

RWTH Aachen

Lehrstuhl für Operations Research

— PROF. DR. M. LÜBBECKE —

Masterarbeit im Fach Mathematik

Sommersemester 2012

Betreuer: Prof. Dr. Marco Lübbecke

Zweitgutachter: Prof. Dr. Ir. Arie M.C.A. Koster

Tuğba Güçlü

**Ein Spaltengenerierungsansatz für die Zuordnung von
Güterzügen**

September 2012

Kurzfassung

In dieser Masterarbeit betrachten wir mit Hilfe der ganzzahligen Programmierung die Planung der Zugzusammenstellungen an Rangierbahnhöfen mit Ablaufbergen. Das Problem beinhaltet das Zusammenstellen von abfahrenden Zügen aus ankommenden Zügen unter Kapazitäts- und Schedulingbedingungen. Um die Kapazität des Rangierbahnhofs zu erhöhen, wird eine Erweiterung betrachtet, bei der einzelne, früh ankommende Wagons verschiedener Züge temporär auf einer speziellen Teilmenge von Gleisen gelagert werden können.

Das Problem wurde zuvor in den Papern [1] und [2] als direktes ganzzahliges Programm modelliert. Doch dieser Ansatz lieferte keine unteren Schranken mit ausreichender Qualität, um Optimalität zu beweisen. Dort wurde das Problem als ein spezielles Listen-Färbungsproblem in Intervallgraphen modelliert und die Komplexität verschiedener Problemvarianten untersucht. Ferner wurden sowohl Heuristiken als auch ein ganzzahliges mathematisches Programm für das Problem entworfen. Einleitend fassen wir in dieser Arbeit die wesentlichen Ergebnisse dieser Paper zusammen.

In [3] wurde zusätzlich ein neues, erweitertes ganzzahliges Modell formuliert und ein Spaltengenerierungsansatz entworfen, um das Problem für Instanzen von industrieller Größe zu lösen. Als Fallstudie wurde eine Instanz des Rangierbahnhofs Hallsberg in Schweden bewertet. Wir bewerten das Verfahren und vergleichen die Ergebnisse mit den früheren Ansätzen. Der neuen Methode gelingt es, optimale Lösungen in allen 192 getesteten Probleminstanzen zu finden. Auch diesen Ansatz führen wir anfänglich ein. Als Erweiterungen dieser Formulierung stellen wir exakte Techniken, wie Early Branching und Early Termination, zur Beschleunigung des Lösungsverfahrens dar. Auch versuchen wir mit Hilfe der Smoothing Methode eine Stabilisierung des Spaltengenerierungsansatzes zu erreichen.

Schließlich entwerfen wir ein weiteres kompaktes Modell, welches sich als die sogenannte originale Formulierung der erweiterten Formulierung ergibt, wenn wir auf diese die Dantzig-Wolfe Dekomposition anwenden. Auch diesen Ansatz testen wir auf den 192 Probleminstanzen und vergleichen die Ergebnisse mit den anfänglich eingeführten Ansätzen.

Vorbemerkung

Elementare Begriffe aus den Bereichen der Linearen Programmierung, der ganzzahligen und kombinatorischen Optimierung, sowie der Komplexitätstheorie werden als bekannt vorausgesetzt. Nur um Notationen zu erklären und einzuleiten, werden an notwendigen Stellen Bezeichnungen eingeführt.

Als generelle Referenzen zur Linearen Programmierung und ganzzahligen und kombinatorischen Optimierung seien die Bücher von Chvátal [4] und von Nemhauser und Wolsey [5] genannt. Weitere Werke sind H.J. Zimmermann [6] und Domschke und Drexl [7].

Alle wichtigen Aspekte der Komplexitätstheorie sind in Garey und Johnson [8] zu finden.

Inhaltsverzeichnis

1	Einleitung	7
1.1	Problemdefinition	9
1.2	Sequenzen und zulässige Lösungen	12
2	Zusammenhang zu Intervall-Färbungsproblemen	15
2.1	Eine Cliquenbasierte Heuristik	17
2.2	Eine Verbesserungsheuristik	19
3	Ein binäres Programm für das Rangierproblem	21
3.1	Einführung in die diskrete Optimierung	21
3.2	Die direkte Formulierung	23
4	Ein Spaltengenerierungsansatz für das Rangierproblem	25
4.1	Erweiterte Formulierung	26
4.2	Pricing	29
4.3	Branching	32
4.4	Ergebnisse	33
4.5	Duale Interpretation	36
4.6	Symmetriefreie Formulierung	50
5	Eine arc-indexed Formulierung für das Rangierproblem	55
5.1	Dantzig-Wolfe Dekomposition	55
5.2	Die originale Formulierung	57
5.3	Integralitätseigenschaft des Subproblems	60
5.4	Ergebnisse	62
6	Fazit und Ausblick	65
7	Anhang	67
7.1	Ergebnisse der Heuristik++, (DF) und (EF)	67
7.2	Ergebnisse der (EF) unter Anwendung der Beschleunigungstechniken	74
7.3	Ergebnisse der (OF) im Vergleich zur (EF)	93
	Literaturverzeichnis	99

Versicherung an Eides statt

101

1 Einleitung

Es gibt zwei grundlegende Betriebsarten im Güterverkehr der Eisenbahn:

- Transport einzelner Güterwagen in "gemischten" Zügen.
- Transport der Güterwagen in Ganzzügen.

Im Ganzzugverkehr gehören alle Güterwagen eines Zuges dem gleichen Kunden und haben ein gemeinsames Ziel. Im Einzelwagenverkehr kann ein Güterzug aus Wagons verschiedener Kunden und unterschiedlicher Zielorte zusammengesetzt sein. Um jeden Wagon an seinen Zielort transportieren zu können, werden die Inbound-Züge in einzelne Wagons zerlegt und zu neuen Outbound-Zügen zusammengesetzt. Die Wagen, die ein gemeinsames Zwischenziel teilen, bilden dann die Outbound-Züge.

Dieser Prozess geschieht in Rangierbahnhöfen, die einen Engpass im Einzelwagenverkehr darstellen. Ein Problem innerhalb dieses Prozesses ist die Zuordnung von Rangiergleisen zu Güterzügen, so dass jeder abfahrende Zug auf einem separaten Gleis zusammengestellt werden und pünktlich abfahren kann. Wenn ein Güterwagen den nächsten Zug entlang der Strecke verpasst, kann die entstandene Verzögerung bis zu mehreren Tagen betragen. Um den Einzelwagenverkehr wettbewerbsfähiger zu machen, ist es wünschenswert die Wagen so schnell wie möglich, unter Beibehaltung aller Verbindungen, über das Netzwerk zu transportieren.

Von besonderem Interesse sind die **Rangierbahnhöfe mit Ablaufberg** (engl. hump yard).

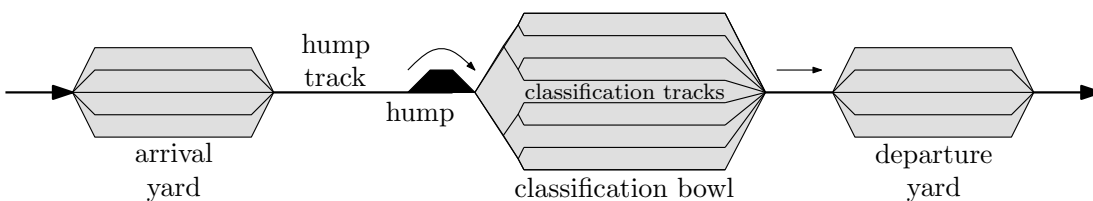


Abbildung 1.1: Eine schematische Darstellung eines typischen Rangierbahnhofs mit Ablaufberg.

Typischerweise besteht ein hump yard aus folgenden Gleisgruppen:

- **Einfahrgruppe:** Anordnung von Gleisen, die für die Lagerung der Güterwagen der Inbound-Züge benutzt werden. Hier werden alle Wagons der Inbound-Züge entkuppelt und auf einem einzelnen Gleis gelagert (vgl. in Abb. (1.1) arrival yard).
- **Richtungsgruppe:** Besteht aus den Rangiergleisen und ist die größte Gleisgruppe des Rangierhofs. Der Ablaufberg ist mit den Rangiergleisen durch eine Menge von Weichen verbunden, so dass ein Güterwagen bzw. eine Lokomotive jeden Rangiergleis von dem Ablaufberg aus erreichen kann (vgl. in Abb. (1.1) classification bowl).
- **Ausfahrgruppe:** Ist eine Zusammensetzung von Gleisen, um die Outbound-Züge zu lagern (vgl. in Abb. (1.1) departure yard).

In hump yards werden die zu rangierenden Wagen von einer Rangierlokomotive auf den Ablaufberg hinaufgefahren und auf dessen Spitze an den erforderlichen Stellen entkuppelt. Sie rollen dann eigenständig durch ihre eigene Schwerkraft -einzeln oder in kleinen zusammengekuppelten Gruppen- das Gefälle hinab und gelangen so nach ihren Bestimmungsbahnhöfen geordnet auf die Rangiergleise der Richtungsgruppe. Das Problem, welches in den Papern [1] und [2] untersucht wird, beschränkt sich auf die Planung der Richtungsgruppe. Insbesondere wird der häufigste Fall betrachtet, bei dem es zu jedem Zeitpunkt eine Bijektion zwischen den Outbound-Zügen und den Rangiergleisen existiert. Dies bedeutet, dass jedes Gleis nur Güterwagen eines einzigen Outbound-Zuges enthalten darf. Deshalb muss für jeden Zug entschieden werden, auf welchem Gleis dieser zusammengestellt werden soll.

