen in allen Szenarien getestet. Es ergibt sich jedoch keine Veränderung in der Wahrscheinlichkeit für einen Deadlock. Die Methode wurde in keinem Fall der Betrachtung aufgerufen.

Da diese Methode in der Simulation keine Anwendung findet, liegt vermutlich daran, dass sie an sehr viele Bedingungen geknüpft ist. So kann zum Beispiel ein Alternativfahrweg nur gewählt werden, wenn die an den betrachteten Bahnhof angrenzende Strecke keinen Konflikt für den zu disponierenden Zug beinhaltet. Weiterhin muss der betriebliche Punkt in der Betriebsstelle derselbe bleiben. D.h. die Methode kann nur in Betriebsstelle zum Einsatz kommen, die aus mehr als einen Block hintereinander bestehen und zwischen den Blöcken eine Kreuzungsmöglichkeit vorhanden ist. Diese Eigenschaften bietet nur der Bahnhof E. Da im Szenario 5 jedoch 2% Deadlocks auftreten, ergibt sich nicht die Möglichkeit diese Methode einzusetzen.

#### **5.3.4 Test der Fahrwegsreservierung**

Die Fahrwegsreservierung kann als Zeitwert und als Anzahl der Blöcke eingestellt werden. Bei der Definition durch den Zeitwert ist der zu reservierende Bereich in seiner Größe flexibel. Fährt ein Zug z.B. langsam, kann er in der gleichen Zeit nur eine kürzere Strecke zurücklegen, als ein schnellerer Zug. Gleiches gilt für den Vorschaubereich.

Die Fahrwegsreservierung wird hier nur über die Anzahl der Blöcke definiert. Es wird in jedem Szenario die Reservierung von jeweils eins, zwei, drei, vier und fünf Blöcken betrachtet. Beide Modellzüge (schnell und langsam) werden jeweils gleich parametrisiert.

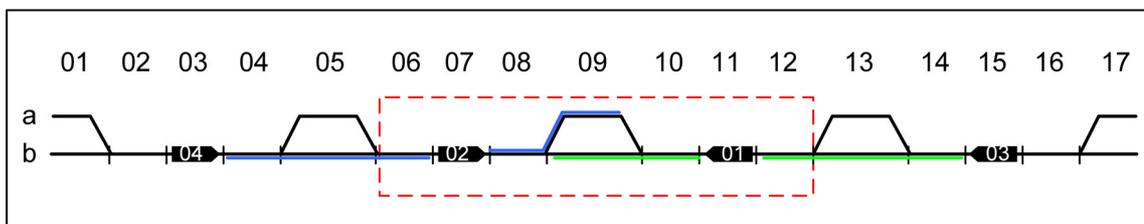
Die optimale Anzahl der zu reservierenden Blöcke wird bei Anzahl vier vermutet, weil dies der Vorschaubereich auf den nächsten Bahnhof ist. Jedoch gilt dies nur bei der Vermeidung von Deadlocks mit drei Zügen (siehe Kapitel 4.3.2). Tabelle 2 stellt die Ergebnisse dar.

---

	Anzahl Blöcke	Szen. 1	Szen. 2	Szen. 3	Szen. 4	Szen. 5	Netz
		[%]	[%]	[%]	[%]	[%]	[%]
Standard	-	100	42	82	85	2	100
Test 1	1	100	42	80	87	2	100
Test 2	2	100	91	75	92	1	100
Test 3	3	100	100	70	99	2	100
Test 4	4	100	100	95	81	2	100
Test 5	5	100	100	98	81	1	100

**Tabelle 2: Deadlock-Wahrscheinlichkeiten durch Fahrwegsreservierung**

In den Szenarien 1 und 2 bewirkt die Methode eher eine Verschlechterung, als eine Verbesserung der Wahrscheinlichkeit für einen Deadlock. Die Deadlocks entstehen schon kurz nach dem Start des Simulationsdurchlaufes. Im typischen Fall nach der ersten Kreuzung zweier Züge. Bei der Einfahrt in den Bahnhof, in dem die Kreuzung stattfinden soll, verkürzen sich die Fahrwegsreservierungen auf den Wert Null, da beide Züge auf ein „Halt“ zeigendes Signal zufahren. Das „Halt“ zeigende Signal wird durch den Reservierungsbereich des entgegenkommenden Zuges ausgelöst. Abbildung 31 stellt eine solche Situation beispielhaft dar. Die Züge 02 und 01 fahren in den Bahnhof im Abschnitt 09 ein. Die farbigen Linien zeigen jeweils den reservierten Bereich an. Aufgrund dessen, dass Zug 03 und Zug 04 jeweils so dicht wie möglich dem vorausfahrenden Zug folgen, entsteht ein Deadlock, wie er in Kapitel 4.3.3 beschrieben wird.



**Abbildung 31: Entstehen einer Deadlock-Situation trotz Fahrwegsreservierung**

In Szenario 3 hat diese Methode durchaus positive Effekte auf die Deadlock-Wahrscheinlichkeit. Jedoch nicht, wie vermutet, bei der Reservierung von vier Blöcken, sondern bei drei Blöcken. Hier ist die Wahrscheinlichkeit von 82% auf 70% gesunken. Im Gegensatz zu den Szenarien 1 und 2 kommt Szenario 3 erneut die größere Kapazität zu gute, die wohl Voraussetzung für den Erfolg dieser Methoden ist.

Im Szenario 4 liegt der Deadlock Schwerpunkt im Bahnhof M. Tendenziell hat die Methode jedoch keinen Einfluss auf die Deadlocks an dieser Stelle. Es entstehen

gering mehr Deadlocks, als ohne die Methode. Dies kann auch an der veränderten Zugfolge, also indirekt an der Methode liegen.

Auf Szenario 5 hat diese Methode ebenfalls keinen Einfluss. Die Anzahl der Deadlocks verändert sich nicht signifikant.

Für die Betriebssimulation des gesamten Netzes gilt gleiches, wie oben beschrieben ist. Die Simulation wird von den Ergebnissen aus Szenario 1 überlagert.

### **5.3.5 Test des Signalvorschaubereiches**

Um einen Test des Signalvorschaubereiches durchzuführen, muss zunächst definiert werden, an welchen Signalen es sinnvoll ist, diese Methode einzusetzen.

Für die Szenarien 1, 2 und 3 erscheint es zweckmäßig immer bei der Ausfahrt aus einem Bahnhof den Signalvorschaubereich einzusetzen. Somit wird die Ausfahrt aus dem Bahnhof nur gewehrt, wenn die Bedingungen durch die Methode gesichert sind. Die Größe des Signalvorschaubereiches orientiert sich wieder an der Blockanzahl zwischen zwei Bahnhöfen. Um zu überprüfen, ob im nächsten Bahnhof ein Gleis zur Einfahrt frei ist, werden mindestens vier freie Blöcke benötigt. Um auszuschließen, dass sich auf dem nachfolgenden eingleisigen Streckenabschnitt kein Zug befindet, sind es sieben Blöcke. Diese beiden Vorschaubereichstiefen werden in jedem der drei Szenarien auf einfache Belegung und auf Belegung durch Gegenfahrt getestet.

In den Szenarien 4 und 5 wird der Signalvorschaubereich nicht am Ausfahrtsignal, sondern am Einfahrtsignal definiert. Da die Bahnhöfe M und E aus zwei aufeinander folgenden Blöcken bestehen, wird der Vorschaubereich jeweils auf 2 und 3 Blöcke eingestellt und getestet. Weiterhin muss nur auf Gegenfahrt geprüft werden, weil nur in diesem Fahrmanöver Deadlocks entstehen.

Tabelle 3 zeigt die Testergebnisse für den Signalvorschaubereich, als die Wahrscheinlichkeiten für einen Deadlock.

---

	Typ	Anzahl Blöcke	Szen. 1	Szen. 2	Szen. 3	Szen. 4	Szen. 5	Netz
			[%]	[%]	[%]	[%]	[%]	[%]
Standard	-	ohne	>99	42	82	85	2	-
Test 1	Gegenfahrt	4	>99	42	95	-	-	-
Test 2	Gegenfahrt	7	>99	>99	57	-	-	-
Test 3	Belegung	4	>99	<1	<1	-	-	>99
Test 4	Belegung	7	>99	>99	>99	-	-	-
Test 5	Gegenfahrt	2	-	-	-	90	2	-
Test 6	Gegenfahrt	3	-	-	-	90	2	-

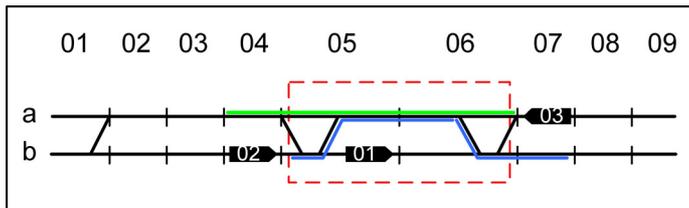
**Tabelle 3: Deadlock-Wahrscheinlichkeiten durch Signalvorschaubereich**

In Szenario 1 bewirkt diese Methode keine Verbesserung der Deadlock-Wahrscheinlichkeit. Es ändert sich jedoch die Art der Deadlocks, die entstehen. Sind es ohne die Methode Deadlocks, die z.B. aus einer Überholung entstehen (siehe Kapitel 4.3.2), sind es nun Deadlocks aus dem Fahrmanöver „Folgefahrt“ (siehe Kapitel 4.3.3). Das Problem liegt in den Ausweichgleisen. Wird ein Zug auf ein anderes Gleis geleitet, das nicht dem Regelfahrweg entspricht, ist für diesen Zug ein anderes Ausfahrtsignal zuständig, welches nicht parametrisiert werden kann. Somit überprüft dieser Zug nicht, ob die Gleise vor ihm frei sind.

In Szenario 2 ergibt sich ein anderes Bild. Hier scheint eine Methode gefunden zu sein, die Deadlocks zu annähernd 100% lösen kann. Jedoch geschieht dies auf Kosten der Leistungsfähigkeit der Strecke. Durch den Signalvorschaubereich von vier Blöcken kann ein Zug B einem Zug A bei der Ausfahrt aus einem Bahnhof nur folgen, wenn Zug A den nächsten Bahnhof schon durchquert hat. Somit wird aus drei Blöcken zwischen den Bahnhöfen ein Block und die Strecke hat bei Folgefahrten eine geringere Leistungsfähigkeit. Jedoch wird hierdurch sichergestellt, dass keine Deadlocks durch Folgefahrten (siehe Kapitel 4.3.3) entstehen können. Die Eigenschaft des Szenarios ist natürlich eine Voraussetzung dafür, dass nur solche Deadlocks produziert werden (Einrichtungsbetrieb in allen Bahnhöfen im Szenario 2).

Für Szenario 3 treffen die gleichen Umstände zu. Es entstehen keine Deadlocks durch Folgefahrten mehr. Deadlocks durch Überholungen treten in diesem Szenario nicht auf, weil die Kapazität der Bahnhöfe größer als in Szenario 1 ist.

Auf die Szenarien 4 und 5 hat die Methode keine direkte Wirkung. Hier ist das Problem, dass der Bereich der geprüft wird, nicht mit dem relevanten Bereich übereinstimmt. Abbildung 32 stellt eine solche Situation in Bahnhof M dar.



**Abbildung 32: Entstehen einer Deadlock-Situation trotz Signalvorschaubereich**

Die Signalvorschautiefe beträgt hier drei Blöcke. Sie werden durch farbige Linien in der Abbildung 32 dargestellt. Die Züge 02 und 03 prüfen nur ihren Fahrweg, in dem der jeweils andere Zug noch nicht relevant ist. Somit entsteht der Deadlock in den Blöcken 05a und 06a.

Für alle Szenarien gilt, dass es einen optimalen Signalvorschaubereich zur Vermeidung von Deadlocks nicht zu geben scheint. In den Fällen, die durch die Methode positiv beeinflusst werden, ergibt sich ein negatives Ergebnis für die Leistungsfähigkeit der Strecke.

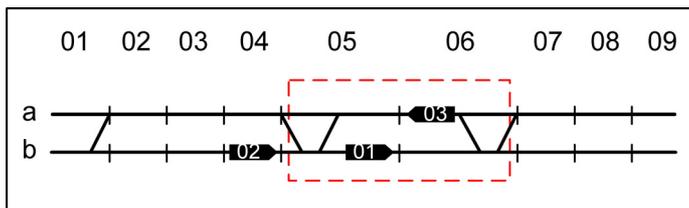
### 5.3.6 Test der abhängigen Kanten

In der Quelle [RMC05] werden die abhängigen Kanten als Methode beschrieben, mit der man Deadlocks vermeiden kann, die immer an der gleichen Stelle und der gleichen Situation auftreten. Eine solche Situation stellt sich z.B. in Szenario 4 dar. Hier beträgt die Deadlock-Wahrscheinlichkeit ohne die Anwendung einer Methode 85%. Annähernd 100% dieser Deadlocks treten im Bahnhof M auf. Eine Beschreibung der dort entstehenden Deadlocks, ist in Kapitel 4.3.5 wiedergegeben.

Dieses Szenario soll beispielhaft für die Wirksamkeit der abhängigen Kanten betrachtet werden. Hierfür muss in einem ersten Schritt definiert werden, wie der Deadlock verhindert werden kann. In Abbildung 26 wird die Deadlock-Situation dargestellt. Beide Züge stehen sich im Bahnhof M gegenüber. Um einen Deadlock zu vermeiden, muss sichergestellt werden, dass einer der beiden Züge (Zug 02 oder 03) nicht in den „Deadlock gefährdeten Bereich“ einfährt. So könnte der jeweils andere Zug den Bahnhof M räumen und anschließend der Zug, der vor dem „Deadlock gefährdeten Bereich“ gewartet hat, den Bahnhof M ohne einen Konflikt passieren. Welche Aufgabe auf einen der beiden Züge anzuwenden ist, wird durch die zeitliche Ankunft am Bahnhof entschieden. Der

erste Zug am Bahnhof darf ihn zuerst passieren, der zweite Zug muss warten. Bei der Vermeidung von Deadlocks werden die Prioritäten der Züge grundsätzlich vernachlässigt, weil die Folgen durch einen Deadlock größer sind, als eine einzelne Verspätung eines Zuges.

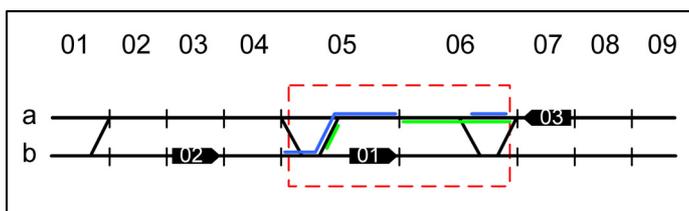
Abbildung 33 stellt eine der Situationen zur Vermeidung des Deadlocks dar. Zug 02 erreicht den Bahnhof als zweiter und wartet vor dem „Deadlock gefährdeten Bereich“. Zug 02 passiert unter dessen den Bahnhof ohne einen Konflikt.



**Abbildung 33: Deadlock-Vermeidung im Bahnhof M**

Da die Definition zur Vermeidung des Deadlocks jetzt aufgestellt ist, muss die Umsetzung durch die abhängigen Kanten nun erfolgen. Es muss sichergestellt werden, dass Zug 02 nicht in Block 05a einfährt, wenn Block 06a durch einen Zug belegt ist. Weiterhin darf Zug 03 nicht in Block 06a einfahren, wenn Block 05 durch einen Zug belegt ist.

Abbildung 34 zeigt die Verknüpfungen der Kanten untereinander. Die blaue Kante in Block 06a ist nur frei, wenn alle blauen Kanten im Abschnitt 05 nicht belegt sind. Gleiches gilt für die grünen Kanten in der Gegenrichtung.



**Abbildung 34: Deadlock-Vermeidung durch abhängige Kanten**

Nach Einrichtung der abhängigen Kanten wird die Simulation erneut gestartet. Vor Anwendung dieser Methode betrug die Wahrscheinlichkeit für einen Deadlock 85%. Nach der Einführung werden annähernd 100% der Deadlock im Szenario 4 vermieden. Hierbei ist wichtig zu erwähnen, dass die Leistungsfähigkeit bei Folgefahrten nicht beeinträchtigt wird. Da die Fahrtrichtung bei der Überprüfung auf Belegung mitberücksichtigt wird, können

Züge im normalen Blockabstand folgen, ohne Kapazitätsverluste in Kauf nehmen zu müssen.