Auf Grund der hohen Verkehrsdichte ist es unmöglich ein ganzes Gleis für jeden Outbound-Zug von Beginn der Ankunft des ersten Wagens bis zum Zeitpunkt seiner Abfahrt zu reservieren. Daher werden einige der Gleise als Pufferzonen verwendet, auf denen Güterwagen der verschiedenen Züge temporär gelagert werden können. Diese werden als **Mixed-Gleise** bezeichnet. Um die gewünschte Zusammensetzung der Wagons auf den Rangiergleisen zu erreichen, werden an bestimmten Zeitpunkten zwei Operationen eingesetzt: die **Pull-Out** und die **Roll-In Operation**. Die Pull-Out Operation ermöglicht, unter der Voraussetzung, dass die Zusammenstellung der jeweiligen Outbound-Zügen für jeden solchen Wagen begonnen hat, jede Teilmenge von Güterwagen von den Mixed-Gleisen auf die Rangiergleise zu bewegen. Diese Operation besteht darin, dass die Wagen eines bestimmten Rangiergleises gekuppelt und mit Hilfe einer Rangierlokomotive über den Ablaufberg auf das Ablaufberggleis geschoben werden. Dort werden die Wagons wieder entkuppelt und sind für die anschließende Roll-In Operation bereit. Während dieser rollen die entkuppelten Güterwagen mit ihrer eigenen Schwerkraft vom Ablaufberg durch die

darauf folgenden Weichen auf die den Zielbahnhöfen entsprechenden, Gleise der Richtungsgruppe hinab. In einem Rangierbahnhof mit Ablaufberg folgt einer Pull-Out Operation immer eine Roll-In Operation. Wenn die Zusammenstellung eines Outbound-Zuges fertig ist, werden dessen Güterwagen gekuppelt und durch eine Rangierlokomotive zu der Ausfahrgruppe gefahren. Eine detailliertere Beschreibung dieser Vorgänge kann in [2] gefunden werden.

1.1 Problemdefinition

Gegeben seien eine Menge von Rangiergleisen O , eine Menge von Perioden P , eine Menge von Güterwagen Q , und schließlich eine Menge von Outbound-Zügen R . Gruppen von Fahrzeugen mit dem gleichen Ziel werden als eine Einheit behandelt. Für jeden Wagen $q \in Q$ wird seine Ankunftszeit $t(q)$, d.h. die Zeit seines ersten Roll-Ins in die Richtungsgruppe, seine Länge $s(q)$ und der zugehörige Outbound-Zug $r(q) \in R$ angegeben. Jeder Zug $r \in R$ hat eine Abfahrtszeit $t(r)$, d.h. einen Zeitpunkt, zu dem es die Richtungsgruppe verlässt. Wir bezeichnen mit $Q(r) \subseteq Q$ die Menge der Güterwagen, die zum Zug r gehören. Die Länge $s(r)$ eines Zuges r ist die Summe der Längen seiner Wagons, $s(r) := \sum_{q \in Q(r)} s(q)$.

Für jeden Rangiergleis $o \in O$ ist die Länge durch $s(o)$ gegeben. Somit kann ein Zug r genau dann auf dem Gleis o zusammengestellt werden, wenn $s(r) \leq s(o)$ gilt. Bezeichne $R(o)$ die Menge der Züge, die auf dem Gleis o zusammengestellt werden können. Zu jedem Zeitpunkt kann ein Rangiergleis nur die Güterwagen eines Outbound-Zuges enthalten und jeder Zug kann genau auf einem Gleis zusammengestellt werden. Ein Zug r ist während der Zeitspanne aktiv, in der das entsprechende Rangiergleis ausschließlich für die Zusammenstellung von r eingesetzt ist.

Die strikte partielle Ordnung \prec wird für die Menge von Outbound-Zügen R definiert, so dass $r \prec r'$ genau dann gilt, wenn Zug $r \in R$ direkt vor Zug $r' \in R$ auf dem gleichen Gleis eingeplant werden kann. Ob $r \prec r'$ gilt oder nicht, hängt unter anderem von den Abfahrtszeiten der Züge r und r' , sowie von technischen Rüstzeiten ab. Die Tatsache, dass zwei Züge nicht auf dem gleichen Gleis zusammengestellt werden können, wird mit $r \parallel r'$ bezeichnet.

Im Allgemeinen gibt es nicht genügend Rangiergleise, so dass jeder Zug von der Ankunft seines frühesten Güterwagens bis zu seiner Abfahrt aktiv ist. Daher existiert eine Menge von Mixed-Gleisen, auf denen die Wagons der verschiedenen Outbound-Zügen temporär gelagert werden können. Zur Vereinfachung werden diese Gleise als ein verkettetes Gleis betrachtet und heißen **Mixing-Gleis**. Das Mixing-Gleis hat

eine vorgegebene Länge s^{mix} . Ein Güterwagen, welcher auf dem Mixing-Gleis ist, wird als **mixed** bezeichnet. Ob ein Wagen auf dem Mixing-Gleis gelagert werden muss oder nicht, hängt allein von der Abfahrtszeit des vorhergehenden Zuges auf dem gleichen Gleis, zu dem dieser Wagen zugeordnet ist, ab. Wenn ein Zug aktiv wird, können die mixed Wagen dieses Zuges von dem Mixing-Gleis durch eine Pull-Out Operation, welche einmal am Anfang jeder Periode durchgeführt wird, auf das Rangiergleis gebracht werden. Für jede Periode $p \in P$ ist seine Startzeit durch $t(p)$ gegeben. Während der Pull-Out Operation, die zum Zeitpunkt $t(p)$ gestartet wird, wird jeder mixed Wagen eines derzeit aktiven Zuges auf das zugewiesene Rangiergleis bewegt. Die restlichen mixed Güterwagen gelangen wieder auf das Mixing-Gleis und bleiben dort bis mindestens die nächste Periode beginnt.

Ziel ist es das Lagern der Wagen auf dem Mixing-Gleis zu vermeiden. Für diesen Ansatz gibt es mehrere Gründe. Denn für jede Periode, während dieser ein Güterwagen auf dem Mixing-Gleis ist, muss dieser Wagen einer Roll-In Operation unterliegen, welche Zeit und Aufwand in Anspruch nimmt und zusätzlich die Weichen und Gleise verschleißt. Als Zielfunktion wird deshalb die Anzahl der zusätzlichen Roll-In Operationen minimiert. Weiterhin muss die begrenzte Kapazität s^{mix} des Mixing-Gleises in jeder Periode eingehalten werden. Da kein Güterwagen das Mixing-Gleis verlassen kann, bis die nächste Pull-Out Operation durchgeführt wird, wächst die Anzahl der Wagen in einer Periode an. Daher ist die Gesamtlänge der mixed Wagen am Ende jeder Periode am größten. Für zwei Züge $r \prec r'$ und eine Periode p bezeichnet $s_p(r, r')$ die Gesamtlänge aller Wagen von r' , die in p mixed sind:

$$s_p(r, r') = \begin{cases} \sum_{q \in Q(r'): t(q) < \min(t(r), t(p+1))} s(q), & \text{if } t(p) < t(r) \quad , \\ 0, & \text{sonst.} \end{cases}$$

Ferner bezeichne $c(r, r')$ die Gesamtanzahl der zusätzlichen Roll-Ins für zwei Züge $r \prec r'$:

$$c(r, r') = \sum_{q \in r': t(q) < t(r)} |q| \cdot k_{qr} \quad ,$$

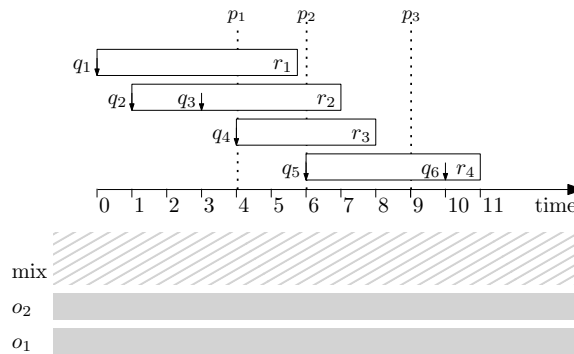
wobei $|q|$ die Zahl der aktuellen Wagen darstellt, die durch die Wagengruppe q repräsentiert wird. k_{qr} ist die Anzahl der Perioden zwischen der Ankunft der Wagengruppen q und der Abfahrt des Zuges r

$$k_{qr} = \#\{p \mid p \in P \wedge t(q) < t(p+1) \wedge t(p) < t(r)\} \quad .$$

Es gibt einige Feinheiten, die beachtet werden müssen. Man betrachte zwei aufeinander folgende Züge r, r' , für die $r \prec r'$ gilt. Jeder Wagen des zweiten Zuges r' , der nach dem Abfahrtszeitpunkt $t(r)$ des ersten Zuges einfährt, kann direkt auf das Rangiergleis geleitet werden, da r' , unmittelbar nachdem der Zug r abgefahren ist, aktiv werden kann. Aber jeder Güterwagen q von r' , der vor der Abfahrt von r ankommt, d.h. für den $t(q) < t(r)$ gilt, muss mixed werden, da das Rangiergleis nur Wagons von einem Outbound-Zug zur gleichen Zeit enthalten darf. Dies bedeutet, dass die partielle Ordnung \prec von den gegebenen Zeiten der Perioden, Zügen und Wagen abhängt. Wenn $r \prec r'$ gilt und der zweite Zug r' einen Güterwagen hat, welcher auf das Mixing-Gleis geschickt werden muss, dann muss eine Periode p mit $t(r) < t(p) < t(r')$ existieren, um den Wagen q pünktlich, bevor der Zug r' abfährt, auf das Gleis o zu platzieren. Zu beachten ist außerdem, dass ein Zug, dessen früheste Güterwagenankunfts- und abfahrtszeit innerhalb der gleichen Periode liegen, keine mixed Wagons besitzen kann. Andernfalls könnte solch ein Güterwagen nicht rechtzeitig auf das zugeordnete Rangiergleis gelangen.

r_i	Wagenankünfte	s	t	$c(r_i, r_j)$		
				r_2	r_3	r_4
r_1	$t(q_1) = 0$	1	5	4	1	0
r_2	$t(q_2) = 1$ $t(q_3) = 3$	1	7		1	1
r_3	$t(q_4) = 4$	1	8			1
r_4	$t(q_5) = 6$ $t(q_6) = 10$	2	11			

(a) Probleminstanz.



(b) Darstellung der Probleminstanz. Nach unten weisende Pfeile zeigen die Wagenankünfte.

Abbildung 1.2: Beispiel Probleminstanz mit zwei Rangiergleisen o_1, o_2 , abfahrenden Zügen r_1, \dots, r_4 , Wagenankünften q_1, \dots, q_6 , und drei Perioden p_1, \dots, p_3 . Die Gesamtanzahl der mixed Wagen, wenn zwei Züge dem gleichen Gleis zugeordnet werden, wird ebenfalls angezeigt. Von den zwei Gleisen kann nur das Gleis o_2 den längsten Zug r_4 unterbringen; alle anderen Züge passen auf beide Gleise.

Eine Beispiel Probleminstanz ist in Abb. 1.2 abgebildet. Hier werden vier Züge r_1, \dots, r_4 zwei Rangiergleisen o_1 und o_2 zugeordnet. Alle Züge passen auf das längste Gleis o_2 , aber nur die ersten drei Züge passen auf das kürzere Gleis o_1 . Die mixing Kapazität wird als unendlich angenommen und die Pull-Outs werden zu den Zeiten $p_1 = 4, p_2 = 6$ und $p_3 = 9$ durchgeführt.

1.2 Sequenzen und zulässige Lösungen

Zulässige Lösungen für das Problem werden als Sequenzen von Zügen definiert, die zu individuellen Gleisen zugeordnet werden können. Eine **Sequenz** g ist eine total geordnete Teilmenge von Zügen. Zwei Züge $r, r' \in R$, die hintereinander in der Sequenz g auftreten, werden mit $(r, r') \in g$ bezeichnet. Für eine gegebene Sequenz $g = \langle r_1, r_2, r_3 \rangle$ gilt beispielsweise $(r_1, r_2) \in g$ und $(r_2, r_3) \in g$, aber $(r_1, r_3) \notin g$. Eine Sequenz, die nach der Ordnung \prec geordnet ist, nennt man **zulässig**. Mit G wird die Menge der zulässigen Sequenzen bezeichnet. Für jedes $g \in G$ und jede Periode $p \in P$ sei $s_p(g)$ die Gesamtlänge der mixed Güterwagen für die Sequenz g in p , d.h.

$$s_p(g) = \sum_{(r,r') \in g} s_p(r, r') \quad .$$

Weiter sei $c(g)$ die Summe aller zusätzlichen Roll-Ins für g :

$$c(g) = \sum_{(r,r') \in g} c(r, r') \quad .$$

Ein **Zeitplan** (engl. Schedule) $f : O \rightarrow G$ ist eine injektive Abbildung von Gleisen zu zulässigen Sequenzen. Eine zulässige Sequenz g kann genau dann auf ein Gleis o eingeplant werden, wenn alle Züge dieser Sequenz auf das Gleis passen, d.h. wenn $g \subseteq R(o)$ gilt. $G(o)$ sei die Menge aller zulässigen Sequenzen, die auf das Gleis o eingeplant werden können. Eine **zulässige Lösung** des Problems kann als Schedule f definiert werden, der

1. jedem Gleis eine zulässige Sequenz zuordnet,

$$\forall o \in O : f(o) \in G(o) \quad ,$$

2. so dass jeder Zug genau in einer Sequenz vorkommt,

$$\forall r \in R : \exists o \in O : r \in f(o) \wedge \forall o' \in O : o \neq o' \rightarrow r \notin f(o') \quad ,$$

3. und so dass in jeder Periode die Kapazität des Mixing-Gleises berücksichtigt wird,

$$\forall p \in P : \sum_{o \in O} s_p(f(o)) \leq s^{\text{mix}} \quad .$$

Eine zulässige Lösung f ist optimal, falls es die Gesamtanzahl der Roll-Ins $\sum_{o \in O} c(f(o))$ minimiert. Abbildung 1.3 illustriert eine zulässige Lösung sowie eine optimale Lösung für die Probleminstanz von Abbildung 1.2.

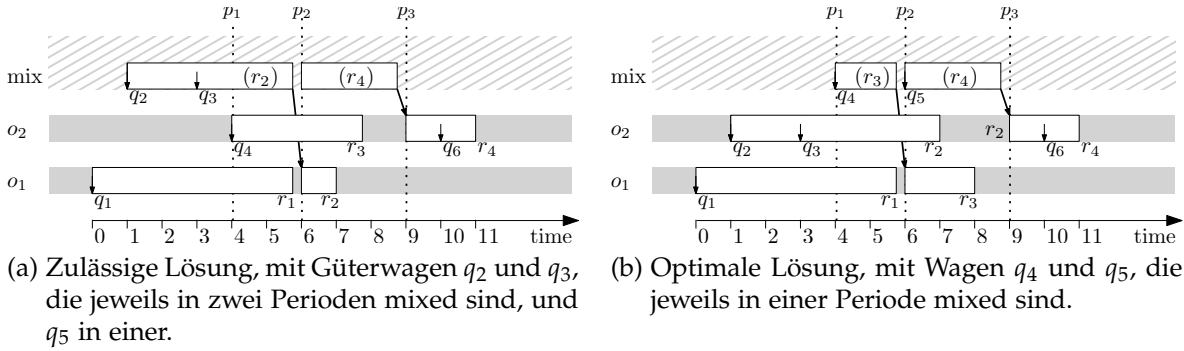


Abbildung 1.3: Zwei zulässige Lösungen für die Probleminstanz in Abb. 1.2. Die Wagen in 1.3a sind in allen fünf Perioden auf dem Mixing-Gleis und in 1.3b in zwei Perioden.

In den folgenden Abschnitten bezeichnen wir mit dem **Rangierproblem**, das in diesem Kapitel eingeführte Problem der Paper [1], [2] und [3].

2 Zusammenhang zu Intervall-Färbungsproblemen

In [2] wird das Rangierproblem als eine Familie von spezifischen Färbungsproblemen von Intervallen betrachtet. In diesem Abschnitt werden wir die Komplexitätsergebnisse, basierend auf dieser Beziehung, angeben und die zwei entwickelten Heuristiken für das Problem kurz einführen (vgl. [2]).

Im Rangierproblem müssen für jeden Outbound-Zug r ein Gleis o und ein Zeitintervall I_r^* bestimmt werden, während dessen das Gleis ausschließlich für die Zusammenstellung dieses Zuges reserviert ist. Unter Beachtung der Roll-In Zeiten der Inbound-Züge, kann für jeden Outbound-Zug r ein Zeitintervall $I_r = (arr, dep)$ angegeben werden, wobei $arr = \min_{q \in Q(r)} t(q)$ und $dep = t(r)$ ist. Somit bezeichnet dieses Intervall den Zeitraum, in dem die Güterwagen von dem Zug r das erste Mal in die Richtungsgruppe ankommen und schließlich diese verlassen. Daher ist o.B.d.A. das Zeitintervall I_r^* ein Teilintervall von I_r der Form $I_r^* = (arr^*, dep)$, $arr^* \geq arr$, d.h. wir schneiden ein *Präfix* von I_r ab, um I_r^* zu erhalten. Wir können kein beliebiges *Präfix* abschneiden: Immer wenn $arr^* \neq arr$ gilt, muss ein Pull-Out des Mixing-Gleises zwischen arr^* und dep existieren, weil gefordert wird, dass jeder Zug pünktlich zusammengestellt werden soll. Dies induziert für jedes Intervall I_r ein unschneidbares Stück von I_r , d.h. ein *Suffix* von I_r , während dessen alle noch nicht eingerollten Wagen des Outbound-Zuges direkt auf ihre zugehörigen Rangiergleise gerollt werden müssen. Die Anforderung, dass jedes Gleis o zu einer bestimmten Zeit nur für höchstens einen Zug reserviert ist, übersetzt sich in die Bedingung, dass, wenn die Züge r und r' dem gleichen Gleis o zugewiesen sind, sich die zugehörigen Intervalle I_r^* und $I_{r'}^*$ nicht überlappen. Die Abbildung 2.1 illustriert dies.

2 Zusammenhang zu Intervall-Färbungsproblemen

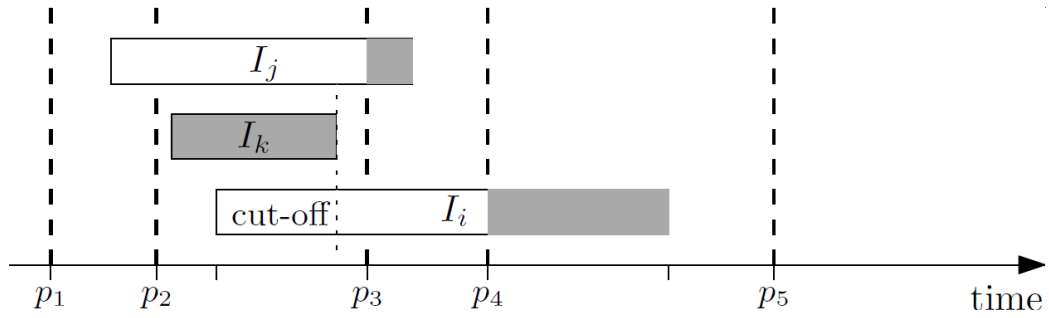


Abbildung 2.1: Drei Outbound-Züge r_i, r_j und r_k induzieren die Intervalle I_i, I_j und I_k , hier dargestellt als Rechtecke. Die Pull-Outs des Mixing-Gleises erfolgen zu den Zeiten p_1, \dots, p_5 . Der unzerschneidbare Abschnitt eines Intervalls ist durch die letzte Pull-Out Zeit des Mixing-Gleises definiert, die im Intervall enthalten ist (dargestellt in grau). Wenn wir annehmen, dass zwei Gleise vorhanden sind, können die drei Intervalle nicht ohne Abschneiden den Gleisen zugewiesen werden. Hier erlaubt das Abschneiden von Intervall I_i am Endpunkt von I_k , dass I_i^* und I_k dem gleichen Gleis zugeordnet werden dürfen.

Unser Problem übersetzt sich in das Problem der Zuordnung eines Gleises o zu jedem Outbound-Zug r und das Abschneiden von einem Präfix von jedem Intervall I_r , um ein Cut-Off Intervall I_r^* zu erhalten, so dass zwei Cut-Off Intervalle zweier Züge, die dem gleichen Gleis zugeordnet sind, sich nicht überlappen. Unter der Annahme, dass das Abschneiden der Intervalle schon durchgeführt wurden ist, kann das Problem der Zuordnung der Gleise ausreichender Länge zu den Outbound-Zügen als ein Listen-Färbungsproblem der Intervalle betrachtet werden. Jeder Zug r besitzt eine Liste L_r von Rangiergleisen, auf die dieser passt und jedes Gleis repräsentiert eine Farbe. Die Aufgabe besteht nun darin, jedes Intervall I_r mit einer Farbe aus der Liste L_r zu färben, so dass zwei sich überlappenden Intervallen $I_{r_1}^*, I_{r_2}^*$ unterschiedliche Farben zugewiesen werden. Allgemein ist das Listen-Färbungsproblem von Intervallen \mathcal{NP} -vollständig. In unserem Fall haben die Listen keine beliebige Struktur. O.B.d.A. wird angenommen, dass die Rangiergleise $o_1, \dots, o_{|O|}$ nach fallender Länge geordnet sind. Für jeden Outbound-Zug r sei $\mu(r)$ das kleinste Gleis, auf das dieser passt (d.h. wir nehmen an, dass jeder Zug auf das erste Gleis mit größter Länge passt und dass das letzte Gleis, auf das dieser passt, das Gleis $o_{\mu(r)}$ ist). Die Liste L_r ist dann $\{o_1, \dots, o_{\mu(r)}\}$. Solch ein Listen-Färbungsproblem wird **μ -Färbungsproblem** genannt. Es ist bekannt, dass das μ -Färbungsproblem für Intervall Graphen \mathcal{NP} -vollständig ist, siehe dazu [9]. Als eine Konsequenz erhalten wir direkt das folgende Theorem.