### **5.3.7 Test der Methoden in Kombination**

In den vorangegangenen Kapiteln wurde die Wirksamkeit von Methoden zur Deadlock-Vermeidung, die in RailSys 4 implementiert sind, getestet. Dieses geschah unter dem Gesichtspunkt eine optimale Parametrierung für die Methoden zu finden. Nun werden alle betrachteten Methoden gleichzeitig auf ihre Wirkung hin untersucht. Weiterhin werden alle weiteren Methoden, die der Disponent zur Verfügung stellt, eingeschaltet, um die Situation der Deadlock-Vermeidung in RailSys abzuschätzen.

Aufgrund der Vielzahl an Parametern können nicht sämtliche Kombinationen durchgeführt werden. Aus den oben ermittelten Parametern und logischen Einschätzungen, werden sechs Testdurchläufe für jedes Szenario durchgeführt.

1. Standardeinstellung 00 – Die Parameter in den Disponenteneinstellungen werden, bis auf die eingleisige Disposition, auf den vorgegebenen Werten belassen, die RailSys vorgibt. Weiterhin wird der Wert für die Fahrwegsreservierung auf einen Block eingestellt. Der Signalvorschaubereich wird nicht aktiviert.
  2. Maximum 00 – Die Parameter in den Dispositionseinstellungen werden für den Handlungsbereich möglichst groß und der Mindestverspätung möglichst klein eingestellt. Weiterhin wird der Wert für die Fahrwegsreservierung auf vier Blöcke eingestellt. Der Signalvorschaubereich wird auf einfache Belegung gesetzt. Die Vorschautiefe beträgt ebenfalls vier Blöcke.
  3. Maximum 01 – Gleiche Einstellungen wie Maximum 00, jedoch wird die Fahrwegsreservierung auf zwei Blöcke eingestellt. Der Signalvorschaubereich wird nur auf Gegenfahrt geprüft.
  4. Standardeinstellungen 01 – Die Parameter des Disponenten werden auf Standard gestellt. Die Fahrstraßenreservierung wird auf zwei Blöcke und
-

der Signalvorschaubereich auf vier Blöcke gesetzt (Test auf einfache Belegung).

5. Standardeinstellung 02 – Die Parameter des Disponenten werden auf Standard gestellt. Die Fahrstraßenreservierung wird auf vier Blöcke und der Signalvorschaubereich auf zwei Blöcke gesetzt (Test auf Gegenfahrt).
6. Standardeinstellung 03 – Die Parameter des Disponenten werden auf Standard gestellt. Die Fahrstraßenreservierung wird auf vier Blöcke und der Signalvorschaubereich ebenfalls auf vier Blöcke gesetzt (Test auf einfache Belegung).

Die Varianten wurden so ausgewählt, dass jeweils die verschiedenen Einstellungen im Disponenten und der Signalvorschaubereich bzw. die Fahrwegsreservierung miteinander kombiniert werden. Dem hinzuzufügen ist, dass die Varianten nicht konkret vorher bestimmt wurden, sondern vielmehr durch einen iterativen Prozess, durch die Ergebnisse der vorherigen Variante, entstanden sind.

Den Ergebnissen der Simulation vorwegzunehmen ist, dass keine der Varianten als optimal zur Deadlock-Vermeidung angesehen werden kann. Mitunter erzielt eine Variante in einem Szenario ein sehr gutes Ergebnis, jedoch im nächsten Szenario nicht mehr. Tabelle 4 stellt die Ergebnisse der Simulation dar.

Nr.	Variantenname	Szen. 1	Szen. 2	Szen. 3	Szen. 4	Szen. 5	Netz
		[%]	[%]	[%]	[%]	[%]	[%]
0	ohne	>99	42	82	85	2	>99
1	Standard 00	>99	40	54	77	41	>99
2	Maximum 00	99	65	17	27	>99	>99
3	Maximum 01	90	29	55	66	<1	>99
4	Standard 01	>99	<1	25	>99	>99	>99
5	Standard 02	>99	>99	90	38	<1	>99
6	Standard 03	>99	8	5	38	>99	>99

**Tabelle 4: Deadlock-Wahrscheinlichkeiten durch Methoden in RailSys**

Erstmalig wurde in Szenario 1 ein Ergebnis erzielt, dass eine Wahrscheinlichkeit für einen Deadlock von annähernd 100% unterschreitet. Die Varianten 2 und 3 haben jeweils in den Disponentenparametern extremere Werte, als in den Standardeinstellungen. D.h. der Vorschaubereich ist im Wert erhöht. Dadurch werden Konflikte früher erkannt. Der Handlungsbereich ist erhöht. Dadurch kann

der Disponent früher einen Konflikt lösen. Und die Mindestverspätung wird verkleinert. Durch diese Einstellung wird versucht auch Konflikte mit geringer Wirkung zu lösen. In Variante 3 wird ein Ergebnis von 90% für die Wahrscheinlichkeit eines Deadlocks erzielt. Hier wurde die Fahrwegsreservierung auf zwei Blöcke gesetzt.

In Szenario 2 ergibt sich ein ähnliches Bild für Variante 2 und 3. Jedoch erzielt das beste Ergebnis die Einstellungen der Variante 4. Wie im Test des Signalvorschaubereiches (Kapitel 5.2.2) werden auch hier vier Blöcke ab dem Ausfahrtsignal auf Belegung geprüft. Daraus folgt eine Deadlock-Wahrscheinlichkeit von unter 1%. Erhöht man jedoch den Wert der Fahrwegsreservierung, wie in Variante 6, wird das Ergebnis auf eine Wahrscheinlichkeit von 6% verschlechtert.

Für Szenario 3 gelten diese Zusammenhänge jedoch nicht. Im Gegensatz zu den Szenarien 1 und 2 scheint es hier vorteilhafter zu sein, den Signalvorschaubereich und die Fahrwegsreservierung auf den gleichen Wert von 4 Blöcken zu setzen. Das beste Ergebnis wird mit Variante 6, mit einer Wahrscheinlichkeit für einen Deadlock von 5% erzielt.

Für das Szenario 4 werden die positiven Ergebnisse wiederum durch die Fahrstraßenreservierung mit vier Blöcken erzeugt. Das um 11% bessere Ergebnis mit Variante 2 gegenüber den Varianten 5 und 6 lässt auf den positiven Einfluss der extremeren Parameter in den Disponenteneinstellungen schließen.

In Szenario 5 wird auch schon ohne den Disponenten und ohne die Definition von Signalvorschaubereich und Fahrwegreservierung ein Ergebnis mit nur 2% Wahrscheinlichkeit für einen Deadlock realisiert. Durch das Einschalten der Disponentenmethoden wird die Wahrscheinlichkeit auf 41% erhöht. Ein Zeichen dafür, dass der Disponent nicht nur Deadlocks vermeiden kann, sondern auch Deadlocks generiert. Die Kombination von Fahrwegsreservierung und Signalvorschaubereich scheint für dieses Szenario von Vorteil zu sein. Jedoch nur, wenn auf Gegenfahrt geprüft wird. Die Deadlock-Wahrscheinlichkeit liegt dann bei unter 1%.

---

Für die Simulation des gesamten Netzes ergibt sich immer noch kein Durchlauf ohne Deadlock. Dies ist nur logisch, da die Einzelwahrscheinlichkeiten für einen Deadlock in den jeweiligen Szenarien immer noch sehr hoch sind.

## **5.4 Fazit über die Methoden in RailSys**

Die Methoden des Disponenten in RailSys sind in ihrer ursprünglichen Art nicht zur Vermeidung von Deadlocks implementiert worden. Sie haben vielmehr die Aufgabe die Entscheidungen eines Disponentens, aufgrund von Verspätungen, zu simulieren, um das Verspätungsniveau zu reduzieren. Das bedingt zwar, dass der Betrieb ohne einen Deadlock durchgeführt werden muss, kann jedoch nicht umgesetzt werden, weil die Methoden eine Deadlock-Situation nicht erkennen, bzw. sie erkennen aber keine Methode, die für einen solchen Konflikt zuständig ist.

Die einzige Methode, die Ihre Aufgabe zu annähernd 100% erfüllt, ist die Disposition der eingleisigen Strecken. Diese Methode vermeidet zuverlässig das Entstehen von trivialen Deadlocks (siehe Kapitel 4.3.1). Die große Wirksamkeit resultiert daraus, dass sie nur für diesen Zweck eingeführt wurde.

Zu den Parametereinstellungen des Disponenten ist zu sagen, dass es auf jeden Fall sinnvoll ist, den Vorschaubereich zur Erkennung der Konflikte auf einen angemessenen hohen Wert zu setzen. Eine Orientierung sollte anhand der Zuggeschwindigkeiten und der Streckenlänge zwischen den einzelnen Bahnhöfen erfolgen. Da der Rechenaufwand der Simulation mit Vergrößerung des Vorschaubereiches ansteigt, muss jedoch situationsbezogen eine Entscheidung getroffen werden. Ist ein Netz z.B. so klein, dass die Rechenleistung ausreichend ist, kann der Vorschaubereich so eingestellt werden, dass dem Disponenten zu jeder Zeit alle Konflikte bekannt sind.

Die Mindestverspätung einer Methode sollte nicht zu klein angesetzt werden. Wird dies getan, versucht der Disponent eine größere Anzahl von Konflikten zu lösen, als eigentlich notwendig ist. Weiterhin kann durch eine Dispositionsentscheidung, die aufgrund einer geringen Mindestverspätung entstanden ist, ein weiterer Konflikt entstehen, der eine viel größere Verspätung

---

als der gelöste Konflikt zur Folge hat. Gleiches gilt natürlich auch für die Entstehung von Deadlocks, weil z.B. die geplante Gleisbelegung nicht eingehalten wird.

Der Handlungsbereich einer Methode sollte hingegen nicht zu groß eingestellt werden, da das frühzeitige Lösen eines Konfliktes durch zwischenzeitlich auftretende Verspätungen durch die gestörten Fahrpläne die Lösung des Disponenten unwirksam machen können.

Allgemein kann gesagt werden, dass die Vorgaben durch die Quelle [RMC05] ein gutes Ergebnis erzeugen. Die Werte sollten jedoch an die Gegebenheiten der Infrastruktur und der Fahrzeuge angepasst werden.

Die Fahrwegsreservierung kann bei alleiniger Anwendung keine Verbesserung für die Deadlock-Betrachtung herbeiführen. In einigen Szenarien kommt es sogar zu einer Erhöhung der Deadlock-Wahrscheinlichkeit. Erst in Kombination mit der Anwendung des Signalvorschaubereiches kommt es hier zu positiven Ergebnissen. Jedoch sollte hier immer mitberücksichtigt werden, welche Nachteile der Einsatz dieser Methoden mit sich bringt. Denn auch hier gilt, dass diese Beeinflussungen der Zugfahrten nicht zur Deadlock-Vermeidung im Programm zur Verfügung stehen, sondern einen anderen Zweck verfolgen.

Der Signalvorschaubereich erzielt eine Wirksamkeit auch ohne die Fahrwegsreservierung. Mit diesem Vorgehen können Deadlocks z.B. in Szenario 2 zu annähernd 100% vermieden werden. Jedoch unter Berücksichtigung, dass die Leistungsfähigkeit der Strecke eingeschränkt wird. Nimmt man diesen Nachteil in Kauf, sollte die Anzahl der Blöcke des Signalvorschaubereiches der jeweiligen Situation angepasst werden. Für einen Bahnhof wie in Szenario 5 (Bahnhof E) mit Doppelblock, ist es z.B. sinnvoll bei der Einfahrt in den Bahnhof zwei Blöcke des Fahrweges auf Gegenfahrt zu prüfen. Diese zwei Blöcke decken dann genau den Bahnhofsbereich ab. Für die angrenzende Strecke muss dann eine andere Regelung getroffen werden. Z.B. für das Ausfahrtsignal auf eine eingleisige Strecke sollte der Signalvorschaubereich bis in den nächsten Bahnhof hinein reichen. Somit wird sichergestellt, dass im nächsten Bahnhof ein Gleis zur Einfahrt frei ist.

Die Anwendung der abhängigen Kanten stellt ein großes Potenzial für die Vermeidung von Deadlocks dar. Es ist jedoch keine Anwendung, die einfach der Simulation zugeschaltet werden kann, wie z.B. der Disponent. Hier müssen zusätzliche Überlegungen stattfinden, wie mit dieser Methode Deadlocks vermieden werden können.

Zusammenfassend lässt sich sagen, dass RailSys nur bedingt in der Lage ist, Deadlocks zu vermeiden. Die Untersuchung wurde auf Grundlage eines konfliktfreien Fahrplans durchgeführt, also liegt die geringe Vermeidungsquote nicht am konstruierten Fahrplan. Die Methoden, die von RailSys bereitgestellt werden, arbeiten alle für ihren Bereich zuverlässig, sind jedoch nicht für die Vermeidung von Deadlock ausgelegt. Insbesondere die Deadlocks in den Szenarien 1, 2 und 3 konnten durch Anwendung des Disponenten und anderer Verfahren nicht ausreichend gelöst werden. Zum Teil traten sogar nach Zuschalten des Disponenten Deadlocks mit einer höheren Wahrscheinlichkeit auf, als ohne die Methoden. Vielversprechend ist der Einsatz von abhängigen Kanten anzusehen. Jedoch nur mit vom Benutzer erstellten Konzepten, zur Lösung bestimmter Konflikte.

---

---

## 6 Deadlock-Vermeidung nach der Literatur

In diesem Kapitel werden zwei Methoden zur Deadlock-Vermeidung nach Pacht vorgestellt. Die Methoden werden jeweils prinzipiell beschrieben und anschließend an Beispielen erläutert. Die Inhalte beziehen sich auf die Quellen [PAC07] und [PAC97].

### 6.1 Movement Consequence Analysis (MCA)

#### 6.1.1 Beschreibung der Methode MCA

Bei dem Verfahren MCA wird vor jeder Zugbewegung in einen Abschnitt (Block) untersucht, ob das Befahren dieses Abschnittes von dem betroffenen Zug einen Deadlock in dem System verursachen würde. Hierzu wird eine Ereigniskette gebildet, die als Ergebnis feststellt, ob ein Deadlock vorliegt oder nicht. Liegt als Ergebnis kein Deadlock vor, kann die Zugbewegung in den angefragten Abschnitt stattfinden. Kommt die Ereigniskette zu dem Ergebnis, dass die Zugbewegung einen Deadlock zur Folge hat, findet die Zugbewegung nicht statt und für den nächsten Zug wird eine Ereigniskette gebildet.

Die Grundidee resultiert aus der Tatsache, dass eine Zugbewegung in einen Block eine weitere Zugbewegung eines anderen Zuges zur Folge hat. Wiederum hat diese Zugbewegung auch eine Zugbewegung des nächsten Zuges zur Folge, usw.. So wird die Ereigniskette gebildet. Ein negatives Ergebnis der Ereigniskette ist, wenn eine Kette geschlossen wird, wie in Kapitel 4.1 beschrieben ist (siehe Abbildung 15). Ist dies der Fall, ist eine Zugbewegung von einer zuvor geplanten Zugbewegung abhängig und ein Deadlock würde entstehen.

Es ist sichergestellt, dass für mindestens einen Zug eine Ereigniskette positiv durch die vorangegangene Kontrolle durch MCA, verläuft. Denn wenn keiner der betroffenen Züge in einen angefragten Abschnitt wechseln darf, entspricht diese Situation einem Deadlock, der erkannt hätte werden müssen.

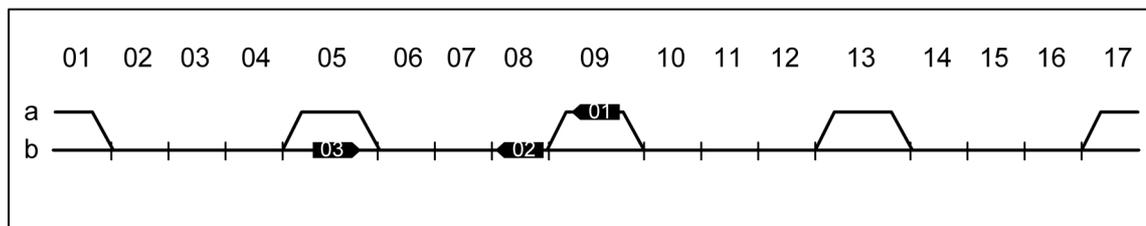
Eine Ausnahme betrifft die Situation, wenn eine Simulation gestartet wird und von Anfang an ein Deadlock im „Deadlock gefährdeten Bereich“ besteht.