(2.1) Theorem

Das Finden einer zulässigen Gleiszuordnung für das Rangierproblem ist sogar für Instanzen mit Kapazität des Mixing-Gleises s^{mix} gleich null oder unbegrenzt und für

Intervalle, die alle beliebige unzerschneidbare Abschnitte besitzen, \mathcal{NP} -vollständig.

Trotz des \mathcal{NP} -vollständigen Kerns hängt die praktische Komplexität des Rangierproblems von der Verteilung der Länge der Rangiergleise und der Outbound-Züge ab. Wenn z.B. jeder Zug auf jedes Gleis passen würde und die Kapazität des Mixing-Gleises null wäre, dann würde sich das Problem auf das Problem der Färbung eines Intervall Graphen reduzieren. Dies kann mit Hilfe eines Greedy-Algorithmus in polynomialer Zeit gelöst werden.

Unter der Annahme, dass jeder Zug auf jedes Gleis passt und dass das Mixing-Gleis eine hinreichende Länge besitzt, kann das Problem in polynomialer Zeit gelöst werden.

(2.2) Theorem

Falls die Gleislängen einheitlich und hinreichend groß sind, ist das Rangierproblem, welches die Summe aller Güterwagen, die über alle Perioden mixed sind, minimiert, in polynomialer Zeit lösbar.

Für die Beweise der Theoreme (2.1) und (2.2) verweisen wir auf [2].

2.1 Eine Cliquenbasierte Heuristik

In diesem Unterkapitel wird eine Heuristik für das Rangierproblem präsentiert, in der iterativ eine Färbung der Intervalle bestimmt und die (problematischen) Intervalle abgeschnitten werden. Das Hauptziel wird dabei sein, die Gesamtlänge aller Güterwagen, die mixed werden, summiert über alle Perioden, niedrig zu halten. Idealerweise wird die Heuristik eine Lösung finden, die in Bezug auf die Kapazität des Mixing-Gleises in jeder Periode zulässig ist. Für jeden Outbound-Zug r muss ein *Suffix* I_r^* für jedes Intervall I_r gewählt und mit einer Farbe aus der Liste L_r gefärbt werden. Die Annahme, dass jeder Outbound-Zug auf das längste Gleis $o_{|O|}$ passt, wird beibehalten.

Färbung der Intervalle:

Zuerst werden die Intervalle greedy gefärbt. Unendlich viele Farben stehen zur Verfügung (d.h. nicht nur $|O|$ verschiedene). Begonnen wird mit dem Sortieren der Intervalle in einer chronologischen Reihenfolge ihrer Startzeitpunkte. Die Intervalle werden in dieser Reihenfolge gefärbt und jedem Intervall I_r wird die kleinste Nicht-Konfliktfarbe, die mindestens $\mu(r)$ ist, zugewiesen. An dieser Stelle ist eine Nicht-Konfliktfarbe eine Farbe, so dass kein Intervall, welches gefärbt ist und sich mit I_r überlappt, die gleiche Farbe wie I_r hat. Auf diese Weise wird garantiert, dass

jedem Intervall I_r eine Farbe mindestens $\mu(r)$ zugeordnet ist. Falls nicht mehr Farben als $|O|$ benutzt werden, hat man eine Listen-Färbung gefunden und kann die Iterationen stoppen. Andernfalls verwendet die Färbung mehr Farben als $|O|$ und es wird mit dem Abschneiden begonnen.

Abschneiden der Intervalle:

Wenn es ein Intervall I_r gibt, welches mit einer Farbe $c > |O| = k$ gefärbt ist, dann überlappt sich I_r mit Intervallen, denen jede Farbe in $\mu(r), \mu(r) + 1, \dots, |O|, \dots, c$ zugeordnet ist. Diese Intervalle überschneiden sich und bilden so eine Clique K . Wir finden eine maximale Clique, die ein Intervall der größten zugewiesenen Farbe c enthält. Es sei $q := c - k$, d.h. die Anzahl der Intervalle von K , die eine Farbe $c > k$ benutzen. Die Heuristik versucht die Größe der Clique durch Abschneiden von q Intervallen von K zu reduzieren, um die Anzahl der verfügbaren Farben für jedes dieser Intervalle zu vergrößern. Die Cut-Offs werden wie folgt berechnet. Zunächst wird der Schnitt aller Intervalle in K berechnet, welches auch ein Intervall ist. Sei t das Ende dieses Intervalls. Wenn möglich, werden q der Intervalle von K am Punkt t abgeschnitten. Insbesondere werden die Intervalle q von K abgeschnitten, die die zusätzliche Nutzung des Mixing-Gleises minimieren. Die Prozedur (Färben und Abschneiden) wird mit den neuen Cut-Off Intervallen durchlaufen. Dieser Vorgang wird solange wiederholt, bis höchstens $|O|$ Farben benutzt werden.

Die Heuristik ist in Abbildung 2.2 dargestellt. Diese garantiert nicht, dass die gefundene Zuordnung hinsichtlich der Kapazität des Mixing-Gleises zulässig ist.

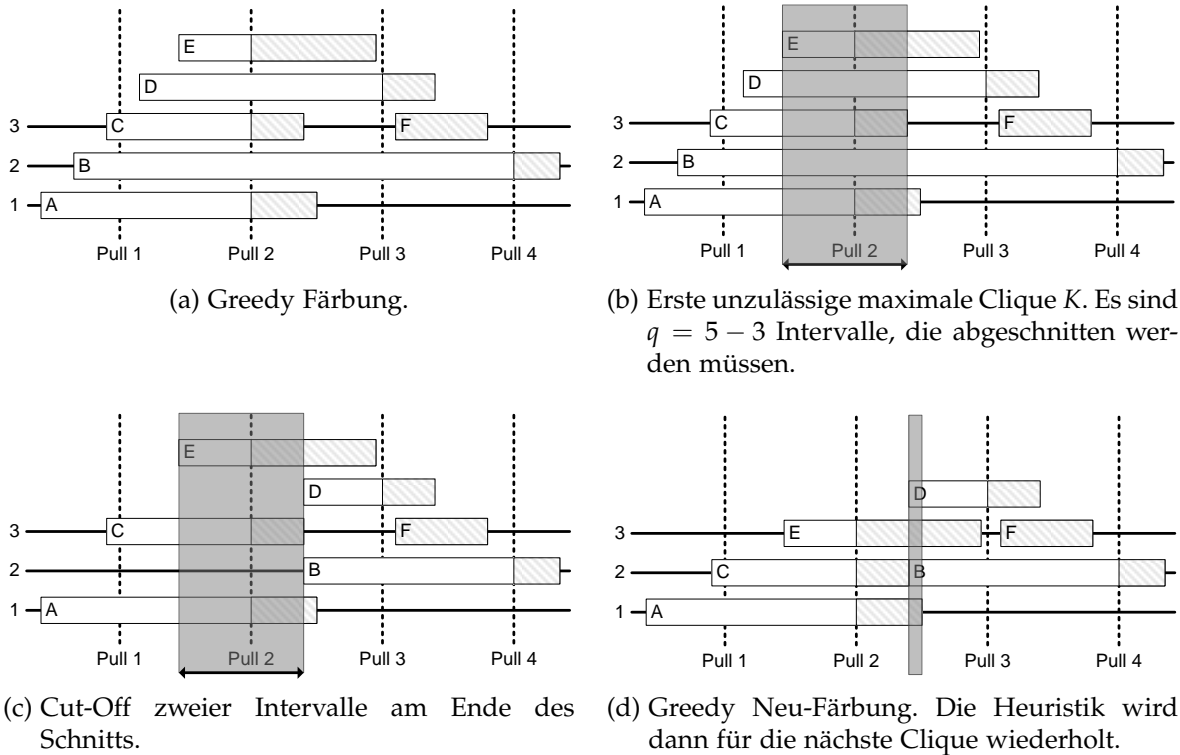


Abbildung 2.2: Die ersten Schritte der Intervall-Färbungsheuristik mit Zügen $A - F$, und drei Rangiergleisen. Die grauen Bereiche der Intervalle zeigen Zuweisungen, die nicht abgeschnitten werden können, da kein Pull-Out (gestrichelte vertikale Linien) des Mixing-Gleises existiert. Das dunkelgraue Rechteck stellt den Schnittpunkt der jeweiligen Clique von Intervallen dar.

2.2 Eine Verbesserungsheuristik

Sobald eine zulässige Zuordnung der Gleise zu den Outbound-Zügen existiert, kann man versuchen, die Lösung zu einem lokalen Optimum zu verbessern. Das Interesse liegt in der Minimierung der Gesamtzahl der Güterwagen (über alle Perioden), die mixed werden. Diese werden kurz als zusätzliche Roll-Ins bzw. extra Roll-Ins bezeichnet.

Die Heuristik basiert auf zwei Beobachtungen. Erstens, wenn man sich eine zulässige Lösung anschaut, so beobachtet man, dass sowohl die Gleise als auch die Züge in Teilmengen, genannt **Buckets**, partitioniert werden können, so dass jeder Zug in einem Bucket auf alle Gleise in dem gleichen Bucket passt. Zweitens, wenn eine

2 Zusammenhang zu Intervall-Färbungsproblemen

zulässige Lösung gegeben ist, kann man die Nutzung des Mixing-Gleises für jeden Bucket von Gleisen unabhängig voneinander minimieren. Weil in einem Bucket alle zugeordneten Züge auf alle Gleise passen, genügt es, ein Zuordnungsproblem für jedes Bucket, wie in (2.2) beschrieben, zu lösen, um eine optimale Neuordnung von Zügen auf die Gleise innerhalb dieses Buckets zu finden.

Zu beachten ist, dass auch diese Heuristik keine Zulässigkeitsgarantie hinsichtlich der Kapazität des Mixing-Gleises liefert.

3 Ein binäres Programm für das Rangierproblem

In diesem Kapitel werden wir anfänglich eine Einführung in die diskrete Optimierung geben, um die wichtigsten Terme, Notationen und Definitionen einzuführen, die in dieser Arbeit verwendet werden. Mit Hilfe dieser wird in [2] ein binäres Programm für das Rangierproblem entworfen, welches wir in diesem Abschnitt darstellen werden (vgl. [2]).

3.1 Einführung in die diskrete Optimierung

Für eine gegebene Menge von reellwertigen Variablen ist ein **lineares Programm** ein Optimierungsproblem, welches eine lineare Zielfunktion entweder minimiert oder maximiert, sodass einige lineare Gleichungen bzw. Ungleichungen erfüllt sind. Mit verschiedenen Transformationen können wir jedes lineare Programm in die Form verwandeln, die in der folgenden Definition vorgestellt wird.

(3.1) Definition

Seien $n, m \in \mathbb{N}$, $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{m \times n}$ und $\mathbf{b} \in \mathbb{R}^m$. Ein Optimierungsproblem der Form

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{Ax} \geq \mathbf{b} \\ & \mathbf{x} \in \mathbb{R}_+^n \end{aligned}$$

wird **lineares Programm** (engl.: linear program) LP genannt.

Durch Hinzufügen von Ganzzahligkeitsbedingungen für einen Teil der Variablen eines LPs erhalten wir ein **gemischt-ganzzahliges Programm**.

(3.2) Definition

Seien $n, m \in \mathbb{N}$, $\mathbf{c} \in \mathbb{Q}^n$, $\mathbf{A} \in \mathbb{Q}^{n \times m}$, $\mathbf{b} \in \mathbb{Q}^m$ und $I \subseteq \bar{n} := \{1, \dots, n\}$. Ein Optimierungsproblem der Form

$$\begin{aligned} \min \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A} \mathbf{x} \geq \mathbf{b} \\ & \mathbf{x} \in \mathbb{Q}_+^n \\ & x_i \in \mathbb{Z}, \forall i \in I \end{aligned} \tag{3.1}$$

wird **gemischt-ganzzahliges Programm** (engl.: mixed integer program) MIP genannt.

Wir unterscheiden die folgenden besonderen Fälle von MIPs.

(3.3) Definition

Ein MIP der Form (3.1) wird

- ein **ganzzahliges Programm** (engl.: integer program) IP genannt, falls $I = \bar{n}$ gilt.
- ein **binäres Programm** (engl.: binary program) BP genannt, falls es ein IP ist und zusätzlich $x_i \in \{0, 1\} \forall i \in I$ gilt.

Es existieren mehrere Methoden, die LPs in polynomialer Zeit lösen können. Hinzufügen von Ganzzahligkeitsbedingungen erhöht die Komplexität des Problems. Das Lösen von MIPs ist \mathcal{NP} -schwer. Eine Methode zur Lösung von MIPs ist die LP basierte **Branch-and-Bound** Methode. In Kapitel 4.1 werden wir den **Branch-and-Price** Algorithmus kennenlernen, der eine Variante der Branch-and-Bound Methode ist, wobei in jedem Knoten die LP-Relaxation mit Hilfe der Spaltengenerierung gelöst wird.

Um nun eine exakte Lösung für das Rangierproblem zu berechnen, wird ein ganzzahliges Programm entworfen. Das Rangierproblem besteht darin, eine Zuordnung der Rangiergleise zu den Outbound-Zügen und für jeden Outbound-Zug eine konfliktfreie Zeitreservierung der ihm zugewiesenen Gleise zu finden. Dabei müssen alle Outbound-Züge pünktlich zusammengestellt werden können und die Kapazität des Mixing-Gleises darf nicht überschritten werden.

3.2 Die direkte Formulierung

Das Modell basiert auf der Beobachtung, dass sowohl die Zahl der zusätzlichen Roll-Ins als auch der Betrag der genutzten mixing Kapazität in jeder Periode nur von den Zeiten abhängt, zu denen die Züge aktiv werden (vgl. [2]). Ferner gilt, dass nur wenige Zeitpunkte relevant sind, damit ein Zug aktiv wird. Nämlich jedes Mal, wenn einer der Wagen des Zuges ankommt oder auf dem Mixing-Gleis einer Pull-Out Operation unterliegt. Bezeichne $\mathcal{T}(r)$ die Menge aller solcher relevanten Zeitpunkte für Zug r . Binäre Variablen x_{rt} geben an, zu welchem Zeitpunkt t die Reservierung von einem Rangiergleis für Outbound-Zug r beginnt, d.h. wann Zug r aktiv wird. Weiterhin bezeichnen binäre Variablen y_{ro} , ob der Outbound-Zug r dem Gleis o zugeordnet wird oder nicht.

(DF)

$$\min \sum_{r \in R} \sum_{t \in \mathcal{T}(r)} c(r, t) \cdot x_{rt} \quad (3.2)$$

$$\text{s.t.} \quad \sum_{o \in O(r)} y_{ro} = 1, \quad r \in R \quad (3.3)$$

$$\sum_{t \in \mathcal{T}(r)} x_{rt} = 1, \quad r \in R \quad (3.4)$$

$$\sum_{t \in \mathcal{T}(r'): t < t(r)} x_{r't} + y_{ro} + y_{r'o} \leq 2, \quad r \prec r', o \in O(r) \cap O(r') \quad (3.5)$$

$$\sum_{r \in R} s_p(r, t) \cdot x_{rt} \leq s^{\text{mix}}, \quad p \in P \quad (3.6)$$

$$x, y \in \{0, 1\}. \quad (3.7)$$

Die Zielfunktion (3.2) gibt das Ziel in Bezug auf die Zahl $c(r, t)$ der zusätzlichen Roll-Ins an, welche dadurch entstehen, dass die Startzeit t für Zug r benutzt wird. Die Gleichungen (3.3) sichern, dass jeder Zug $r \in R$ einem Gleis zugeordnet ist, auf das dieser passt. Die Restriktionen (3.4) garantieren, dass jeder Zug $r \in R$ zu einem relevanten Zeitpunkt $t \in \mathcal{T}(r)$ aktiv wird. Die Nebenbedingungen (3.5) sichern, dass jedes Paar von Zügen $r \prec r'$, die hintereinander auf dem gleichen Gleis zusammengestellt werden können, entweder auf verschiedenen Gleisen eingeplant sind, oder dass r' erst aktiv wird, wenn r abfährt. Schließlich werden durch die Ungleichungen (3.6) die mixing Kapazität in jeder Periode berücksichtigt, wobei $s_p(r, t)$ die Länge aller Güterwagen von r bezeichnet, die sich in der Periode p auf dem Mixing-Gleis befinden, wenn r zum Zeitpunkt t aktiv wird.

Scheduling Probleme liefern im Allgemeinen schwache LP-Relaxationen. Dies gilt auch für die binäre Formulierung (3.2) - (3.7), wie später in Kapitel 4.4 bestätigt wird.

4 Ein Spaltengenerierungsansatz für das Rangierproblem

Um das Problem der schwachen Formulierung im vorherigen Kapitel zu umgehen, wird das Rangierproblem in [3] mit Hilfe eines Spaltengenerierungsansatzes gelöst. Spaltengenerierung ist eine klassische Technik, um ein mathematisches Programm iterativ durch Hinzufügen der Variablen des Modells zu lösen. In der Regel wird nur ein Bruchteil der Variablen benötigt, um die Optimalität der Technik zu beweisen. Diese Tatsache macht dieses Verfahren interessant für Probleme mit einer großen Anzahl von Variablen. Das Verfahren startet mit einer Formulierung des Problems bzw. seiner LP-Relaxation, die eine große Anzahl von Variablen, d.h. Spalten aufweist. Da diese Formulierung exponentiell in der Anzahl der Variablen ist, löst man dieses sogenannte **Master-Problem** (MP) nicht mit allen Spalten. Statt dessen beginnt man mit einer kleinen Menge von Spalten und löst dieses sogenannte **restricted Master-Problem** (RMP). Durch Lösen des **Pricing-Problems** werden nun Variablen bestimmt, deren Aufnahme in das (RMP) vorteilhaft erscheint. Dieses Generieren von Variablen bzw. Spalten für ein LP wird **Spaltengenerierung** (engl.: Column-Generation) genannt. Es entspricht exakt der Aufnahme einer Nicht-Basisvariable in die Basis beim Simplexalgorithmus. Bei **Branch-and-Price** wird dieser Ansatz in ein Branch-and-Bound-Verfahren integriert, um die Ganzzahligkeit einer Lösung zu erreichen. Beim Column-Generation wechseln sich Reoptimierung des (RMP)s und Pricing ab. Allgemein sehen das Master-Problem, das restricted Master-Problem und das entsprechende duale Problem wie folgt aus:

(MP)

$$z_{MP}^* = \min \sum_{j \in J} c_j \lambda_j \quad (4.1)$$

$$\text{s.t.} \quad \sum_{j \in J} \mathbf{a}_j \lambda_j \geq \mathbf{b} \quad (4.2)$$

$$\lambda_j \geq 0, j \in J, \quad (4.3)$$

(RMP)

$$z_{RMP}^* = \min \sum_{j \in J} c_j \lambda_j \quad (4.4)$$

$$\text{s.t.} \quad \sum_{j \in J} \mathbf{a}_j \lambda_j \geq \mathbf{b} \quad (4.5)$$

$$\lambda_j \geq 0, j \in J^- \subseteq J, \quad (4.6)$$

(DMP)

$$z_{DMP}^* = \max \sum_{i \in I} b_i \pi_i \quad (4.7)$$

$$\text{s.t.} \quad \sum_{i \in I} \mathbf{a}_i^T \pi_i \leq \mathbf{c} \quad (4.8)$$

$$\pi_i \geq 0, i \in I. \quad (4.9)$$

4.1 Erweiterte Formulierung

Um die Probleme mit der direkten Formulierung (DF) zu umgehen, führen wir die erweiterte Formulierung aus [3] ein, in der die Variablen Zuordnungen von gesamten Sequenzen zu den Gleisen darstellen. Wir werden sehen, dass diese Formulierung effizient durch Branch-and-Price gelöst werden kann und zu einer starken LP-Relaxation für die Probleminstanzen führt (vgl. [3]).