---

Die Zugbewegungen als Zugfolgen unterteilen sich in zwei Gruppen:

- Primäre Zugfolgen
- Sekundäre Zugfolgen

Primäre Zugfolgen sind diese, die eine Zugbewegung eines anderen Zuges voraussetzen. Sekundäre Zugfolgen sind solche, die von einem Zug durchgeführt werden müssen, nachdem eine primäre Zugfolge stattgefunden hat, um z.B. den Fahrweg für die nächste Zugfolge zu räumen. Somit kann eine Zugbewegung primäre und sekundäre Zugfolge zugleich sein. Um den Unterschied der Zugfolgen zu verdeutlichen, wird im Folgenden ein Beispiel betrachtet. Abbildung 35 stellt eine Situation mit drei Zügen dar.



**Abbildung 35: Darstellung der primären und sekundären Zufolge**

Ein Beispiel für eine primäre Zugfolge ist, wenn Zug 03 in Block 06b einfahren möchte, Zug 02 zuvor den Block 08b räumen muss, bis er in Block 05a einfährt. Gleiches gilt für die Zugbewegung von Zug 01 in den Block 08b. Auch hier muss Zug 02 räumen, als primäre Zugfolge.

Eine sekundäre Zugfolge ist hingegen, wenn Zug 02 in Block 05a angekommen ist und Zug 03 in Abschnitt 08b vorgefahren, dass Zug 03 in Block 09b weiter fahren muss, um den Weg für Zug 01 frei zu machen. Bezogen auf Zug 02 ist die Zugbewegung von Zug 03 sekundär und bezogen auf Zug 01 primär.

Für die beschriebene Situation ergibt sich nun folgende Ereigniskette für Zug 01, die für die erste Zeile so zu deuten ist: Zug 01 (Z01) möchte in Block 08b ((B08b)) einfahren. Um diese Zugbewegung durchzuführen, muss Zug 02 (Z02) in den Block 07b ((B07b)) einfahren. Die weiteren Zeilen sind diesem Schema nachzudeuten.

Z01 (B08b) -> Z02 (B07b) primäre Zugfolge

Z01 (B08B) -> Z01 (B07b) sekundäre Zugfolge

Z01 (B07b) -> Z02 (B06b) primäre Zugfolge

Z01 (B07b) -> Z01 (B06b) sekundäre Zugfolge

Z01 (B06b) -> Z02 (B05a) primäre Zugfolge

Z01 (B06b) -> Z01 (B05a) sekundäre Zugfolge

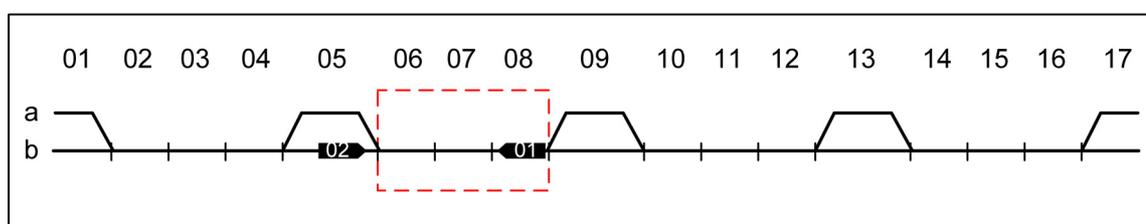
Z01 (B05a) -> Z02 (B04b) primäre Zugfolge

Zug 01 ist nun in seinem Zielblock virtuell angekommen und die Ereigniskette wurde dabei nicht geschlossen. Somit könnte die Zugbewegung von Zug 01 stattfinden. Natürlich muss für jede Bewegung, die Zug 02 durchführt, ebenfalls eine Ereigniskette gestartet werden. Verlaufen beide Ereignisketten positiv, ohne geschlossen zu werden, kann die Zugbewegung der beiden Züge stattfinden.

### 6.1.2 Test der Methode MCA

Um zu überprüfen, ob die Methode Deadlocks vermeiden kann, wird sie anhand der in den Szenarien aufgetretenen Deadlocks getestet. Hierbei wird das System immer ab dem Zeitpunkt betrachtet, bei dem ein Zug kurz vor der Einfahrt in den „Deadlock gefährdeten Bereich“ steht und somit eine Deadlock-Situation herstellen könnte.

Als erstes wird ein trivialer Deadlock betrachtet. Auch wenn diese Art von Deadlocks in RailSys relativ zuverlässig verhindert wird, sollte die Methode in der Lage sein, einen solchen Konflikt zu lösen. Denn durch eine falsche Einstellung in RailSys kann ein solcher Deadlock in RailSys ebenfalls entstehen. Die Situation stellt sich wie in Abbildung 36 gezeigt dar.



**Abbildung 36: Vermeidung eines trivialen Deadlocks durch MCA**

Zug 02 steht in Block 05b und möchte in den Abschnitt 09 wechseln, also in den nächsten Bahnhof. Die Methode überprüft anhand der Ereigniskette nun, ob

diese Zugbewegung einen Deadlock erzeugen könnte. Die Ereigniskette sieht dann für Zug 02 wie folgt aus:

Z02 (B06b) -> Z02 (B07b) sekundäre Zugfolge

Z02 (B07b) -> Z02 (B08b) sekundäre Zugfolge

Z02 (B08b) -> Z01 (B07b) primäre Zugfolge

Z01 (B07b) -> Z02 (B08b) primäre Zugfolge -> Deadlock

Die Ereigniskette für Zug 02 führt zu einem Deadlock, weil in den beiden letzten Zeilen die Züge aufeinander warten. Diese Zugbewegung wird nicht durchgeführt. Die Methode betrachtet nun den Zug 01, ob seine Bewegung ebenfalls einen Deadlock verursachen würde. Zug 01 möchte in Block 05a wechseln.

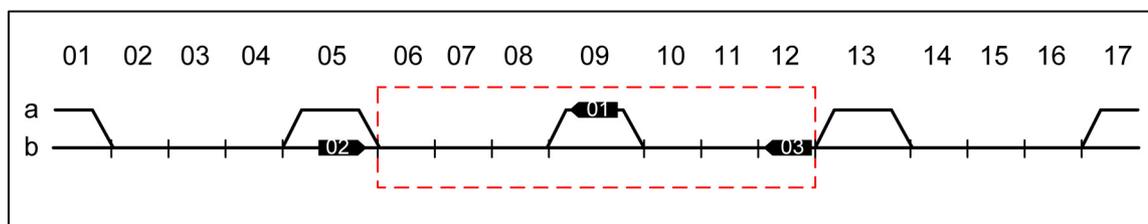
Z01 (B08b) -> Z01 (B07b) sekundäre Zugfolge

Z01 (B07b) -> Z01 (B06b) sekundäre Zugfolge

Z01 (B06b) -> Z01 (B05a) sekundäre Zugfolge

In diesem Augenblick sind beide Züge nicht mehr voneinander abhängig und die Ereigniskette hat sich nicht geschlossen. Somit kann diese Zugbewegung, bis in den Block 05a stattfinden.

Die nächste Kategorie von Deadlocks, die mit der Methode untersucht werden soll, ist die, in der drei Züge durch eine Überholung in einem Bahnhof einen Deadlock erzeugen. Abbildung 37 stellt eine solche Situation dar. Zug 03 möchte Zug 01 durch die Einfahrt in Block 09b überholen.



**Abbildung 37: Vermeidung eines Deadlocks aus Parallelfahrt durch MCA**

Da Zug 03 schon in den eingleisigen Abschnitt eingefahren ist, gilt als sekundäre Zugfolge für Zug 01, dass Zug 03 den gewünschten Block 09b erreicht. Bzw. die Ereigniskette für Zug 03 liefert schon ein positives Ergebnis für die Zugbewegung

in Block 09b. Die Ereigniskette muss also erst ab dem Augenblick geprüft werden, ab dem Zug 03 in seinen Zielblock eingefahren ist.

Z02 (B05b) -> Z02 (B06b) sekundäre Zugfolge

Z02 (B06b) -> Z02 (B07b) sekundäre Zugfolge

Z02 (B07b) -> Z02 (B08b) sekundäre Zugfolge

Z02 (B08b) -> Z02 (B09b) sekundäre Zugfolge

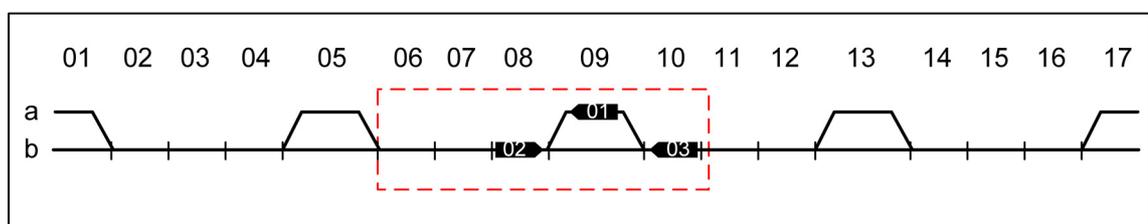
Z02 (B09b) -> Z01 (B08b) primäre Zugfolge

Z01 (B08b) -> Z02 (B09b) primäre Zugfolge -> Deadlock

Die Ereigniskette für Zug 02 führt zu einem Deadlock. Für das Überholmanöver von Zug 03, also Richtung Block 05a, schließt sich die Ereigniskette nicht.

Bei diesem Beispiel wird vorausgesetzt, dass Zug 03 den Bahnhof als erstes erreicht, aufgrund dessen, dass Zug 03 einen der angrenzenden eingleisigen Abschnitte zuerst befahren hat. Dies ist jedoch noch kein Indiz dafür, dass er den Bahnhof um Abschnitt 09 auch als erstes erreicht. Somit hätte Zug 02 doch in den „Deadlock gefährdeten Bereich“ einfahren können, wenn sichergestellt werden würde, dass er als erster den Block 09b erreicht. Dieses kann ebenfalls durch die Methode überprüft werden.

Geht man davon aus, dass die Zugbewegungen der Züge 02 und 03 auf den eingleisigen Streckenabschnitten zunächst unabhängig voneinander sind, wird von der Methode erst bei der Einfahrt in den Bahnhof geprüft, ob ein Deadlock entsteht. Abbildung 38 stellt diese Situation dar.



**Abbildung 38: Vermeidung eines Deadlocks durch MCA, keine Parallelfahrt**

Zunächst wird für Zug 03 die Zugbewegung in den Bahnhof geprüft. Die Ereigniskette stellt sich wie folgt dar:

Z03 (B10b) -> Z03 (B09b) sekundäre Zugfolge

Z03 (B09b) -> Z03 (B08b) sekundäre Zugfolge

Z02 (B08b) -> Z02 (B09b) primäre Zugfolge

Z02 (B09b) -> Z03 (B08b) primäre Zugfolge -> Deadlock

Die Ereigniskette führt zu einem Deadlock. Also wird die Zugbewegung von Zug 02 geprüft.

Z02 (B08b) -> Z02 (B09b) sekundäre Zugfolge

Z02 (B09b) -> Z02 (B10b) sekundäre Zugfolge

Z03 (B10b) -> Z03 (B09a) primäre Zugfolge

Z01 (B09a) -> Z01 (B08b) primäre Zugfolge

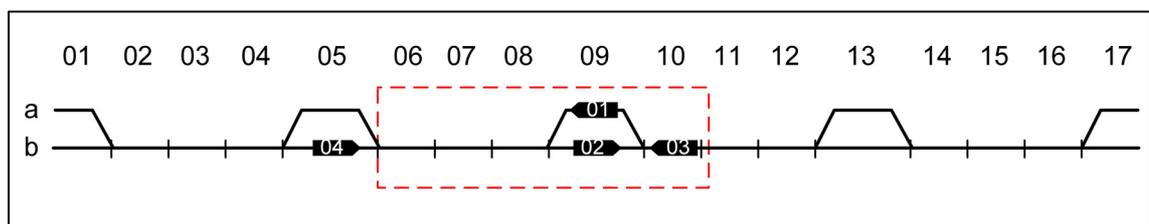
Z01 (B08b) -> Z01 (B07b) sekundäre Zugfolge

Z01 (B07b) -> Z01 (B06b) sekundäre Zugfolge

Z01 (B06b) -> Z01 (B05a) sekundäre Zugfolge

In diesem Fall findet zwar die Überholung nicht statt, jedoch wird ein Deadlock verhindert.

Eine weitere Art von Deadlocks, die vor allem in den Szenarien 02 und 03 auftritt, ist ein Deadlock durch Folgefahrt (siehe Kapitel 4.3.3). Bei dieser Art von Deadlock verklemmen sich die Züge im Bahnhof unter der Bedingung, dass mindestens ein Zug eine andere Fahrtrichtung als die anderen Züge im Bahnhof hat. Abbildung 39 stellt die Situation für einen zweigleisigen Bahnhof dar.



**Abbildung 39: Vermeidung eines Deadlocks aus Folgefahrt mit MCA**

Es gilt zu prüfen, ob Zug 04 in den Block 06b einfahren darf. Hierzu wird die Ereigniskette aufgestellt.

Z04 (B05b) -> Z04 (B06b) sekundäre Zugfolge

Z04 (B06b) -> Z04 (B07b) sekundäre Zugfolge

Z04 (B07b) -> Z04 (B08b) sekundäre Zugfolge  
Z04 (B08b) -> Z04 (B09b) sekundäre Zugfolge  
Z02 (B09b) -> Z02 (B10b) primäre Zugfolge  
Z03 (B10b) -> Z03 (B09a) primäre Zugfolge  
Z01 (B09a) -> Z01 (B08b) primäre Zugfolge  
Z04 (B08b) -> Z04 (B09b) primäre Zugfolge -> Deadlock

Die Einfahrt von Zug 04 in Block 06b wird also nicht gestattet, weil sie zu einem Deadlock führt.

Auch in komplexen Situationen werden Deadlocks zuverlässig erkannt. Das Bilden der Ereignisketten ist dann nur aufwendiger und dauert dementsprechend länger.

### **6.1.3 Auswertung von MCA**

Die Methode MCA ist in der Lage 100% aller Deadlocks zu erkennen und diese zu lösen. Hierbei kann es jedoch vorkommen, dass aufgrund der Einteilung in primäre und sekundäre Zugfolgen nicht immer eine optimale Lösung gefunden wird. Diesem Argument steht gegenüber, dass eine Deadlock-Vermeidung grundsätzlich losgelöst von Prioritäten erfolgen kann. D.h. ein sich bewegendes System ist wichtiger als die Priorität der Züge.

Damit eine 100%tige Erkennung gewährleistet ist, müssen jedoch in eine Ereigniskette auch Züge miteinbezogen werden, die noch sehr weit vom eigentlichen Konfliktzug entfernt sind. Bis jetzt gibt es keine Untersuchungen, welcher Vorschaubereich für diese Methode sinnvoll ist.

Ein weiterer Nachteil ist, dass nur der Regelfahrweg aller Züge geprüft werden kann. Spontane Zielgleisänderungen sind nicht möglich. Wird eine Ereigniskette geprüft, muss für jeden beteiligten Zug feststehen, welchen Fahrweg er nimmt. Ein weiteres Problem ist die Komplexität der Untersuchung. Es muss für jede Zugsbewegung eine Ereigniskette gebildet werden. Somit steigt der Rechenaufwand der Simulation.

---

Weiterhin ist die Frage, ob es erwünscht ist, in einer Simulation eine Methode zu verwenden, die analytisch das Problem zu lösen versucht und somit alle Deadlocks lösen kann.

## **6.2 Dynamic Route Reservation (DRR)**

### **6.2.1 Beschreibung der Methode DRR**

Im Gegensatz zur MCA, welches als Verfahren sehr starr ist, wird mit DRR ein dynamischer Ansatz gewählt. Hier wird nicht nur ein Belegungsabschnitt in die Zukunft vorausgeschaut, sondern es werden für jeden Zug mehrere Abschnitte, die seinem Fahrweg entsprechen, reserviert. Die Anzahl, wie viele Abschnitte ein Zug reservieren muss, bevor er den nächsten Abschnitt befahren darf, ist von definierten Regeln abhängig (s.u.).

Um eine größere Flexibilität der Zugbewegungen zu gewährleisten, werden zwei Vereinfachungen gegenüber MCA getroffen, die die sekundären Zugfolgen betreffen.