Für jedes Gleis o und jede mögliche Sequenz $g \in G(o)$ wird eine Variable verwendet x_{go} , die angibt, ob die Sequenz g auf dem Gleis o benutzt wird oder nicht. Wie wir in Abschnitt 1.1 gesehen haben, ist es ausreichend, die zugehörige Sequenz für jedes Gleis zu kennen, um die Nutzung des Mixing-Gleises und die Anzahl der extra Roll-Ins eines Schedules zu bestimmen. Darüber hinaus wird jedes Sequenz-Gleis-Paar, welches in der endgültigen Lösung enthalten ist, diese Größen unabhängig von allen anderen Paaren erhöhen, welches uns ermöglicht ein Modell mit einer linearen Zielfunktion und linearen Nebenbedingungen zu konstruieren. Um das Branching zu erleichtern, werden auch die Variablen y_{ro} aus dem vorherigen Abschnitt verwendet, die angeben, ob der Zug r dem Gleis o zugeordnet wird oder nicht.

Das vollständige ganzzahlige Programm für die erweiterte Formulierung (EF) sieht wie folgt aus:

(EF)

$$\min \sum_{\substack{o \in O \\ g \in G(o)}} c(g) \cdot x_{go} \quad (4.10)$$

$$\text{s.t.} \quad \sum_{o \in O} y_{ro} \geq 1 \quad r \in R \quad (4.11)$$

$$\sum_{\substack{g \in G(o) \\ r \in g}} x_{go} \geq y_{ro} \quad r \in R, o \in O \quad (4.12)$$

$$\sum_{g \in G(o)} x_{go} \leq 1 \quad o \in O \quad (4.13)$$

$$\sum_{\substack{o \in O \\ g \in G(o)}} s_p(g) \cdot x_{go} \leq s^{\text{mix}} \quad p \in P \quad (4.14)$$

$$x, y \in \{0, 1\}. \quad (4.15)$$

Die Zielfunktion (4.10) minimiert die Gesamtzahl der zusätzlichen Roll-Ins. Die Ungleichungen (4.11) und (4.12) gewährleisten, dass jeder Zug in einer Sequenz erscheint. Wir fordern keine Gleichheit, da die mehrmalige Benutzung eines einzigen Zuges die Zielfunktion nie verbessern kann. Dadurch schränken wir die dualen Variablen im Vorzeichen ein und "halbieren" somit den dualen Zulässigkeitsbereich. Falls ein einzelner Zug mehrmals in einer optimalen Lösung auftritt, kann es von allen bis auf einer Sequenz entfernt werden. Die Ungleichungen (4.13) sichern, dass höchstens eine Sequenz pro Gleis verwendet werden kann und die Nebenbedingungen (4.14) garantieren, dass wir nicht mehr als die verfügbare mixing Kapazität in jeder Periode verwenden. Die Restriktionen (4.11) – (4.14) sind zu den Bedingungen einer zulässigen Lösung aus Kapitel 1.1 äquivalent.

In dem obigen Modell gibt es eine x Variable für jede Kombination aus Sequenz $g \in G(o)$ und Gleis $o \in O$. Da die Größe von $G(o)$ von der Ordnung $\mathcal{O}(|R|!)$ ist, wird anfänglich nur eine Teilmenge von x Variablen betrachtet und Spaltengenerierung wird verwendet, um neue erforderliche Variablen zu generieren. Die Anzahl der y_{ro} ist klein, weshalb alle diese Variablen von Anfang an im Modell enthalten sind.

Um Column-Generation zu verwenden, schauen wir uns das duale Problem der LP-Relaxation von (EF) an:

$$\max \quad \sum_{r \in R} \alpha_r + \sum_{o \in O} \gamma_o + s^{\text{mix}} \sum_{p \in P} \delta_p \quad (4.16)$$

$$\text{s.t.} \quad \alpha_r \leq \beta_{ro} \quad r \in R, o \in O \quad (4.17)$$

$$\sum_{r \in g} \beta_{ro} + \gamma_o + \sum_{p \in P} s_p(g) \cdot \delta_p \leq c(g) \quad o \in O, g \in G(o) \quad (4.18)$$

$$\alpha, \beta \geq 0 \quad \gamma, \delta \leq 0. \quad (4.19)$$

Die dualen Variablen sind wie folgt gewählt: die α Variablen entsprechen den primalen Nebenbedingungen (4.11), β den (4.12), γ den (4.13) und schließlich δ den (4.14). Für jede primale Variable x erhalten wir eine Ungleichung der Form (4.18) und für jedes y eine Nebenbedingung der Form (4.17).

Wie oben erwähnt, beginnen wir mit dem Lösen des restricted Master-Problems. Eine optimale Lösung des (RMP)s braucht natürlich nicht optimal für das Master-Problem zu sein. Wenn das (RMP) primal unbeschränkt ist, dann gilt dies auch für das (MP). Wir können mit dem Lösungsprozess aufhören, da wir gezeigt haben, dass das Master-Problem unbeschränkt ist. Falls das (RMP) primal unzulässig ist, könnte das (MP) ebenfalls primal unzulässig sein. In den meisten Fällen müssen jedoch nur weitere Variablen hinzugefügt werden, um die Zulässigkeit wiederherzustellen. Insbesondere, wenn wir das (RMP) zum ersten Mal lösen, können wir ohne Variablen starten, so dass das (RMP) nicht zulässig ist. Der Prozess, der nach Variablen sucht, die helfen die Unzulässigkeit zu zerstören, heißt **Farkas Pricing**. Falls wir eine optimale Lösung für das (RMP) erhalten, wissen wir nicht, ob diese auch optimal für das Master-Problem ist. Daher müssen wir nach Variablen suchen, die diese Lösung verbessern und fügen diese dem (RMP) hinzu. Wir bezeichnen dieses Verfahren als **(reduzierte Kosten) Pricing**, da wir dem (RMP) Variablen mit negativen reduzierten Kosten hinzufügen. Nachdem Variablen hinzugefügt werden, wird das (RMP) erneut gelöst und der Prozess wird wiederholt. Dieser wird solange wiederholt, bis keine weitere Variable die aktuelle Lösung verbessern kann oder helfen kann, Unzulässigkeit zu beweisen. Wenn das (RMP) immer noch unzulässig ist, ist es das (MP) ebenfalls. Falls das (RMP) eine optimale Lösung besitzt, dann ist diese Lösung auch optimal für das Master-Problem. Sobald das (RMP) zulässig ist, bleibt es auch zulässig, da das Hinzufügen von Variablen die Zulässigkeit nicht verletzt. Daher verwenden wir Farkas Pricing bis das (RMP) zulässig wird und beginnen dann mit dem Pricing.

Aus der Dualitätstheorie wissen wir, dass die optimale Lösung der (EF) eine entsprechende duale Lösung hat, die alle Ungleichungen in (4.18) erfüllt. Wenn also die Lösung des (RMP)s alle Nebenbedingungen (4.18) erfüllt, ist diese auch optimal für

die (EF). Wenn nicht, müssen wir die Variablen, die den verletzten Ungleichungen entsprechen, dem (RMP) hinzufügen und von vorne anfangen, bis keine Restriktion mehr verletzt ist. Das Pricing-Problem dient nun dazu, verletzte Nebenbedingungen (4.18) zu identifizieren, ohne jede einzeln zu überprüfen (was auf Grund der großen Anzahl ineffizient wäre). In Abschnitt 4.2 geben wir die detaillierte Beschreibung aus [3], wie das Pricing-Problem funktioniert, wieder.

Wenn wir eine optimale LP-Lösung der (EF) haben, wird diese möglicherweise nicht ganzzahlig sein. Daher wird mit Hilfe eines Branch-and-Price Verfahrens die Ganzzahligkeit erzeugt, wobei darauf geachtet werden muss, dass keine unzulässigen Zugsequenzen während des Branchens hinzugefügt werden. In Abschnitt 4.3 werden wir darstellen, wie eine Branching-Regel aussehen könnte und wie wir diese Entscheidungen im Pricing-Problem berücksichtigen (vgl. [3]).

4.2 Pricing

Das Pricing wird eingesetzt, wenn das (RMP) optimal gelöst ist, sodass eine optimale primale Lösung λ^* und die zugehörige duale Lösung π^* vorhanden sind. Die reduzierten Kosten einer nicht-negativen primalen Variable λ_j mit Zielfunktionskoeffizienten c_j und Koeffizienten \mathbf{a}_j , in Bezug auf die Nebenbedingung (4.5), ist für die aktuelle duale Lösung π gegeben durch $c_j - \pi^T \mathbf{a}_j$. Die zugehörigen dualen Nebenbedingungen lauten $\sum_{i \in I} \mathbf{a}_i^T \pi_i \leq \mathbf{c}$, d.h. die duale Restriktion ist genau dann verletzt, wenn die zugehörige primale Variable negative reduzierte Kosten hat. Daher ist die Suche nach einer Variable, die die optimale Lösung des (RMP)s verbessern kann, gleich der Suche nach einer Nebenbedingung von Typ (4.8) im dualen Master-Problem, die durch die optimale duale Lösung π^* des (RMP) verletzt ist. In dem Pricing-Schritt des Simplex-Verfahrens suchen wir eine Nicht-Basisvariable mit negativen reduzierten Kosten, um diese in die Basis aufzunehmen. Um dies in Spaltengenerierung zu erreichen, löst man das Pricing-Problem

$$\min \left\{ c_j - (\pi^*)^T \mathbf{a}_j \mid j \in J \right\}. \quad (4.20)$$

Falls (4.20) < 0 ist, werden die Variable λ_j und dessen Koeffizientenspalte (c_j, \mathbf{a}_j) dem (RMP) hinzugefügt. Dieses wird optimal gelöst, um optimale duale Variablen zu erhalten. Dieser Prozess wird wiederholt, bis keine verbessernde Variable mehr gefunden werden kann. In diesem Fall löst λ^* auch das Master-Problem optimal.

Es scheint nicht klar, warum das Pricing-Problem eine Hilfe darstellen soll, wenn $|J|$ groß ist. Doch in fast jeder Anwendung zählen die Indizes in J Einheiten, welche als Zulässigkeitsbereich X eines Optimierungsproblems

$$\min_{\mathbf{x} \in X} \left\{ c(\mathbf{x}) - (\boldsymbol{\pi}^*)^T a(\mathbf{x}) \right\}$$

beschrieben werden können, wobei $c_j = c(\mathbf{x}_j)$ und $\mathbf{a}_j = a(\mathbf{x}_j)$ und $\mathbf{x}_j \in X$, für $j \in J$ gilt. Das heißt, anstatt explizit alle Variablekandidaten zu pricen, muss ein gut strukturiertes Optimierungsproblem gelöst werden, so dass die Suche nach einer Variablen mit negativen reduzierten Kosten implizit wird.

Falls das (RMP) unzulässig ist, liefert das Farkas-Lemma eine Möglichkeit dies zu beweisen. Es benutzt die bekannte Tatsache, dass der duale Wert eines unzulässigen linearen Programms unbeschränkt ist. Dies wird in dem Farkas-Lemma formalisiert.

(4.1) Lemma

Sei \mathbf{A} eine $m \times n$ Matrix und $\mathbf{b} \in \mathbb{R}^m$. Dann gilt:

- Entweder gibt es ein $\mathbf{x} \in \mathbb{R}^n$ mit $\mathbf{Ax} = \mathbf{b}$ und $\mathbf{x} \geq \mathbf{0}$,
- oder es gibt ein $\boldsymbol{\pi} \in \mathbb{R}^m$ mit $\boldsymbol{\pi}^T \mathbf{A} = \mathbf{0}$ und $\boldsymbol{\pi}^T \mathbf{b} > 0$.

Ein solcher Vektor beweist die Unzulässigkeit des ersten Systems, da $\boldsymbol{\pi}^T \mathbf{Ax} = \boldsymbol{\pi}^T \mathbf{b}$ nicht erfüllt werden kann. Die Idee besteht nun darin, der Matrix \mathbf{A} eine Variable mit Spaltenkoeffizient \mathbf{a} und $\boldsymbol{\pi}^T \mathbf{a} > 0$ hinzuzufügen, welche so diesen Beweis der Unzulässigkeit zerstört. So eine Variable kann durch Lösen von

$$\max_{\mathbf{x} \in X} \left\{ (\boldsymbol{\pi}^*)^T a(\mathbf{x}) \right\} = \min_{\mathbf{x} \in X} \left\{ -(\boldsymbol{\pi}^*)^T a(\mathbf{x}) \right\}$$

gefunden werden. Der Vektor $\boldsymbol{\pi}$ wird typischerweise von dem LP-Solver, im Falle eines unzulässigen linearen Programms, mitgeliefert. Zu beobachten ist, dass beide Pricing-Probleme sehr ähnlich sind.

Nun werden wir uns das Pricing-Problem für das Rangierproblem anschauen und erklären, wie wir effizient Variablen durch Finden von verletzten Nebenbedingungen (4.18) ermitteln können, die das (RMP) verbessern. Da die Menge O nicht zu groß ist, können wir uns die Ungleichungen für jedes Gleis separat anschauen. Somit müssen wir für jedes Gleis $o \in O$ eine Sequenz $g \in G(o)$ finden, für die die Ungleichung

$$\sum_{r \in g} \beta_{r,o} + \sum_{p \in P} s_p(g) \cdot \delta_p - c(g) \leq -\gamma_o$$

verletzt ist. Es können viele solcher Ungleichungen existieren und wir suchen die Ungleichung, die am meisten verletzt ist, d.h. wir möchten die Linke-Seite maximieren:

$$\max_{g \in G(o)} \sum_{r \in g} \beta_{r,o} + \sum_{p \in P} s_p(g) \cdot \delta_p - c(g) \quad .$$

Dazu benutzen wir die Tatsache, dass $s_p(g)$ und $c(g)$ als Summen über Paare von Zügen berechnet werden, die in einer Sequenz hintereinander erscheinen, d.h. die Größen hängen nur von dem unmittelbaren Vorgänger ab (siehe Kapitel 1.1).

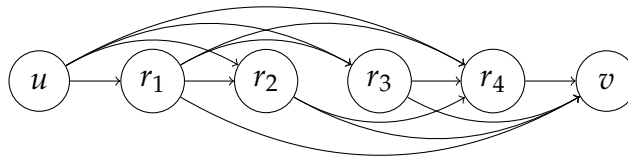


Abbildung 4.1: Längste Wege Graph für das Wurzel Pricing-Problem für Gleis o_2 aus dem Beispiel in Abbildung 1.2.

Zuerst konstruieren wir einen gerichteten Graph $G = (V, E)$ mit der Knotenmenge $V = R(o) \cup \{u, v\}$, d.h. einen Knoten für jeden Zug, der auf das Gleis o passt plus zwei weitere Knoten. Die Kantenmenge E beinhaltet die Kanten (u, r) und (r, v) für jeden Zug $r \in R(o)$ und eine Kante (r_1, r_2) falls $r_1 \prec r_2$ gilt. Jeder Weg von u zu v in G entspricht einer zulässigen Sequenz $g \in G(o)$. Die Abbildung 4.1 zeigt wie dieser Graph für das Beispiel, welches in Abbildung 1.2 dargestellt wurde, aussieht, wenn wir eine neue Sequenz für das Gleis o_2 suchen.

Als nächstes fügen wir G Kantengewichte hinzu. Wir wählen folgende Gewichte:

- $w_{(u,r)} = \beta_{r,o}$
- $w_{(r_1,r_2)} = \beta_{r_2,o} + \sum_{p \in P} s_p(r_2, r_1) \cdot \delta_p - c(r_2, r_1)$
- $w_{(r,v)} = 0$

Für jede Sequenz $g \in G(o)$ existiert ein äquivalenter Weg in G mit Gesamtgewicht

$$\sum_{r \in g} \beta_{ro} + \sum_{p \in P} s_p(g) \cdot \delta_p - c(g) \quad .$$

Da dies die Größe ist, die wir maximieren möchten, können wir einen längsten Weg von u nach v in G suchen. Auf Grund der partiellen Ordnung der Züge ist G zyklensfrei und ein längster Weg kann in $\mathcal{O}(|V| + |E|)$ Zeit berechnet werden (siehe [10]). In unserem Fall wäre dies $\mathcal{O}(|R|^2)$, da der Graph nahe zu vollständig ist (d.h. vollständig bis auf die Kante (u, v) , falls \prec eine Totalordnung ist).

4.3 Branching

Man kann nicht erwarten, dass nach Lösung des (MP)s alle Variablen ganzzahlig sind. Zur Lösung ganzzahliger Programme können z.B. Branch-and-Bound-Verfahren benutzt werden. Dieser Abschnitt beschäftigt sich daher mit **Branch-and-Price**, d.h. einem Verfahren, welches eine Kombination aus Column-Generation und Branch-and-Bound ist. Zuerst erwähnen wir, dass es nicht ausreicht, die durch die Lösung des (RMP) gefundenen Spalten allein einem Branch-and-Bound-Verfahren zur Herstellung von Ganzzahligkeitsbedingungen zu unterwerfen. Fügt man nämlich dem (RMP) weitere Restriktionen hinzu, mit denen die Ganzzahligkeit erzwungen wird, so ändern sich die dualen Preise. Variablen, deren Aufnahme in die Basis bisher nicht vorteilhaft war, könnten nun negative reduzierte Kosten aufweisen und im Pricing als Lösung erscheinen. Eine weitere Schwierigkeit kommt hinzu. Neu hinzugefügte Restriktionen verändern innerhalb des Branch-and-Bound-Suchbaums die Struktur des Pricing-Problems. So bedeutet dieses Verzweigen, dass die Variablen, die im entsprechenden Teilbaum ausgeschlossen werden, auch nicht als Lösungen des Pricing-Problems neu erzeugt werden dürfen. Die Branching-Strategie muss demzufolge so gewählt werden, dass sie auf das Subproblem übertragbar ist - Branching und Pricing müssen kompatibel sein.

Wenn die LP-Relaxation der (EF) in einem der Branch-and-Bound Bäumen zugehörigen Knoten gelöst worden ist und die Lösung fraktional ist, muss gebrancht werden. Allerdings ist das Branching auf den x Variablen problematisch, da in dem Zweig $x_{go} = 0$ nur eine Sequenz g auf dem Gleis o ausgeschlossen wird. Das Pricing-Problem müsste daher einzelne Wege ausschließen, was das Problem \mathcal{NP} -schwer macht.

Wir können dieses Problem umgehen, indem wir stattdessen auf den Zuordnungen der Züge r zu den Gleisen o branchen. Diese entsprechen den originalen y_{ro}

Variablen aus dem direkten Modell. In jedem fraktionalem Knoten wählen wir eine fraktionale Variable y_{r_0} (so eine Variable muss existieren, da die Lösung sonst nicht fraktional wäre). Das Problem wird dann in die Fälle $y_{r_0} = 0$ und $y_{r_0} = 1$ aufgeteilt. Als nächstes müssen wir überlegen, wie wir sicherstellen, dass die Branching-Entscheidungen im Pricing-Problem berücksichtigt werden. Dies kann durch Änderung des Graphens, welcher im Pricing benutzt wird, erreicht werden (siehe Abschnitt 4.2). Wir schauen uns die zwei möglichen Fälle an:

Fall $y_{r_0} = 0$: Entferne Knoten r von der Knotenmenge V und alle Kanten, die mit diesem verbunden sind. Auf diese Weise wird das Generieren einer Sequenz, die Zug r enthält, verboten.

Fall $y_{r_0} = 1$: Entferne für alle Knoten r' und r'' mit $t(r') < t(r) < t(r'')$, die Kanten (r', r'') zusammen mit (u, r'') und (r', v) , von der Kantenmenge E . Entferne ebenfalls alle Knoten r' , für die $r' \parallel r$ gilt, zusammen mit all ihren Kanten. Nun werden alle Wege von einem Knoten vor r zu einem nach r auch r enthalten. Aus diesem Grund kann kein Weg von u nach v den Knoten r überspringen.

Um dies zu illustrieren, betrachten wir den Beispielgraphen aus Abbildung 4.1. Abbildung 4.2 zeigt, wie der transformierte längste Wege Graph in beiden Zweigen für Variable $y_{r_2 o_2}$ aussehen würde.

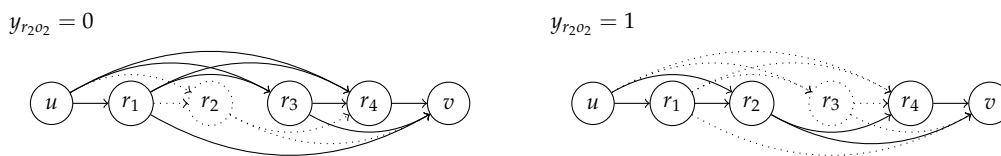


Abbildung 4.2: Längste Wege Graphen für das Pricing-Problem für Gleis o_2 im Beispiel aus Abbildung 1.2 nach branchen auf $y_{r_2 o_2}$. Die gepunkteten Knoten und Kanten sind von dem Pricing-Problem entfernt worden.