1. Ein Zug, der in einen Streckenblock im Zweirichtungsbetrieb einfährt, ist von den Zugbewegungen ihm entgegenkommender Züge unabhängig.
2. Ein Zug, der in einen Stationsblock einfährt, ist immer von ihm entgegenkommenden Zügen abhängig.

Durch die größere Flexibilität der Zugfolgenplanung ergibt sich jedoch der Nachteil, dass nicht mehr 100% der Deadlocks vermieden werden können.

Ein Block kann von mehreren Zügen gleichzeitig reserviert werden. Die zeitliche Reihenfolge, in der die Züge den betrachteten Abschnitt reservieren können, ist maßgeblich. D.h. ein Zug darf einen Abschnitt zu seinem reservierten Fahrweg zählen, wenn er in der Reservierungsliste für diesen Abschnitt ganz oben steht. Ist dies für alle Abschnitte, die reserviert werden müssen, der Fall, darf der Zug den nächsten Abschnitt befahren. Ein Zug darf in einen Block nicht einfahren wenn dieses regulär nicht möglich ist und wenn er in der Liste der Belegungsabfragen nicht ganz oben steht.

---

---

Die nachfolgende Liste von Regeln bestimmt unter anderem die Anzahl der zu reservierenden Blöcke des Fahrweges. Weiterhin bestimmen die Regeln, ob eine Reservierung des Fahrweges stattfinden darf. Aufgrund von weiteren Erkenntnissen des Prof. Pachl wächst diese Liste ständig.

1. Bevor ein Zug in einen Block einfahren darf, muss der Weg in diesen Block für ihn reserviert sein.
2. Wenn ein Fahrweg reserviert wird, der in einen Block mit Zweirichtungsbetrieb führt, der kein Stationsblock ist, dann muss der Weg, um diesen Block zu verlassen, ebenfalls reserviert werden.
3. Ist ein zureservierender Block durch einen anderen Zug reserviert oder belegt, wird die gewünschte Reservierung in der Liste unten angestellt.
4. Wenn ein Zug einen Block reservieren möchte, der von einem anderen Zug reserviert ist, muss der Zug, der den Block als erstes reserviert hat, eine weitere Reservierung durchführen, um den zu reservierenden Block für den nachfolgenden Zug zu räumen.
5. Ist ein zu reservierender Stationsblock durch einen in Gegenrichtung fahrenden Zug reserviert oder belegt, der wiederum in den Block des anfragenden Zuges einfahren möchte, scheitert die Anfrage auf Reservierung.
6. Eine Anfrage auf Reservierung wird nur wirksam, wenn alle folgenden Reservierungen, die durch die Regeln 2 und 4 nötig werden, reserviert werden können.

Mit diesen Regeln wird etwas ähnliche versucht, wie es in RailSys der Signalvorschaubereich und die Fahrwegsreservierung abbilden. Nur dass hier die Anzahl der zu reservierenden Blöcke der Situation entsprechend variieren kann. Weiterhin können auch Vorreservierungen stattfinden, die in einer Liste für jeden Block gespeichert werden.

Um die Flexibilität noch weiter zu erhöhen, wird noch eine weitere Regel aufgestellt. Diese ist jedoch noch nicht ausreichend getestet und wird aus diesem Grund hier nur unter Vorbehalt erwähnt und angewandt.

---

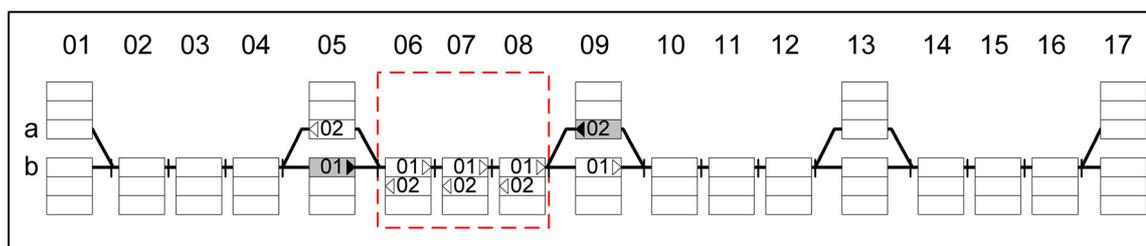
7. Wenn ein Zug in einem Stationsblock endet, in dem er in der Reservierungsliste ganz oben steht, kann er in den Reservierungslisten der vorliegenden Blöcke auch an die erste Stelle gesetzt werden.

Im nachfolgenden Kapitel werden diese Regeln auf ihre Wirksamkeit hin zur Deadlock-Vermeidung untersucht.

### 6.2.2 Test der Methode DRR

Um zu überprüfen, ob die Methode in der Lage ist, die in den Szenarien aufgetretenen Deadlocks erfolgreich zu verhindern, wird in diesem Kapitel die Methode in ihrer Funktion dargestellt und getestet. Es werden die gleichen Deadlock-Beispiele herangezogen, wie in Kapitel 6.1.

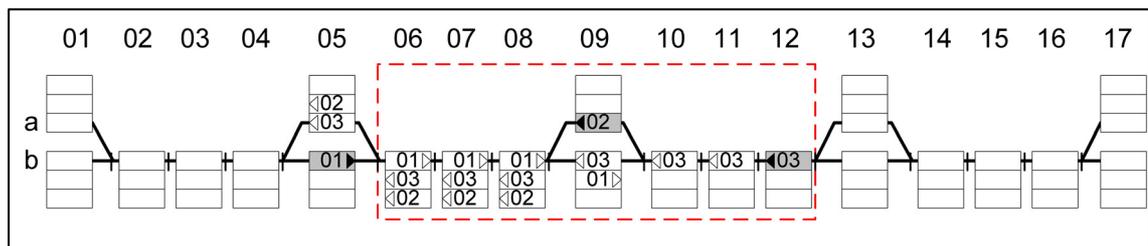
Zunächst muss untersucht werden, ob die Methode in der Lage ist, einen trivialen Deadlock zu verhindern. Abbildung 40 stellt die Ausgangssituation für einen trivialen Deadlock dar. Die grau hinterlegten Felder zeigen die aktuelle Position des jeweiligen Zuges. Die weiß hinterlegten Felder sind die Reservierten. Der Pfeil gibt die Fahrtrichtung an. Der Zug, der in dem Feld einer Liste auf dem Gleis angezeigt wird, hat die aktuelle Reservierung des Blockes.



**Abbildung 40: Vermeidung eines trivialen Deadlocks durch DRR**

In diesem Beispiel hat Zug 01 die Blöcke 06b, 07b und 08b als erstes reserviert, weil für ihn als erstes eine Zugbewegung beschlossen wird. Dadurch, dass Zug 02 in der Reservierungsliste an zweiter Stelle steht, kann er erst diese Blöcke befahren, wenn Zug 01 im Block 09b angekommen ist. Somit kann ein trivialer Deadlock nicht stattfinden, weil die Begegnungsfahrt auf dem eingleisigen Abschnitt verhindert wird. Die Reservierungen der Fahrwege basieren in diesem Fall auf den Regeln 1, 2 und 3. Regel 2 ist dafür verantwortlich, dass der jeweilige Fahrweg bis in den nächsten Bahnhof führt. In diesem Beispiel wäre es auch möglich die Regel 7 anzuwenden, da Zug 02 im Zielblock an erster Stelle steht.

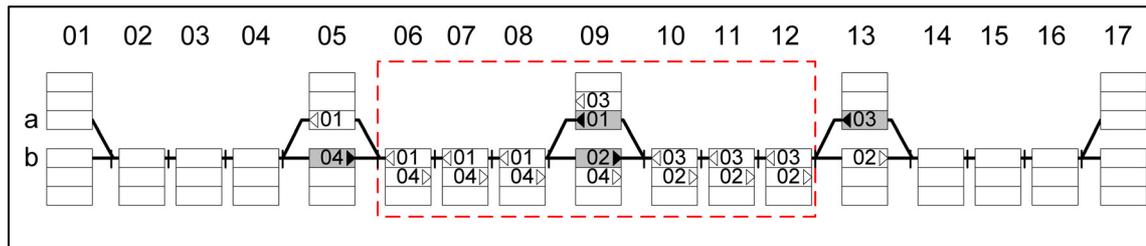
Das nächste Beispiel, welches betrachtet wird, ist eine Deadlock-Situation mit drei Zügen, wobei in einem Bahnhof eine Überholung bzw. eine Parallelfahrt stattfindet. Abbildung 41 stellt die Situation mit Reservierungslisten noch einmal dar. Zug 03 möchte Zug 02 im Bahnhof 09 überholen. Zug 01 möchte Zug 02 im Bahnhof 09 kreuzen. Aufgrund dessen, dass Zug 03 vor Zug 02 in den „Deadlock gefährdeten Bereich“ eingefahren ist, kann er als erstes den Block 09b belegen. Die Reservierung bis dorthin wird durch die 2. Regel nötig. Kurz nach der Reservierung durch Zug 03 reserviert Zug 01 mit derselben Regel seinen Fahrweg in Block 09b. Weil Block 09b jedoch durch Zug 03 reserviert ist, steht Zug 01 für diesen Block nicht ganz oben in der Liste. Als nächstes tritt für Zug 03 Regel 4 in Kraft. Er muss einen Fahrweg reservieren, um Block 09b zu verlassen, damit Zug 01 ihn belegen kann. Diese Reservierung führt bis in Block 05a. Damit diese Zugbewegung stattfinden kann, muss Regel 7 für die Blöcke 06b, 07b und 08b bezogen auf Zug 03 Anwendung finden. Als letzte Zugbewegung verlässt Zug 02 seine Position aus Block 09b.



**Abbildung 41: Vermeidung eines Deadlocks aus Parallelfahrt durch DRR**

In diesem Beispiel kann ein Deadlock, der ohne diese Methode entsteht, verhindert werden. Die Zugfolgen, die sich durch die Methode ergeben, sind sinnvoll und mit den Entscheidungen eines Disponenten zu vergleichen.

Ein weiteres Beispiel, um die Methode zu testen, ist die Deadlock-Situation durch eine Folgefahrt, bei der mindestens vier Züge beteiligt sind. Abbildung 42 stellt diese Situation mit Reservierungslisten für jeden Block noch einmal dar.



**Abbildung 42: Vermeidung eines Deadlocks aus Folgefahrt durch DRR**

Zug 02 und Zug 01 führen eine Kreuzung in Bahnhof 09 durch, wobei jeder der beiden Züge von einem anderen Zug verfolgt wird. Es gilt hier zu verhindern, dass beide folgenden Züge in den „Deadlock gefährdeten Bereich“ gleichzeitig einfahren. Durch die Anwendung von DRR fährt in den „Deadlock gefährdeten Bereich“ keiner der beiden folgenden Züge ein. Solange keines der beiden Stationsgleise im Bahnhof 09 frei ist, wird dies durch Regel 1 und Regel 2 verhindert. Somit kann z.B. erst nach dem Verlassen des Blockes 09a von Zug 01, eine Zugbewegung von Zug 03 stattfinden, weil erst dann der gesamte Fahrtweg reserviert ist. Es wäre jedoch theoretisch möglich, einen der beiden folgenden Züge in den „Deadlock gefährdeten Bereich“ einfahren zu lassen, ohne dass ein Deadlock entsteht. Eine Folgefahrt von Zügen gleicher Richtung auf einer eingleisigen Strecke im Zweirichtungsbetrieb kann jedoch nicht stattfinden.

Diese Eigenschaft hat zur Folge, dass im genannten Beispiel die Kapazität der Strecke eingeschränkt wird, weil Züge nicht dicht aufeinander folgen können. Somit macht eine größere Anzahl von Blöcken (als ein Block) auf einer eingleisigen Strecke keinen Sinn, da sich durch Regel 1 und Regel 2 immer nur ein Zug auf einem eingleisigen Abschnitt befinden darf – egal ob Richtungs- oder Gegenfahrt.

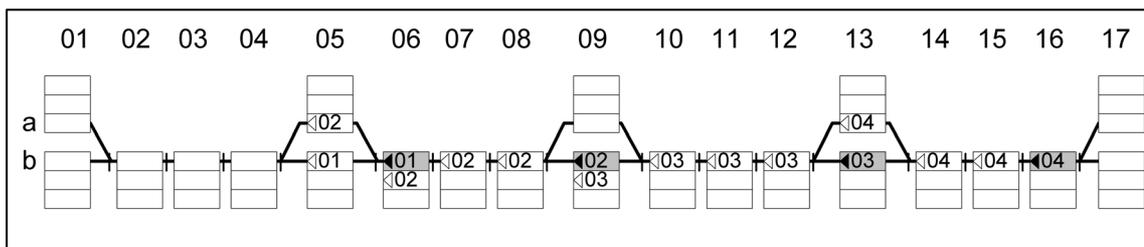
Ein Deadlock wird natürlich auch hier verhindert. Jedoch unter Beeinträchtigung der Leistungsfähigkeit. Durch das Definieren weiterer Regeln kann diese Situation jedoch gelöst werden. Eine Definition solcher ergänzenden Regeln erfolgt in Kapitel 7.2.1.

### 6.2.3 Auswertung von DRR

Das Vermeiden von Deadlocks mit der Methode DRR geschieht in den betrachteten Beispielen zu 100%. Bei der Betrachtung des trivialen Deadlocks

und aus Überholung waren die resultierenden Fahrwege eindeutig nachvollziehbar.

In Deadlock-Situationen, bei denen der Deadlock durch Folgefahrten entsteht, wird der Deadlock auch verhindert, jedoch unter Reduzierung der Leistungsfähigkeit. Natürlich kann auch hier das Argument angeführt werden, dass eine Deadlock-Vermeidung oberste Priorität hat, auch vor der Leistungsfähigkeit der zu untersuchenden Strecke. Jedoch würde hier auch die Leistungsfähigkeit beschnitten werden, wenn keine Deadlock-Gefahr besteht. Um dieses zu zeigen, wird eine solche Situation in einem Beispiel demonstriert. Abbildung 43 stellt die bekannte eingleisige Strecke dar. In diesem Beispiel gibt es nur aufeinander folgende Züge, die vom Bahnhof 17 Richtung Bahnhof 01 jeweils eine Zugfahrt durchführen.



**Abbildung 43: Darstellung der Leistungsfähigkeitsminderung durch DRR**

Die Strecke bietet die Möglichkeit zwischen den Bahnhöfen dicht aufeinander zu folgen. D.h. es existieren jeweils drei Blöcke, die theoretisch jeder von einem Zug belegt werden könnten. Durch die Methode DRR bzw. durch die Anwendung der Regeln 1 und 2 kommt es jedoch zu der Situation, dass ein Zug nur aus einem Bahnhof ausfahren darf, wenn der angrenzende eingleisige Streckenabschnitt vollständig frei ist. Hierbei ist es egal, in welches Gleis (belegt oder frei) ein Zug in einem Bahnhof belegen möchte. Das Beispiel in Abbildung 43 stellt verschiedene Situationen dar. So möchte Zug 02 in ein anderes Bahnhofsgleis in Bahnhof 05, als Zug 01. Jedoch solange Zug 01 auf der eingleisigen Strecke fährt, kann er nicht alle benötigten Blöcke reservieren.

Eine ähnliche Situation entsteht für Zug 03, der im Bahnhof 09 dasselbe Gleis befahren möchte, wie Zug 02. Erst wenn Zug 02 diesen Bahnhofsblock geräumt hat, kann Zug 03 seine Zugbewegung Richtung Bahnhof 09 von Bahnhof 13 beginnen.

Obwohl bei reiner Folgefahrt keine Deadlock-Gefahr besteht, wird die Leistungsfähigkeit der Strecke eingeschränkt.

Ein Vorteil, den die Methode mit sich bringt, ist, dass sie in die Programmstruktur einfach zu implementieren ist, da sie nach einem Prinzip abgewickelt wird, das in RailSys zum Teil schon vorhanden ist. Zugbewegungen dürfen in RailSys nur stattfinden, wenn der nächste Block frei ist. Dieser wird auf Belegung geprüft. Auch die Methoden des Signalvorschaubereiches und der Fahrwegsreservierung, die in RailSys bereits vorhanden sind, verfolgen ein ähnliches Prinzip, welches gegenüber DRR eher starr ist.

---

---

## 7 Entwurf von Verfahrung zur Deadlock-Vermeidung

### 7.1 Ausschlussverfahren durch abhängige Kanten (AAK)

Wie in Kapitel 5.2.3 festgestellt wurde, besitzen die abhängigen Kanten das größte Potenzial als Möglichkeit in RailSys Deadlocks zu vermeiden. Es ist nur keine Methode, die einfach zugeschaltet werden kann, sondern sie muss mit Überlegungen bzw. einer Strategie durch den Anwender ergänzt bzw. kombiniert werden.

In dem hier betrachteten Infrastrukturnetz sind besonders die Szenarien 1 bis 3 zu betrachten. Szenario 4 wurde bereits durch die Anwendung von abhängigen Kanten positiv, hinsichtlich der Deadlocks, beeinflusst. Szenario 5 ist eher uninteressant, weil zum einen auftretenden Deadlock ähnlich verhindert werden können, wie im Szenario 4, und zum anderen nur eine geringe Wahrscheinlichkeit besteht, dass ein Deadlock überhaupt auftritt. Somit wird vordergründig eine ergänzende Strategie für die Deadlock-Vermeidung auf eingleisigen Strecken entworfen.

Um eine Strategie für den Einsatz von abhängigen Kanten entwickeln zu können, müssen die drei Arten von Deadlocks, die zurzeit auf eingleisigen Strecken auftreten können, nochmals betrachtet werden. Die Situationen werden ab dem Zeitpunkt betrachtet, in dem gerade noch keine Deadlock-Situation besteht. D.h. immer kurz vor der Einfahrt eines Zuges in den „Deadlock gefährdeten Bereich“.

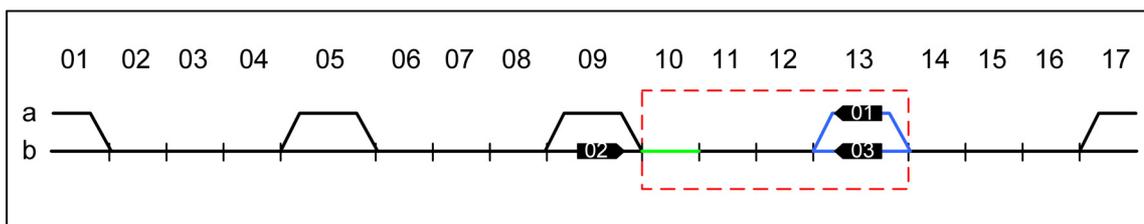
Da triviale Deadlocks von RailSys erfolgreich erkannt und verhindert werden, bleibt diese Betrachtung erst einmal außen vor.

#### 7.1.1 Regeln für Deadlocks aus Überholungen

Zunächst wird eine Deadlock-Situation betrachtet, die mit drei Zügen entstehen kann. Hierfür ist zunächst das Fahrmanöver „Überholung“ verantwortlich. Abbildung 19 zeigt einen solchen Deadlock. Wie in Kapitel 4.3.2 beschrieben ist, kann ein solcher Deadlock aus mehreren Situationen entstehen. Abbildung 20 stellt die erste Situation dar. Hier sind beide Gleise im Bahnhof 13 mit Zügen belegt, die in die gleiche Richtung fahren möchten. Durch das Einfahren des

---

Zuges 02 in den „Deadlock gefährdeten Bereich“ wird die Deadlock-Situation hergestellt. Es gilt zu verhindern, dass Zug 02 in diesen Bereich einfährt. Zug 02 darf nicht in diesen Bereich einfahren, wenn die Gleise im Bahnhof 13 von Zügen belegt sind, die nicht die gleiche Richtung wie er haben. Mit Hilfe der abhängigen Kanten kann diese Überprüfung stattfinden. Abbildung 44 stellt dies schematisch dar. Die grüne Kante im Block 10b ist die Bezugskante, die von logischen Bedingungen abhängig gemacht wird. Diese Bedingungen beziehen sich auf die Kanten im Block 13a und 13b, die mit einem logischen „UND“ miteinander verknüpft werden. D.h. der Block 10b wird für den Zug 02 nur freigegeben, wenn mindestens einer der Blöcke im Bahnhof 13 frei ist oder mindestens ein Zug in die gleiche Richtung wie Zug 02 fährt. Anders ausgedrückt: Die grüne Kante wird gesperrt, wenn beide blauen Blöcke mit Zügen belegt sind, die beide nicht die gleiche Fahrtrichtung haben, wie Zug 02.



**Abbildung 44: Darstellung der Regel 01 der Methode AAK**

Die zu programmierende Logik ergibt sich dann wie folgt: Block 10b ist gesperrt, wenn der folgende Ausdruck logisch „true“ ergibt, wobei die eine richtungsbezogene Belegung durch einen Zug den Zustand „true“ entspricht.

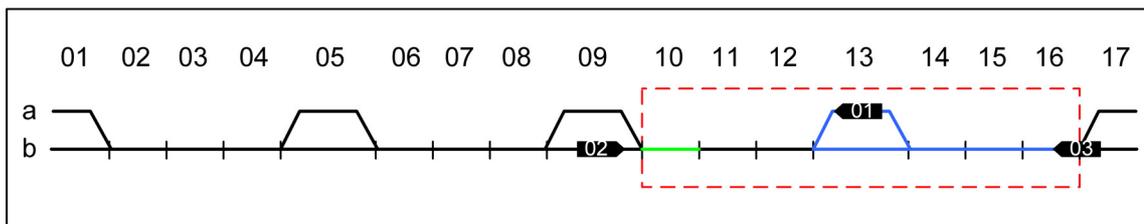
Regel 01:

$$13bGR \wedge 13aGR$$

So wird verhindert, dass Zug 02 in den „Deadlock gefährdeten Bereich“ einfährt, bis einer der beiden Züge (Zug 01 oder Zug 03) den Bahnhof 13 verlässt und somit ein Gleis für Zug 02 frei wird.

Ein Deadlock, wie er in Abbildung 19 dargestellt ist, kann jedoch auch aus einer Situation entstehen, wie sie in Abbildung 21 dargestellt wird. Hierbei ist Zug 03 noch nicht im Bahnhof 13 angekommen, sondern befährt gerade den eingleisigen Streckenabschnitt, der zum Bahnhof 13 führt. Diese Situation kann in einem Deadlock enden, muss sie jedoch nicht. Je nachdem welcher der beiden

Züge (Zug 02 oder Zug 03) den Bahnhof 13 zuerst erreicht, kann ein Deadlock entstehen oder nicht. Ist Zug 02 schneller im Bahnhof, wird ein Deadlock verhindert. Um jedoch sicher einen Deadlock zu verhindern, sollte Zug 02 nur den eingleisigen Streckenabschnitt befahren dürfen, wenn mindestens ein Gleis im Bahnhof 13 frei ist und kein Zug auf dem angrenzenden Streckenabschnitt (Blöcke 14b, 15b und 16b) in Gegenrichtung fährt. Diese Bedingung wird ebenfalls durch abhängige Kanten umgesetzt. Abbildung 45 stellt schematisch dar, welche Kanten für diese Situation verknüpft werden müssen, um einen Deadlock zu verhindern. Die Einfahrt in den Block 10b (grüne Kante) von Zug 02 darf nur erfolgen, wenn alle Blöcke des eingleisigen Streckenabschnittes frei sind und mindestens ein Bahnhofsgleis frei ist. Es muss gewährleistet sein, dass wenn beide Gleise im Bahnhof frei sind, eine Zugbewegung beider Züge (Zug 02 und Zug 03) in den Bahnhof stattfinden kann, um eine Kreuzung durchzuführen.



**Abbildung 45: Darstellung der Regel 02 der Methode AAK**

Die zu programmierende Logik ergibt sich dann wie folgt: Block 10b ist gesperrt, wenn der folgende Ausdruck logisch „true“ ergibt.

Regel 02:

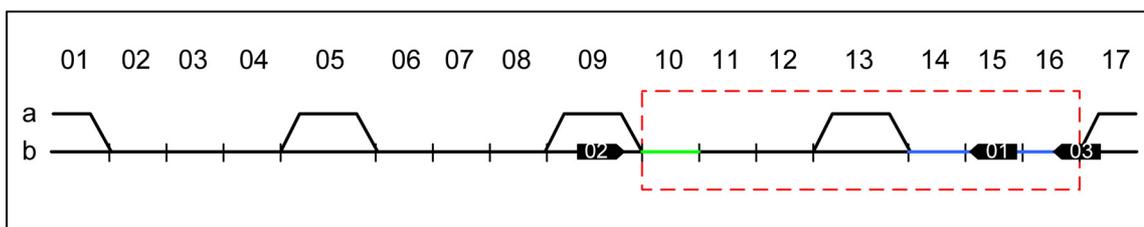
$$((14bGR \vee 15bGR \vee 16bGR) \wedge 13aGR) \vee ((14bGR \vee 15bGR \vee 16bGR) \wedge 13bGR)$$

Bei Betrachtung dieser Situation ergibt sich eine weitere Möglichkeit, aus der ein Deadlock entstehen kann. Da die angrenzenden eingleisigen Strecken jeweils aus drei Blöcken bestehen, kann es durchaus sein, dass noch kein Zug im Bahnhof ist, sondern sich mehrere Züge auf der zulaufenden Strecke befinden, die den Bahnhof blockieren könnten. Für diesen Fall greift keine der bis jetzt aufgestellten Bedingungen. Hierfür muss eine neue Logik erstellt werden.

Ein Problem stellt die Erfassung der Anzahl der Züge im eingleisigen Streckenabschnitt dar. Theoretisch können bis zu drei Züge in diesem Abschnitt

aufeinander folgen. Sobald mehr als zwei Züge dies tun, ist eine Zugbewegung von Zug 02 in den eingleisigen Abschnitt nicht mehr erlaubt. Die kritische Anzahl der Züge ist von Kapazität des betrachtenden Bahnhofes abhängig. Da die Anzahl der Züge nicht direkt erfasst werden kann, muss auch hier über die Logik für jede mögliche Situation das Zählen der Züge nachgebildet werden.

Abbildung 46 zeigt eine solche Situation. Die grüne Kante stellt wieder den Block dar, der gesperrt wird, wenn die Bedingungen eine Weiterfahrt nicht zulassen. Mit den blauen Kanten wird die Anzahl der Züge ermittelt, die auf dem eingleisigen Streckenabschnitt fahren.



**Abbildung 46: Darstellung der Regel 03 der Methode AAK**

Die zu programmierende Logik ergibt sich dann wie folgt: Block 10b ist gesperrt, wenn der folgende Ausdruck logisch „true“ ergibt.

Regel 03:

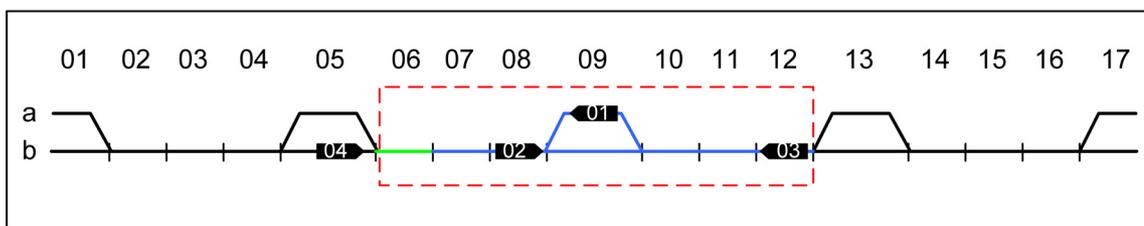
$$(14bGR \wedge 15bGR) \vee (14bGR \wedge 16bGR) \vee (15bGR \wedge 16bGR)$$

In RailSys könnte sich bei dieser Betrachtung das Problem ergeben, dass ein Zug, der von einem in den nächsten Block wechselt, zwei Blöcke belegt. Somit kann die Bedingung auch durch einen fahrenden Zug erfüllt werden. Dieses Problem verändert die Fahrtentscheidung aber nur im Zeitpunkt eines Zuges, der in einen „Deadlock gefährdeten Bereich“ einfahren möchte. Er hat immer die Möglichkeit den Block zu befahren, wenn sich der Zug auf dem gegenüberliegenden eingleisigen Streckenabschnitt in genau einem Block befindet. Es entsteht eine zeitliche Abweichung, jedoch kein totaler Ausfall der Abfrage.

### 7.1.2 Regeln für Deadlocks aus Kreuzungen

Nachdem die Logik zur Vermeidung von Deadlocks aus Überholungen beschrieben wurde, muss nun eine Betrachtung von Deadlocks aus Kreuzungen erfolgen. Hierbei wird genau so verfahren, wie bei den Deadlocks mit drei Zügen.

Zuerst wird definiert, welche Zugbewegung nicht stattfinden darf und dann unter welchen Bedingungen sie nicht stattfinden darf. Abbildung 22 stellt die Ausgangssituation für einen solchen Deadlock dar. Sobald Zug 04 in den „Deadlock gefährdeten Bereich“ eingefahren ist, ist die Deadlock-Situation hergestellt und somit ein Deadlock unausweichlich. Zug 04 darf also nicht einfahren, wenn im Bahnhof beide Gleise belegt sind, wobei mindestens ein Zug in Fahrtrichtung und ein Zug gegen die Fahrtrichtung steht und weiterhin auf dem angrenzenden eingleisigen Streckenabschnitt mindestens ein Zug gegen die Fahrtrichtung von Zug 04 verkehrt. Zug 02 kann sich auch im Zulauf zum Bahnhof 09 befinden. Abbildung 47 stellt die Situation und die betroffenen Kanten dar. Die grüne Kante ist die, die in Abhängigkeit von den blauen Kanten freizugeben oder zusperrt ist.



**Abbildung 47: Darstellung der Regel 04 der Methode AAK**

Die zu programmierende Logik ergibt sich dann wie folgt: Block 06b ist gesperrt, wenn der folgende Ausdruck logisch „true“ ergibt.

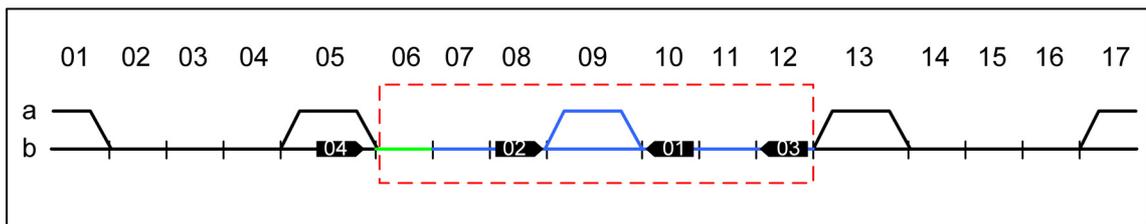
Regel 04:

$$\begin{aligned}
 & ((10bGR \vee 11bGR \vee 12bGR) \wedge 09bGR \wedge 09aR) \vee \\
 & ((10bGR \vee 11bGR \vee 12bGR) \wedge 09bR \wedge 09aGR) \vee \\
 & ((10bGR \vee 11bGR \vee 12bGR) \wedge 08bR \wedge 09aGR) \vee \\
 & ((10bGR \vee 11bGR \vee 12bGR) \wedge 07bR \wedge 09aGR) \vee \\
 & ((10bGR \vee 11bGR \vee 12bGR) \wedge 08bR \wedge 09bGR) \vee \\
 & ((10bGR \vee 11bGR \vee 12bGR) \wedge 07bR \wedge 09bGR)
 \end{aligned}$$

Das Problem, welches sich hier ergibt, ist dass diese Situation nur eintreten kann, wenn Zug 02 vor Zug 03 auf den Bahnhof 09 vom angrenzenden Bahnhof aus zufährt. Denn auch hier ist entscheidend, welcher der beiden Züge zuerst den Bahnhof erreicht. Ist Zug 02 als erstes im Bahnhof, kann Zug 03 erst auf den eingleisigen Streckenabschnitt einfahren, wenn Zug 02 im Bahnhof 09

angekommen ist. Dies wird durch die zuvor aufgestellten Regeln für Deadlocks aus Überholungen bedingt. Theoretisch wäre es auch möglich, dass zwei Züge gleichzeitig auf einen Bahnhof aus unterschiedlicher Richtung zufahren.

Eine weitere Situation, aus der ein Deadlock mit vier Zügen entstehen kann, ist in Abbildung 48 dargestellt. Auch diese Situation kann nur unter bestimmten Bedingungen, die von den bereits definierten Bedingungen abhängig sind, auftreten. Damit diese Situation eintritt, muss gegeben sein, dass Zug 02 in den eingleisigen Abschnitt eingefahren ist, bevor Zug 03 dies tut. Zug 03 kann solange in den eingleisigen Abschnitt einfahren, wie ihm keine zwei Züge auf der angrenzenden eingleisigen Strecke entgegenkommen. Zug 04 darf nicht auf den eingleisigen Streckenabschnitt einfahren, wenn mehr als zwei Züge ihm auf dem angrenzenden eingleisigen Streckenabschnitt entgegenkommen und er einen Zug mit gleicher Richtung vor sich hat, der den Bahnhof oder die zu befahrende Strecke befährt.



**Abbildung 48: Darstellung der Regel 05 der Methode AAK**

Die zu programmierende Logik ergibt sich dann wie folgt: Block 06b ist gesperrt, wenn der folgende Ausdruck logisch „true“ ergibt.

Regel 05:

$$((10bGR \wedge 11bGR) \vee (10bGR \wedge 12bGR) \vee (11bRG \wedge 12bGR)) \wedge (07bR \vee 08bR \vee 09bR \vee 09aR)$$

Regel 04 und 05 könnten auch zusammen aufgestellt werden, weil es jeweils die gleiche Abfrage für Zug 04 ist, ob ihm zwei Züge entgegenkommen und er selbst einem Zug folgt. Der Übersichtlichkeit halber werden diese Regeln getrennt betrachtet.

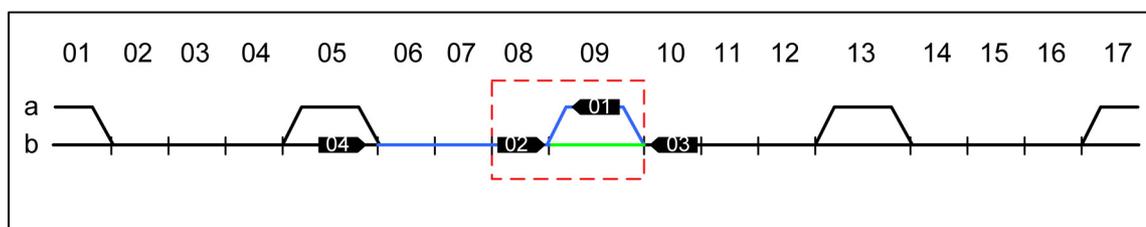
Theoretisch ist es auch hier möglich, dass Zug 01 und Zug 03 den Bahnhof vor Zug 02 erreichen und somit beide Gleise im Bahnhof belegen. Dies würde bei der

Einfahrt von Zug 03 in den Bahnhof die Deadlock-Situation herstellen. Abbildung 49 stellt diese Situation dar. Es muss geprüft werden, ob Zug 03 in den Bahnhof einfahren darf. Er darf nicht einfahren, wenn ihm im Bahnhof nur ein Gleis frei ist und die anderen Gleise von Zügen belegt sind, die in die gleiche Richtung fahren wie er. Weiterhin darf ihm auf der angrenzenden Strecke kein Zug entgegenkommen.

Die zu programmierende Logik ergibt sich dann wie folgt: Block 09b ist gesperrt, wenn der folgende Ausdruck logische „true“ ergibt.

Regel 06:

$$(09aR \vee 09bR) \wedge (06bGR \vee 07bGR \vee 08bGR)$$



**Abbildung 49: Darstellung der Regel 06 der Methode AAK**

### 7.1.3 Regel aus der Implementierung in RailSys

Die in den Kapiteln 7.1.1 und 7.1.2 beschriebenen Regeln werden nun in RailSys implementiert, um die Wirksamkeit zu ermitteln. Natürlich muss dies für jeden Bahnhof und für jede Richtung geschehen, was in den Kapiteln 7.1.1 und 7.1.2 nur beispielhaft an nur einem Bahnhof und für eine Richtung geschehen ist.

Die aufgestellten Regeln zeigen in der Simulation ihre Wirkung. Die oben beschriebenen Deadlocks aus Überholungen und Kreuzungen werden vermieden. Jedoch entstehen in den Simulationsdurchläufen nun neue Deadlocks, die auf eine Korrektur bzw. Ergänzung der bis jetzt aufgestellten Regeln hindeuten, da sie in ihrem Auftreten komplexer sind.

Ein Problem ist, welches nun auftritt, dass Deadlocks entstehen können, bei denen in zwei benachbarten Bahnhöfen Überholungen stattfinden. Somit stehen sich vier Züge vor einer eingleisigen Strecke gegenüber (siehe Abbildung 50). Eine Ursache dafür ist, dass bei der Prüfung der Kanten auf Belegung oder Freisein keine Reservierung des Fahrweges stattfindet, welcher zuvor geprüft wurde. Ein

weiteres Problem stellen Regel 02 und Regel 03 dar. Sie lassen eher eine Überholung als eine Kreuzung als Fahrmanöver zu. Da bei einer Überholung nur drei Züge benötigt werden und bei einer Kreuzung vier Züge, ist die Wahrscheinlichkeit, dass ein Deadlock auftritt, höher, weil weniger Züge benötigt werden.

Es sollten also Regeln definiert werden, die einen Deadlock vermeiden können, jedoch ein vermehrtes Überholen vermeiden. Die bereits aufgestellte Regel 06 tut dies. Hier werden beide Züge auf den Bahnhof zufahren gelassen, wobei der überholende Zug vor der Einfahrt in den Bahnhof anhalten muss. Somit wird dem kreuzenden Zug die Einfahrt in den Bahnhof gewährt. Regel 02 und Regel 03 können also aus der Betrachtung herausgenommen werden.

Das Fahrmanöver „Überholen“ ist jedoch weiterhin möglich. Solange Zug 02 noch nicht auf den eingleisigen Abschnitt eingefahren ist, kann Zug 03 in den Bahnhof einfahren (siehe Abbildung 49).

Da die Regeln so definiert sind, dass alle Fahrmanöver weiterhin möglich sind, muss eine Regel definiert werden, die verhindert, dass Überholungen stattfinden können, die zu einem höherwertigen Deadlock führen. Abbildung 50 stellt beispielhaft eine solche Situation dar. Fährt Zug 04 in den „Deadlock gefährdeten Bereich“ ein, stellt er die Deadlock-Situation in dem Augenblick her, in dem er in den Bahnhof 09 einfährt.

Die zu programmierende Logik ergibt sich dann wie folgt: Die Einfahrt in Bahnhof 09 wird verweigert, wenn der folgende Ausdruck logische „true“ ergibt.

Regel 07:

$$13aGR \wedge 13bGR \wedge (09bR \vee 09aR)$$

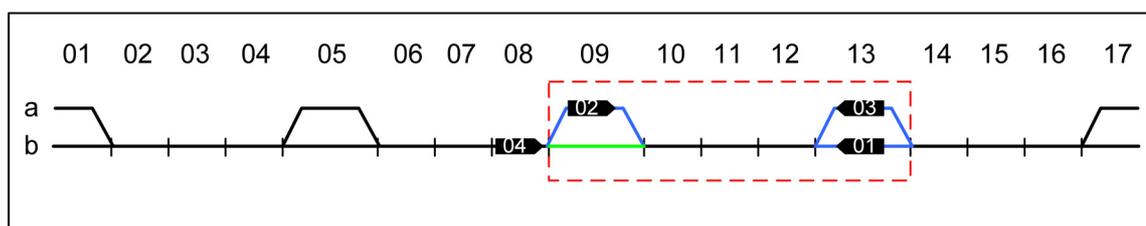


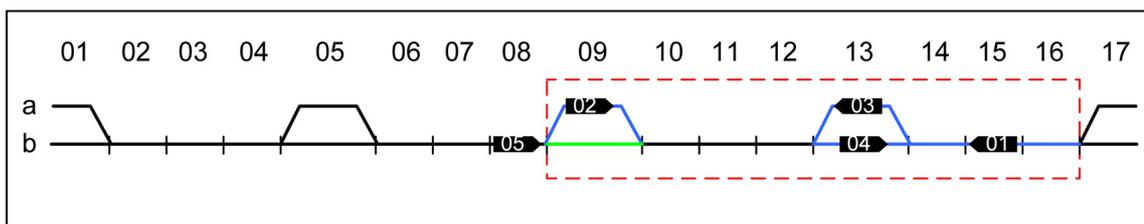
Abbildung 50: Darstellung der Regel 07 der Methode AAK

Es zeichnet sich an dieser Stelle der Arbeit immer mehr ab, dass die Betrachtung der Deadlock-Vermeidung durch die Definition von Regeln für abhängige Kanten nicht nur auf den nächsten bzw. übernächsten Bahnhof ausreicht. Vielmehr muss für die gesamte Strecke, mitunter das gesamte Netz in die Betrachtung für eine Zugbewegung miteinbezogen werden. Um den Beweis für diese Behauptung aufzustellen, wird ein weiteres Beispiel betrachtet. Abbildung 51 zeigt, dass durch die Einfahrt von Zug 05 in Block 09b die Deadlock-Situation hergestellt würde.

Die zu programmierende Logik ergibt sich wie folgt: Die Einfahrt in Bahnhof 09 wird verweigert, wenn der folgende Ausdruck logische „true“ ergibt.

Regel 8:

$$(09aR \vee 09bR) \wedge ((13aGR \wedge 13bR) \vee (13aR \wedge 13bGR)) \wedge (14bGR \vee 15bGR \vee 16bGR)$$

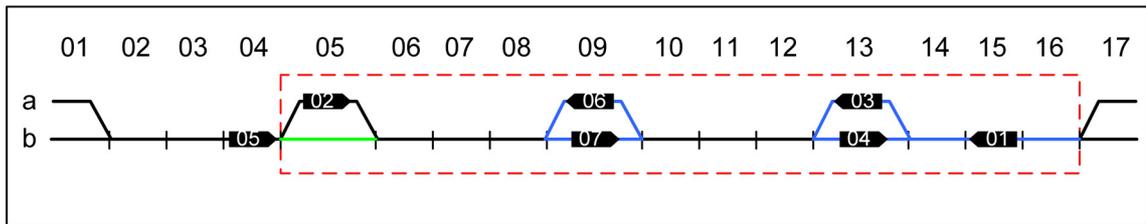


**Abbildung 51: Darstellung der Regel 08 der Methode AAK**

Diese Situation lässt sich beliebig erweitern. Abbildung 52 zeigt dies beispielhaft. Die Züge 06 und 07 belegen den Bahnhof 09. Durch weiteres Einlegen von Zügen zwischen den Blockierungen in den Bahnhöfen 13 und 05 kann der „Deadlock gefährdete Bereich“ beliebig vergrößert werden. Im extremen Fall auf die gesamte Strecke. Somit ist nun klar, dass für jede Möglichkeit eines Deadlocks eine Regel aufgestellt werden muss.

Durch die bereits aufgestellten Regeln werden Deadlocks verhindert. Jedoch entstehen bedingt aus dieser Tatsache Deadlocks, die zuvor eher unwahrscheinlich waren. Es müssen weiterhin nicht nur Situationen durch die abhängigen Kanten ausgeschlossen werden, wie in Abbildung 52 dargestellt ist, sondern auch solche, aus denen sie entstehen können. Je größer dann der „Deadlock gefährdete Bereich“ ist, desto mehr Möglichkeiten müssen über die abhängigen Kanten definiert werden. Die Anzahl der auszuschließenden

Möglichkeiten steigt mit Größerwerden des „Deadlock gefährdeten Bereiches“ exponentiell an.



**Abbildung 52: Beispiel für weitere Regeln nach AAK**

### 7.1.4 Bewertung der Methode

Die Erkenntnis der vorangegangenen Kapitel bringt die Methode mit Ausschlussverfahren durch abhängige Kanten nicht zum Scheitern, beschränkt sie jedoch in ihrer Anwendbarkeit. Werden alle Möglichkeiten in Betracht gezogen, um Deadlocks zu vermeiden, ist die Wahrscheinlichkeit, dass ein Deadlock auftritt, bei 0%. Der große Aufwand die Methode vollständig auf die betrachteten Szenarien anzuwenden, macht eine weitere Betrachtung unmöglich. Die weitere Betrachtung bzw. das weitere Aufstellen von Regeln würden den Rahmen dieser Arbeit überschreiten.

Die Vorteile sind jedoch, dass alle möglichen Fahrmanöver, wie „Überholen“, „Folgefahrt“, „Kreuzen“ usw., weiterhin bestehen bleiben. Es konnte auch gezeigt werden, dass diese Art zur Vermeidung von Deadlocks ihre Wirkung zeigt. Mit der Anwendung der Regeln traten die Deadlocks mit größerer Komplexität wesentlich später in der Simulation auf und das Auftreten von Deadlocks mit geringer Komplexität wurde vermeiden.

Die Anwendung auf Fahrpläne, die realitätsnäher gestört werden, verspricht somit ein positives Ergebnis, da die stochastischen Einflüsse im Laufe der Simulation aus dem System herausgefahren werden können. Der hier zugrunde gelegte Fahrplan ist jedoch so stark gestört, dass er die Komplexität der Deadlocks, bezogen auf die Zeit fördert. Dies ist so, weil die Störungen über den gesamten Simulationszeitraum bei jedem Einbruch eines Zuges in das System Einfluss üben und sich somit das Verspätungsniveau aufschaukelt.

Eine Anwendung der Regeln 1, 4, 5, 6, 7 und 8 ist also bei Betrachtungen, die zeitlich begrenzte Störungen betrachten bzw. denen eine geringere, realitätsnahe Störungsverteilung zugrunde liegt, durchaus sinnvoll. Die Implementierung ist jedoch unhandlich und aufwendig in RailSys vorzunehmen. Es ist jedoch zurzeit die einzige Möglichkeit Deadlocks auf eingleisigen Strecken, bei geringer Kapazität der Bahnhofsinfrastruktur in RailSys, zu vermeiden.

## **7.2 Ergänzung der Methode DRR**

Wie in Kapitel 6.2.3 gezeigt ist, vermeidet die Methode DRR zuverlässig die dort betrachteten Deadlocks. Bei Deadlocks, die aus einer Folgefahrt entstehen, kommt es jedoch dazu, dass die Leistungsfähigkeit der Strecke eingeschränkt wird. Auch in Situationen, aus denen keine Deadlock-Gefahr hervorgeht, kommt es zu diesem Effekt. Die Ursache dafür liegt in den definierten Regeln. Regel 1 und Regel 2 schließen aus, dass Züge sich auf einem eingleisigen Streckenabschnitt folgen können, wenn dieser Abschnitt aus mehreren Blöcken besteht. D.h. ein Zug bleibt immer solange im Bahnhof stehen, bis der ihm vorausfahrende Zug mindestens die Streckenblöcke geräumt hat.

Um diesen Misstand zu beheben, soll in diesem Kapitel versucht werden, ergänzende Regeln aufzustellen bzw. die bestehenden Regeln zu modifizieren. An dieser Stelle muss jedoch darauf hingewiesen werden, dass der Erfolg der Methode durchaus mit dieser Einschränkung zu tun haben kann.

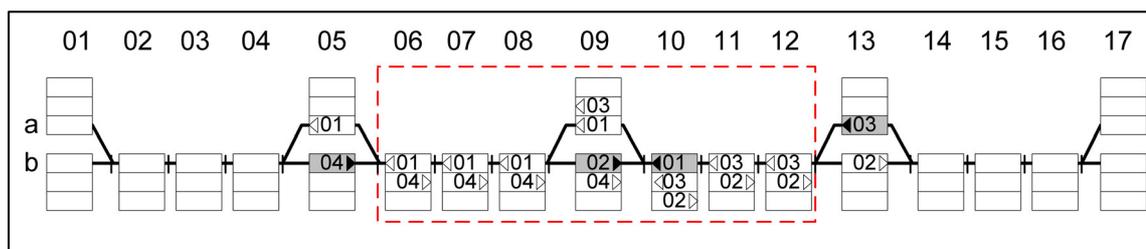
### **7.2.1 Definition der Regeln**

Um weitere Regel zu definieren, muss betrachtet werden, unter welchen Umständen ein Zug einem anderen Zug auf einer eingleisigen Strecke folgen kann. Wie die Untersuchungen in dieser Arbeit gezeigt haben, kann das Fahrmanöver „Folgefahrt“ bei geringer Kapazität der Bahnhofsinfrastruktur zu einem Deadlock führen. Aus diesem Grund dürfen nicht beliebig viele Züge auf einer eingleisigen Strecke folgen. Abbildung 42 stellt eine Ausgangssituation für diese Betrachtung dar. In dem betrachteten Beispiel ist es unter Anwendung der gültigen Regeln so, dass weder Zug 03 noch Zug 04 in den eingleisigen Streckenabschnitt einfahren dürfen, solange nicht einer der beiden Züge im

---

Bahnhof 09 ausgefahren ist. Es ist jedoch theoretische möglich, dass einer der beiden Züge, die „Folgen“ möchten (Zug 03 und 04), dieses Fahrmanöver durchführt, ohne dass eine Deadlock-Situation entsteht.

In diesem Beispiel wird angenommen, das Zug 03 als erster die Möglichkeit erhält, seinem vorausfahrenden Zug (Zug 01) zu folgen. Diese Folgefahrt darf nur stattfinden, wenn gewährleistet ist, dass Zug 01 den Bahnhof räumen kann oder ein weiteres Bahnhofsgleis frei ist. Um die Situation zu verdeutlichen, wird sie zu einer früheren Zeit in Abbildung 53 noch einmal dargestellt. Zug 01 befindet sich hier noch vor der Einfahrt in Bahnhof 09. Zug 03 kann nicht folgen, weil noch nicht alle benötigten Blöcke reserviert sind. Da der Fahrweg für Zug 01 aus Bahnhof 09 in Bahnhof 05 bereits reserviert ist, könnte Zug 03 ihm bereits jetzt folgen.



**Abbildung 53: Analyse der Leistungsfähigkeitsbeeinflussung durch DRR**

Dieser Aussage steht die Regel 1 im Wege. Sie sagt aus, dass ein Zug erst in einen Block einfahren darf, wenn der Weg in den Block vollständig reserviert ist. Der Zielblock wird durch Regel 2 und Regel 4 beeinflusst. Regel 1 muss also angepasst werden, dass eine Reservierung des gesamten Fahrweges nicht immer zwingend notwendig ist. Hierfür werden zwei Begriffe eingeführt:

- Primäre Fahrwegsreservierung
- Sekundäre Fahrwegsreservierung

Diese Definition ist eine Unterteilung in der Art, wie ein Fahrweg von einem Zug reserviert wird. Die primäre Fahrwegsreservierung ist die bis jetzt betrachtete Reservierung. Bei ihr muss ein Zug an der ersten Position in der Liste der Reservierungen für einen Block stehen. Die sekundäre Fahrwegsreservierung hingegen bedingt nicht, dass der Zug in der ersten Position der Reservierungsliste

steht. Es reicht hier, dass der vorausfahrende Zug den Weg reserviert hat und der aktuelle Zug dem voraus fahrenden Zug in der Reservierungsliste direkt folgt.

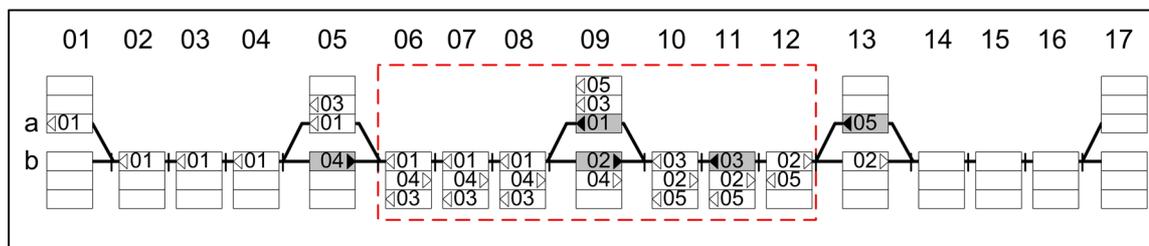
Durch diese Unterscheidung der Fahrwegsreservierung können die Regel neu definiert werden:

1. Bevor ein Zug A in einen Block einfahren darf, muss der Weg in diesen Block für ihn primär oder sekundär reserviert sein, wobei primär eine direkte Reservierung bedeutet.
  2. Wenn ein Fahrweg reserviert wird, der in einen Block mit Zweirichtungsbetrieb führt, der kein Stationsblock ist, dann muss der Weg, um diesen Block zu verlassen, ebenfalls primär oder sekundär reserviert werden.
  3. Ist ein zu reservierender Block durch einen anderen Zug reserviert oder belegt, wird die gewünschte Reservierung einer Reservierungsliste unten angestellt.
  4. Folgt in der Reservierungsliste ein Zug A auf einen Zug B, wobei beide Züge die gleiche Fahrtrichtung haben und Zug B eine primäre oder sekundäre Reservierung für diesen Block besitzt, so ist dies eine sekundäre Reservierung.
  5. Wenn ein Zug A einen Block primär oder sekundär reservieren möchte, der von einem Zug B primär oder sekundär reserviert ist, muss Zug B eine weitere Reservierung (primär oder sekundär) durchführen, um den zu reservierenden Block für den nachfolgenden Zug A zu räumen.
  6. Ist ein zu reservierender Stationsblock durch einen in Gegenrichtung fahrenden Zug reserviert oder belegt, der wiederum in den Block des anfragenden Zuges einfahren möchte, scheitert die Anfrage auf Reservierung.
  7. Eine Anfrage auf Reservierung wird nur wirksam, wenn alle folgenden Reservierungen, die durch die Regeln 2 und 5 nötig werden, reserviert werden können.
-

8. Wenn ein Zug in einem Stationsblock endet, in dem er in der Reservierungsliste ganz oben steht, kann er in den Reservierungslisten der vorliegenden Blöcke auch an die erste Stelle gesetzt werden.

### 7.2.2 Test der ergänzten Methode DRR

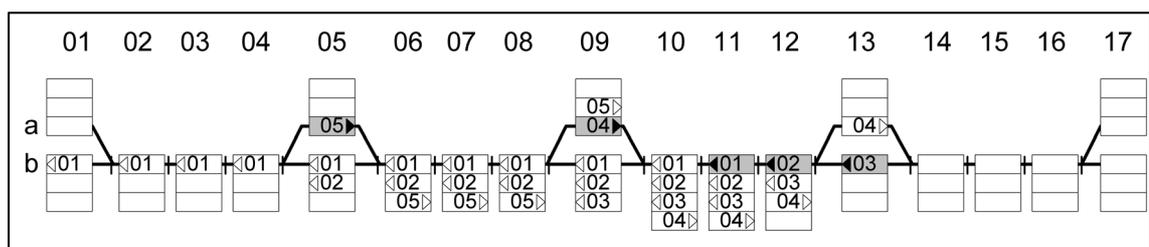
Durch die neu definierten Regeln ist es möglich, dass Zug 03 Zug 01 auf der eingleisigen Strecke dicht folgen kann. Abbildung 54 stellt die Situation dar, nachdem Zug 01 den Bahnhof erreicht hat und Zug 03 auf die eingleisige Strecke eingefahren ist. Weiterhin wird ein neuer Zug (Zug 05) in das System ab Bahnhof 13 eingeführt. Nachdem Zug 03 die eingleisige Strecke verlassen wird, ist diese für Zug 02 reserviert, der in den Block 13b einfahren möchte.



**Abbildung 54: Einfachen Folgefahrt durch Zug 03; Ergänzung DRR**

Zug 01 wird währenddessen Block 09a verlassen und nach Block 05a fahren. Regel 5 bedingt, dass Zug 01 bis Block 01a und Zug 03 bis Block 05a reserviert. Zug 03 kann jedoch Zug 01 nicht folgen, weil er in den Blöcken 08b, 07b und 06b weder primär noch sekundär reserviert hat. Das Beispiel zeigt, dass eine Folgefahrt möglich ist, ohne einen Deadlock zu produzieren.

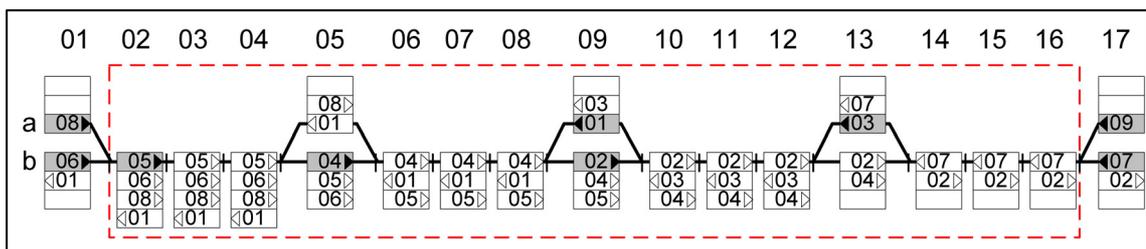
Jetzt muss überprüft werden, ob die modifizierten Regeln auch ausreichend sind, wenn ein weiterer Zug folgen möchte. Abbildung 55 zeigt ein Beispiel, in dem die Züge 01, 02 und 03 aufeinander folgen möchten.



**Abbildung 55: Mehrfachen Folgefahrt durch Zug 03; Ergänzung DRR**

Je mehr Züge aufeinander folgen möchten, desto weiter im Voraus muss z.B. Zug 01 seinen Fahrweg reservieren. In diesem Beispiel bis in Block 01b. Gleiches gilt für die anderen Züge. Ist eine Parallelfahrt der folgenden Züge in einem Bahnhof möglich und dies vom Disponenten gewünscht, verlängert sich der zu reservierende Abschnitt natürlich nicht in dem Ausmaß, wie in Abbildung 55 gezeigt ist. Regel 5 tritt dann nicht in Kraft.

In einem letzten Beispiel soll gezeigt werden, dass die modifizierte Methode auch komplexe Deadlocks verhindern kann. In Abbildung 56 wird diese Situation dargestellt. Es befinden sich zurzeit der Betrachtung neun Züge im System. Zug 05 bekommt aufgrund der neudefinierten Regeln die Möglichkeit Zug 04 zu folgen, obwohl dieser den Bahnhof noch nicht verlassen hat. Die daraus resultierenden Reservierungen übertragen sich bis auf Zug 02, der bis in den Block 13b reserviert hat. Damit Zug 06 ebenfalls folgen kann, muss für Zug 02 jedoch gewährleistet sein, dass er in Block 17b einfahren kann. Zug 09 hat hier noch keine Anfrage auf Reservierung gestellt und belegt einfach den Block 17a.



**Abbildung 56: Vermeidung einer komplexen Deadlock-Situation durch DRR**

Trotz der komplexen Situation werden die Zugbewegungen so abgewickelt, dass kein Deadlock entsteht.

Das Beispiel ist weiterhin eine Rechtfertigung für die Regel 8. Zug 08 möchte Block 05a reservieren, den Zug 01 bereits primär reserviert hat. Durch die Anfrage auf Reservierung durch Zug 08 muss Zug 01, aufgrund der Regel 5 eine Fahrwegsreservierung vornehmen, um den Block 05a zu verlassen. Dadurch steht er in der Reservierungsliste für die Blöcke 04b, 03b und 02b unterhalb von Zug 08 und kann somit erst diese Blöcke befahren, wenn Zug 08 in den Bahnhofsblock 05b eingefahren ist, den er selbst belegt. Dies entspricht einer Deadlock-Situation. Durch Regel 8 wird dies jedoch verhindert, da durch eine

primäre Reservierung im Block 01b die Reservierung für die Streckenblöcke bevorzugt gegenüber Zug 08 behandelt werden kann.

### **7.2.3 Bewertung der Methode**

Die Erweiterung der DRR Methode führt dazu, dass nun alle Fahrmanöver theoretisch stattfinden können. Es müssen jedoch weiterhin bestimmte Randbedingungen eingehalten werden, um z.B. Folgefahrten stattfinden zu lassen. Diese Einschränkungen sind jedoch die Voraussetzung, dass Deadlocks erfolgreich verhindert werden können. Der resultierende Zugbewegungsablauf, der jeweils betrachteten Züge, ist logisch und durchaus mit den Entscheidungen eines Disponenten zu vergleichen.

Weiterhin ist durch die Anwendung der erweiterten DRR Methode eine dichte Zugfolge nun möglich, die eine hohe Auslastung der Infrastruktur impliziert.

Auch anhand von komplexen Beispielen wurde gezeigt, dass beide Eigenschaften (Deadlock-Vermeidung und dichte Zugfolge) in Kombination zu realisieren sind.

Bei einer Implementierung in RailSys wäre es zudem möglich, die Methode unabhängig vom Disponentenmodus zu betrachten und somit die Funktion der Fahrplanstabilisierung und der Deadlock-Vermeidung zu trennen.

---

---

## 8 Diskussion zur Deadlock-Vermeidung

In den Kapiteln 5, 6 und 7 wurden unterschiedliche Möglichkeiten zur Vermeidung von Deadlocks vorgestellt. Die Verfahren gliedern sich somit in bereits in RailSys implementierte Methoden, Methoden und Verfahren aus der Literatur und dem Entwurf eigener Methoden. In diesem Kapitel soll versucht werden eine zusammenfassende Aussage zu treffen, um die jeweiligen Möglichkeiten einschätzen zu können. Die Methoden und Verfahren werden anhand der folgenden Kriterien betrachtet und beurteilt:

- Wirksamkeit der Deadlock-Vermeidung
- Anwendbarkeit, bezogen auf die Implementierung
- Anwendbarkeit, bezogen auf den Wirkungsbereich
- Realitätsnähe der Fahrtverläufe
- Kapazitätsbeeinflussung

In RailSys besteht nur bedingt die Möglichkeit durch implementierte Methoden Deadlocks zu vermeiden. Eine der Methoden, die nur für diesen Zweck in RailSys implementiert ist, beschäftigt sich mit der eingleisigen Disposition. Hierdurch wird das Entstehen von trivialen Deadlocks vermieden. Die Wirksamkeit der Methode ist unumstritten. Jedoch versagt auch diese Methode unter bestimmten Bedingungen. Dieses kann jedoch mit geringem Aufwand, nach bekannt werden der Orte an denen die Deadlocks entstehen, durch manuelle Nachbearbeitung gelöst werden. Ansonsten ist die Anwendung mit geringem Aufwand verbunden, da die Methode lediglich zugeschaltet werden muss. Die Größe des Netzes spielt hierbei keine Rolle, wenn man die Rechenzeit bei erweitertem Umfang vernachlässigt. Weiterhin entstehen durch die Disposition auf eingleisigen Strecken keine Kapazitätseinbußen und die Fahrtverläufe sind logische Folgen und somit eindeutig nachvollziehbar.

Durch den Disponenten in RailSys existieren sonst keine weiteren Methoden, die einen nennenswerten Einfluss auf die Deadlock-Vermeidung haben. Das liegt daran, dass die Methoden des Disponenten nicht zur Deadlock-Vermeidung in

---

---

RailSys implementiert worden sind. Sie sollen eher das Störungsmanagement eines Disponenten nachempfinden. Theoretisch ist der Disponent in der Lage die auftretenden Konflikte zu erkennen. Ihm fehlt jedoch das Werkzeug, um auf eine bevorstehende Deadlock-Situation zu reagieren. Somit steht fest, dass bis auf die Disposition auf eingleisigen Strecken, keine der anderen Dispositionsmethoden sich zur Deadlock-Vermeidung eignet.

Ein anderes Bild stellt sich bei den weiteren Möglichkeiten in RailSys dar. Der Signalvorschaubereich und die Fahrwegreservierung sind zwei Instrumente, die zwar nicht direkt der Deadlock-Vermeidung dienen, jedoch eine gewisse Wirkung in bestimmten Situationen haben. Die Fahrwegsreservierung hat jedoch nur eine Wirkung auf die Deadlock-Vermeidung, wenn sie zusammen mit dem Signalvorschaubereich eingesetzt wird. Ein positives Ergebnis wird dann jedoch nur erzielt, wenn die Gleise in den Bahnhöfen im Richtungsbetrieb betrieben werden oder die Bahnhofsinfrastruktur ausreichend groß ist. Die Fahrwegreservierung ist sehr einfach in den Parametern der Modellzüge einzugeben. Sollen verschiedene Bereiche eingegeben werden, wird das Anlegen weiterer Modellzüge nötig. Der Signalvorschaubereich ist dagegen schon aufwendiger einzugeben. Hier muss an jedem Signal, ab dem eine Signalvorschau erfolgen soll, eine Definition stattfinden. Dieses Vorgehen kann bei größeren Netzen schnell zu einem unangemessenen Aufwand führen. Ein weiterer Nachteil ist, dass ein dichtes Folgen der Züge nach der Definition der Methode zum Teil nicht mehr möglich ist, da die Deadlock-Vermeidung auf diesem Prinzip beruht. Hierdurch kann also die Leistungsfähigkeit der Strecke eingeschränkt werden. Dies bedeutet wiederum, dass die Fahrmanöver nicht richtig abgebildet werden können.

Die Methode MCA ist in der Lage zu 100% alle Deadlocks zu vermeiden. Es ist jedoch schwierig sie in die Programmstruktur zu implementieren, da das Verfahren eine neue Struktur bedingt. Weiterhin ist die Anwendung eingeschränkt, weil für jede Zugbewegung eine Ereigniskette angestoßen werden muss, die theoretisch die Zugbewegung jedes Zuges, der sich aktuell im System befindet, betrachtet. Hieraus lässt sich ein großer Rechenaufwand ableiten. Somit ist die Anwendung auf kleinere Netze beschränkt, in Abhängigkeit von der zur

---

Verfügung stehenden Rechenleistung und der Anzahl der Züge im System. Eine weitere Einschränkung ergibt sich aus der Tatsache, dass die Fahrwege der betrachteten Züge bekannt sein müssen. Somit fallen spontane Gleisänderungen, die z.B. durch den Disponenten nötig werden, aus. Dadurch ist eine realitätsnahe Umsetzung der Fahrtwege nicht mehr gegeben. Auf die Kapazität hat die Methode keinen direkten Einfluss. Jedoch kann, aufgrund der nicht vorhandenen Flexibilität bei der Routenwahl, die Kapazität indirekt negativ beeinflusst werden.

Die DRR Methode mit ihren sieben Regeln vermeidet diesen Nachteil dadurch, dass die Routenwahl flexibel bleibt. Somit kann die gesamte Kapazität im Bahnhof ausgenutzt werden. Die Vermeidung der betrachteten Deadlocks geschieht auch zuverlässig. Die Methode muss in RailSys ebenfalls implementiert werden, was sich jedoch nicht so schwierig gestaltet, wie beim MCA Verfahren. Der Grundsatz von DRR basiert auf Blockreservierungen, welches bereits ein Grundprinzip in RailSys ist bzw. durch andere Methoden, wie die Fahrwegreservierung, umgesetzt wird. Da die Methode in die Programmstruktur implementiert werden würde, ist sie somit auf beliebig große Netze anwendbar. Der zusätzliche Rechenaufwand ist gering, da der zu reservierende Bereich durch einen Zug überschaubar ist. Die Nachteile der Methode DRR liegen, unter Berücksichtigung der sieben Regeln, bei den nicht realitätsnahen Fahrwegsentscheidungen und der eingeschränkten Leistungsfähigkeit auf den eingleisigen Strecken mit mehr als einem Block.

Das Ausschlussverfahren durch abhängige Kanten zielt darauf ab, den Nachteil einer Leistungsfähigkeitseinschränkung zu vermeiden. Weiterhin ist ein Ziel, dass alle in der Realität vorkommenden Fahrmanöver trotz der Methode erhalten bleiben. Diese beiden Forderungen werden auch eingehalten, jedoch unter einem großen Aufwand die Methode in RailSys zu implementieren. Durch die bis jetzt definierten Regeln werden nicht alle Deadlocks vermieden. Somit ist die Anwendbarkeit auf kleinere Netze bzw. Strecken und Fahrpläne mit einem geringen Verspätungsniveau beschränkt. Durch das Definieren weiterer Regel kann jedoch der Anwendungsbereich, bezogen auf das Verspätungsniveau, erweitert werden. Dieses schränkt jedoch den Anwendungsbereich auf die Größe des betrachteten Netzes stark ein.

---

Die Erweiterung der DRR Methode versucht die Vorteile der DRR Methode mit sieben Regeln aufzunehmen und durch die Erweiterung die Nachteile auszuschließen. Somit ist gewährleistet, dass Deadlocks zuverlässig vermieden werden. Dies geschieht unter Berücksichtigung einer generellen Implementierung in die Programmstruktur. Somit ist die Anwendbarkeit, bezogen auf Implementierung, gewährleistet. Die Erweiterung stellt sicher, dass alle Fahrmanöver, die in der Realität vorkommen können, in der Simulation auch abgebildet werden können. Hierdurch ergibt sich keine Leistungsfähigkeitseinschränkung auf den Strecken.

Die nachfolgende Tabelle 5 stellt alle betrachteten Verfahren noch einmal gegenüber, um die Vor- bzw. Nachteil besser vergleichen zu können. Hierbei wird jedes Kriterium nach dem Schulnotensystem bewertet. Das Kriterium der Wirksamkeit bei der Deadlock-Vermeidung ist ein Ausschlusskriterium. D.h. eine Methode kann in allen Kriterien gut sein. Wenn sie jedoch nicht in der Lage ist Deadlocks zu vermeiden, hat die Methode keinen Nutzwert.

Getestete Methoden	Kriterien						Notendurchschnitt
	Wirksamkeit der Dealock-Vermeidung	Anwendbarkeit, Implementierung	Realitätsnähe der Wirkungsbereich	Kapazitätsbeeinflussung	Kapazitätsbeeinflussung	Notendurchschnitt	
ingleisige Disposition	2	1	1	1	1	1,2	
Disponent in RailSys	5	1	2	1	1	2	
Signalvorschaubereich	3	3	2	4	4	3,2	
Fahrwegsreservierung	4	2	2	1	3	2,4	
MCA	1	4	3	4	4	3,2	
DRR mit sieben Regeln	2	2	1	4	4	2,6	
Ausschlussverfahren durch abhängige Kanten (AAK)	2	5	2	2	2	2,6	
Erweiterung der DRR Methode	2	2	1	2	2	1,8	

**Tabelle 5: Übersicht der Bewertung der untersuchten Methoden**

Aus der Tabelle 5 werden die einzelnen Unterschiede deutlicher. Die Disponenten-Methoden in RailSys fallen durch das Ausschlusskriterium aus der Betrachtung heraus. Die beste Wirksamkeit in der Deadlock-Vermeidung besitzt das Verfahren durch MCA. Dieses Verfahren ist jedoch in drei anderen Kriterien

mit der Note vier bewertet. Die Wirksamkeit in der Deadlock-Vermeidung ist mitunter das wichtigste Kriterium. Die anderen Kriterien sind jedoch teilweise auch zwingend notwendig. So kommt das Verfahren durch MCA nur auf eine Durchschnittsnote von 3,2.

Jede Methode ist in einem Kriterium gut. Je nachdem auf welche Eigenschaft der Benutzer Wert legt, ist eine andere Methode zu wählen. Es sollte jedoch eine Methode vorgezogen werden, die in allen Kriterien ausreichend gut ist. Dieses trifft auf zwei Methoden zu. In der Gesamtbewertung hat die eingleisige Disposition den besten Wert. Jedoch muss berücksichtigt werden, dass diese Methode nur triviale Deadlocks erkennt und lösen kann. Als zweites ist die erweiterte DRR Methode zu erwähnen. Sie ist in allen Kriterien mindestens gut. Sie muss nur fest in RailSys implementiert werden.

Solange dieses nicht der Fall ist, bietet das Ausschlussverfahren durch abhängige Kanten eine gute Möglichkeit Deadlocks zu vermeiden. Das Problem ist die aufwendige Eingabe in RailSys, die zu diesem Zeitpunkt nur durch den Benutzer stattfinden kann. Es muss beachtet werden, dass die Methode nur in kleinen Netzen bzw. auf kurzen Strecken, aufgrund des großen Aufwandes, angewandt werden kann. Weiterhin kann diese Methode nicht angewandt werden, wenn der Fahrplan mit einem großen Verspätungsniveau belastet ist.

Alle anderen Methoden sind in ihren Möglichkeiten der Deadlock-Vermeidung nicht ausreichend und/oder beeinflussen die Fahrverläufe zu stark, so dass keine realitätsnahe Simulation mehr möglich ist.

---

---

## 9 Zusammenfassung und Ausblick

Die Arbeit beschäftigt sich mit dem Thema der Deadlock-Vermeidung, welches eine wesentliche Rolle in synchronen Simulationsprogrammen für spurgeführten Verkehr spielt. Als Simulationsprogramm wird beispielhaft RailSys der Firma RmCon herangezogen. Mit diesem Programm wird die Untersuchung durchgeführt.

Zunächst werden Szenarien in einem Infrastrukturnetz definiert. Die Szenarien werden so gewählt, dass ein möglichst breites Spektrum bezüglich der Fahrplankonstruktion und der Infrastruktur betrachtet werden kann. Die gesamte Arbeit orientiert sich an diesen Szenarien, was eine sorgfältige Betrachtung nötig macht. Da der Fahrplan konfliktfrei konstruiert ist, wird er gezielt durch Verspätungen gestört. Diese Störungen liefern eine weitere Grundlage für die Deadlock-Betrachtung.

Aus den gestörten Fahrplänen leiten sich also nun die einzelnen Deadlock-Situationen ab. Diese werden analysiert, um das Entstehen einer Deadlock-Situation betrachten zu können. Wie oben gezeigt wird, ergeben sich grundsätzlich drei Arten von Deadlocks. Wobei die erste Art triviale Deadlocks sind, die ein Gegenüberstehen zweier Züge auf einer eingleisigen Strecke beschreiben. Weiterhin entstehen Deadlocks aus dem Fahrmanöver „Überholung“ bzw. einer Parallelfahrt in einem Bahnhof. Hierbei wird die Strecke in der einen Richtung durch Züge blockiert, die im Bahnhof alle Gleise in der gleichen Richtung belegen. In der Gegenrichtung blockiert ein Zug die Strecke auf einem eingleisigen Streckenabschnitt, der an den blockierten Bahnhof grenzt. Die dritte Art von Deadlocks entsteht durch das Fahrmanöver „Folgefahrt“ auf zwei eingleisigen Streckenabschnitten, die an einem gemeinsamen Bahnhof angrenzen. Der Deadlock entsteht hier, weil der Bahnhof nicht genügend Kapazität hat, um alle Züge aufzunehmen und somit einen flüssigen Betriebsablauf zu gewährleisten. Es werden auch Deadlocks betrachtet, die auf zweigleisigen Strecken und in komplexen Bahnhöfen entstehen können.

---

Der Hauptgrund ist also ein Mangel an ausreichender Kapazität in Bahnhöfen. Dieses Defizit muss durch Methoden und Verfahren ausgeglichen werden, die ein Zufahren der Bahnhöfe verhindern und ein kontrolliertes Zurückhalten der Züge gewährleisten.

Um dies sicherzustellen, werden in der Arbeit Verfahren vorgestellt, die diese Eigenschaft beinhalten oder beinhalten könnten. Das Spektrum der betrachteten Methoden reicht von

- in RailSys implementierten Methoden über
- Vorschläge der Literatur bis hin zu
- eigenentwickelten Verfahren.

Alle Methoden haben unterschiedliche Eigenschaften und somit spezifische Vor- und Nachteile. Die in RailSys implementierten Methoden des Disponenten eignen sich generell nicht zur Vermeidung von Deadlocks. Ihr Vorhandensein resultiert aus dem Störungsmanagement, um den Fahrplan stabil zu halten. Es wird weiterhin gezeigt, dass weitere Methoden, die durchaus einen Einfluss auf das Deadlock-Verhalten haben, die Kapazität der Bahnhöfe bzw. die Leistungsfähigkeit der Strecke einschränken, auch wenn keine Deadlock-Gefahr besteht. Die eigen- bzw. weiterentwickelten Methoden versuchen die Vorteile der vorhandenen Methoden zu übernehmen und die Nachteile durch Ergänzungen auszugleichen. Durch eine Methode (Ausschlussverfahren durch abhängige Kanten) wird anhand der Szenarien gezeigt, dass es möglich ist, mit den zur Verfügung stehenden Mitteln in RailSys Deadlocks erfolgreich zu vermeiden. Jedoch geschieht dies unter einem großen Aufwand, der in großen Simulationsnetzen nicht aufzubringen ist.

Eine genaue Untersuchung bestimmter Verfahren konnte nur theoretisch erfolgen, da sie nicht direkt in RailSys implementiert werden können. Ein weiterer Schritt in der Deadlock-Vermeidung ist also nun die verbesserten Methoden direkt in der Simulation zu testen und gegebenenfalls weitere Regeln zu definieren. Das Thema Deadlock-Vermeidung ist mit dieser Arbeit keineswegs abgearbeitet. Es müssen weiterhin Überlegungen stattfinden, die die Entscheidungen eines Disponenten in der Wirklichkeit nachempfinden können.

---

Dabei sollte stets berücksichtigt werden, dass alle Fahrmanöver erhalten bleiben müssen, um die Abbildung der Realität zu gewährleisten.

Ein Ansatz, der in dieser Arbeit nicht direkt betrachtet wird, beschäftigt sich mit der Analyse der zur Verfügung stehenden Kapazität der betrachteten Strecke. Zugbewegungen können hier nur durchgeführt werden, wenn die Auslastung der Infrastruktur dies zulässt.

Dieser und weitere Ansätze sollten in Zukunft betrachtet werden. Zudem sind die bereits vorhandenen Methoden weiterhin zu betrachten und kontinuierlich zu verbessern. Die Deadlock-Vermeidung ist ein wichtiger Bestandteil der synchronen Simulation, um alle möglichen Szenarien untersuchen zu können und erlangt daraus seine Daseinsberechtigung.

Natürlich kann ein solches Verfahren nicht nur in Simulationen angewandt werden. Auch der Einsatz in der Realität, als Kontroll- oder Hilfsinstrument für die Disposition ist denkbar. Weiterhin kann ein vollautomatisierter Zugbetrieb nur mit einem solchen Verfahren ermöglicht werden, der sich direkt aus der heutigen synchronen Simulation ableiten lässt.

---

---

## Literatur und Quellenverzeichnis

- [BRU83] Bruns, M.: Zum Entwurf von Steuerungen für komplexe Transportsysteme, Dissertation, RWTH Aachen 1983
- [BEC03] Beckmann, A.; Einführung in die Logik; 2. Auflage; Walter de Gruyter, Berlin 2003
- [DLR06] DLR: Institutsflyer – IFS, Technologien aus Luft- und Raumfahrt für Straße und Schiene, Braunschweig 2006
- [FER04] Ferchland, Ch.; Warninghoff, C.-R.: „Nutzung von Simulationen zur Unterstützung der betrieblichen Infrastrukturplanung“, in: ETR, Heft 7/8, Hamburg 2004, S. 490 – 489
- [LAD01] Lademann, F.: Bemessung von Begegnungsabschnitten auf eingleisigen S-Bahn-Strecken, Dissertation, <http://elib.tu-darmstadt.de/diss/000175>, TU Darmstadt 2001
- [NAU04] Naumann, P.; Pachl, J.: Leit- und Sicherungstechnik im Bahnbetrieb, Band 2, 2. Auflage, Tetzlaff, Hamburg 2004
- [PAC93] Pachl, J.: Steuerlogik für Zuglenkanlagen zum Einsatz unter stochastischen Betriebsbedingungen, Dissertation, TU Braunschweig 1993
- [PAC97] Pachl, J.: „Zum Deadlockproblem bei der automatischen Betriebssteuerung“, in: Signal + Draht, Heft 1+2, Hamburg 1997, S. 22 – 25
- [PAC01] Pachl, J.: „Übertragbarkeit US-amerikanischer Betriebsverfahren auf europäische Verhältnisse“, in: ETR, Heft 7/8, Hamburg 2001, S. 452 – 462
- [PAC04] Pachl, J.: Systemtechnik des Schienenverkehrs, 4. Auflage, Teubner, Wiesbaden 2004
- [PAC07] Pachl, J.: Avoiding Deadlocks in Synchronous Railway Simulations, <http://www.digibib.tu-bs.de/?docid=00020794>, Hannover 2007
-

- [RMC05] RmCon: RailSys 4 Handbuch, Version 1.0, Hannover 2005
  - [RMC07] RmCon: [http://www.rmcon.de/uber\\_railsys.html](http://www.rmcon.de/uber_railsys.html), Hannover 2007
-

## **Eidesstattliche Erklärung**

Hiermit erkläre ich an Eides Statt, dass ich die vorliegende Arbeit selbstständig und ohne unerlaubte Hilfe angefertigt, andere als die angegebenen Quellen nicht benutzt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

15.08.2007, Salzgitter

---

## Anhang

### Anhang A: Netzansicht in RailSys