4.4 Ergebnisse

Die Problemformulierungen der Paper [1], [2] und [3], die während dieser Arbeit in den Kapiteln 3.2 und 4.1 präsentiert wurden, werden an einem historischen Datensatz des Rangierbahnhofs Hallsberg in Schweden ausgewertet.

Der Datensatz ist für ankommende und abfahrende Züge und Güterwagen des Rangierbahnhofs Hallsberg im Zentrum von Schweden und umfasst einen Zeitraum von fünf Monaten zwischen Dezember 2010 und Mai 2011. Der Ankunftshof von Hallsberg hat 8 Gleise mit den Längen zwischen 595 und 693 Meter, zwei parallele Ablaufberge (von denen nur einer in Betrieb ist), 32 Rangiergleise von unterschiedlicher

Länge (zwischen 374 und 760 Meter) und 12 Gleise mit einer Länge von 562 bis 886 Meter auf dem Abfahrtshof. Obwohl mehrere andere Gleise auf dem Hof existieren, werden sie normalerweise nicht zum Rangieren benutzt und daher berücksichtigen wir diese in unserem Modell nicht. Der Aufbau von Hallsberg wird in Abbildung 4.3 gezeigt.

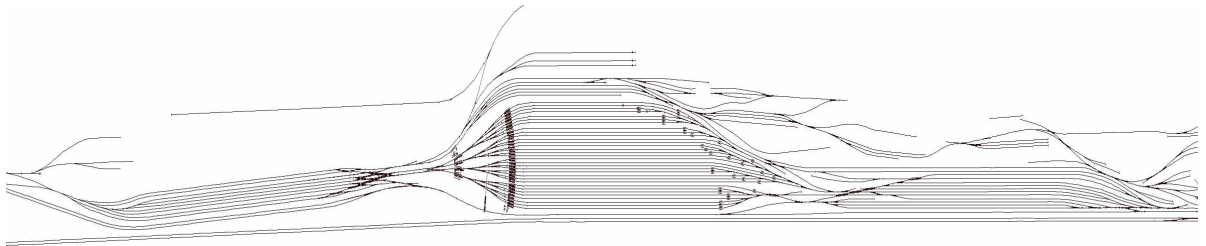


Abbildung 4.3: Aufbau des Rangierbahnhofs Hallsberg in Schweden. Der Ankunftshof ist auf der linken Seite gefolgt von dem Ablaufberg, dem Weichensystem, den Rangiergleisen, und schließlich dem Abfahrtshof auf der rechten Seite. Das Bild ist skaliert, um Details hervorzuheben.

Das Rangierproblem ist durch die Roll-In Zeiten der Inbound-Züge, die Abfahrtszeiten der Outbound-Züge aus der Rangiergruppe, sowie die Zeitpunkte der Pull-Outs gegeben. Die Daten, die uns bereitgestellt wurden, enthalten diese Informationen nicht. Diese bestehen nur aus den Ankunftszeiten der Züge im Ankunftshof und den Abfahrtszeiten der Züge von dem Abfahrtshof. Daher werden in einem Preprocessingschritt die fehlenden Daten berechnet (siehe [2] für weitere Details).

Der resultierende Datensatz aus den fünf Monaten besteht aus den Ankunftszeiten für 3653 Outbound-Züge und 18366 Güterwagengruppen. Da in der Praxis die jeweils aktuelle Betriebsplanungszeitraum für einen Zeitraum von ein paar Tagen aufgestellt wird, werden die resultierenden Daten in separate Planungsprobleme aufgeteilt, die jeweils alle Güterwagengruppen und die entsprechenden Züge enthalten, welche auf dem Rangierbahnhof zu den gewählten Intervallen behandelt werden. Wir planen für Zeiträume von zwei bis fünf Tagen und nehmen an, dass der Rangierbahnhof zu Beginn eines jeden Zeithorizonts leer ist. Insgesamt erhalten wir 192 Problem instanzen, die ausgewertet werden können.

Die resultierenden Problem instanzen optimieren wir mit

- der Heuristik aus 2.1, auf die wir zusätzlich die Verbesserungsheuristik anwenden (Heuristik ++),
- dem direkten Modell aus Kapitel 3.2 (DF),

- dem neuen Spaltengenerierungsansatz von Abschnitt 4.1 (EF).

Im Folgenden werden die Ergebnisse des Swedish Institute of Compute Science präsentiert.

Tabelle 4.1: Ergebnisse für verschiedene Planungshorizonte und Lösungsmethoden. Für jeden Planungshorizont sind die Anzahl der Instanzen in der Datei und die durchschnittliche Problemgröße für jede Instanz dargestellt. Für die unterschiedlichen Lösungsansätze zeigt die Tabelle die durchschnittliche Zahl der extra Roll-Ins, die durchschnittliche Ausführungszeit zusammen mit der Anzahl der Instanzen, für die optimale, zulässige und keine zulässige Lösungen gefunden wurden. Die durchschnittliche Optimalitätsgap wird für Instanzen, für die eine zulässige Lösung mit einer von null verschiedenen unteren Schranke gefunden wurde, angegeben. Nur zulässige Instanzen sind in den Berechnungen des durchschnittlichen extra Roll-Ins enthalten.

		<i>Planungshorizont (Tage)</i>			
		2	3	4	5
<i>Anzahl der Instanzen</i>		75	50	37	30
<i>Durchschn. Anzahl der Züge</i>		48.7	73.0	97.7	121.7
<i>Durchschn. Anzahl der Gruppen</i>		244.9	367.3	492.0	612.2
Heuristik++	Durchsch. extra Roll-Ins	8.3	16.1	26.0	31.6
	Durchsch. Zeit (s)	0.0	0.1	0.1	0.2
Inst. Klassen	Zulässige Lösung gefunden	73	47	34	27
	Keine zulässige Lösung gefunden	2	3	3	3
DF	Durchsch. extra Roll-Ins	10.1	18.8	29.8	25.2
	Durchschn. Zeit (s)	360.6	530.9	684.1	722.4
Inst. Klassen	Optimalität bewiesen	50	27	14	12
	Zulässige Lösung, LB>0	3(6.0)	2(16.2)	3(18.9)	1(7.4)
	Zulässige Lösung, LB=0	22	21	20	14
	Keine zulässige Lösung gefunden	0	0	0	3
EF	Durchsch. extra Roll-Ins	10.1	18.2	27.6	39.0
	Durchsch. Zeit (s)	2.0	17.5	76.4	168.2
Inst. Klassen	Optimalität bewiesen	75	50	37	30
	Zulässige Lösung, LB>0	0(0.0)	0(0.0)	0(0.0)	0(0.0)
	Zulässige Lösung, LB=0	0	0	0	0
	Keine zulässige Lösung gefunden	0	0	0	0

Falls eine heuristische Lösung gefunden wurde, wurde diese auch als Startlösung für die (DF) und (EF) verwendet. CPLEX 12.4.0.0 wurde zum Lösen der (DF) im deterministischen parallelen Modus mit bis zu 8 threads verwendet. (EF) benutzt SCIP

2.1.0 als ein Branch-and-Price framework mit der gleichen CPLEX Version als LP-Solver. Die Experimente wurden auf einer Linux-Workstation mit zwei Intel Core i7 – 2600 quad-core Prozessoren, mit jeweils 3.4 GHz und 16 GB RAM durchgeführt, auf welchem openSuse 12.1 läuft. Ein Zeitlimit von 20 Minuten wurde für jede Problem Instanz festgelegt, nach denen dann die beste gefundene ganzzahlige Lösung zurückgegeben wurde.

Die oben erwähnte Gap wird als $|p - d|/p$ berechnet, wobei p die primale Schranke (der Wert der besten gefundenen ganzzahligen Lösung) und d die duale Schranke ist. Zu beachten ist, dass, wenn ein Ansatz keine zulässige Lösung für eine Instanz findet, diese Instanz von der Berechnung des Durchschnitts der extra Roll-Ins ausgeschlossen wird. Da diese Instanzen oft viel Verkehr aufweisen, reduziert dies die durchschnittliche Zahl der zusätzlichen Roll-Ins für diese Ansätze. Daher ist die extra Roll-In Anzahl der Heuristik im Durchschnitt niedriger als bei dem Optimierungsverfahren. Ebenso ist dies der Grund, warum (DF) eine niedrigere extra Roll-In Durchschnittszahl hat als die (EF) für den fünf Tage Planungshorizont. Wir können sehen, dass der Spaltengenerierungsansatz für alle Problem Instanzen Optimalität beweist, hingegen der direkte Ansatz nur bei 54% der Instanzen dies erreicht. Dies deutet auf eine starke LP-Relaxation des Column-Generations hin. Im Gegensatz terminieren bei der direkten Formulierung 90% der zulässigen Instanzen, die nicht optimal gelöst wurden, mit einer trivialen unteren Schranke von Null. Für (EF) dauerte es für keine Instanz länger als 13 Minuten diese optimal zu lösen, während (DF) das Zeitlimit für etwa 45% der Fälle erreicht (siehe für Details Kapitel 7).

4.5 Duale Interpretation

In diesem Abschnitt beschäftigen wir uns mit der dualen Sichtweise des Spaltengenerierungsansatzes des Papers [3] und stellen Techniken zur Beschleunigung des Lösungsverfahrens dar.

Sei z_{RMP}^* der optimale Funktionswert des (RMP)s. Wenn eine obere Schranke $\kappa > \sum_{j \in J} \lambda_j$ für die optimale Lösung des (MP)s z_{MP}^* existiert, haben wir nicht nur eine obere Schranke z_{RMP}^* für z_{MP}^* in jeder Iteration, sondern auch eine untere Schranke gefunden. Wir können z_{RMP}^* nicht um mehr als das κ -fache der kleinsten, reduzierten Kosten \bar{c}^* reduzieren (vgl. [12]):

$$z_{RMP}^* + \kappa \bar{c}^* \leq z_{MP}^* \leq z_{RMP}^*.$$

So können wir die Qualität der Lösung jederzeit überprüfen. Im Optimum gilt $\bar{c}^* = 0$ und somit $z_{RMP}^* = z_{MP}^*$.

Da wir in unserem Problem das Pricing für jedes Gleis einzeln lösen und wegen (4.13) $\kappa = 1$ gilt, folgt bei uns folgende Beziehung:

$$z_{RMP}^* + \sum_{o \in O} \bar{c}_o^* \leq z_{MP}^* \leq z_{RMP}^*.$$

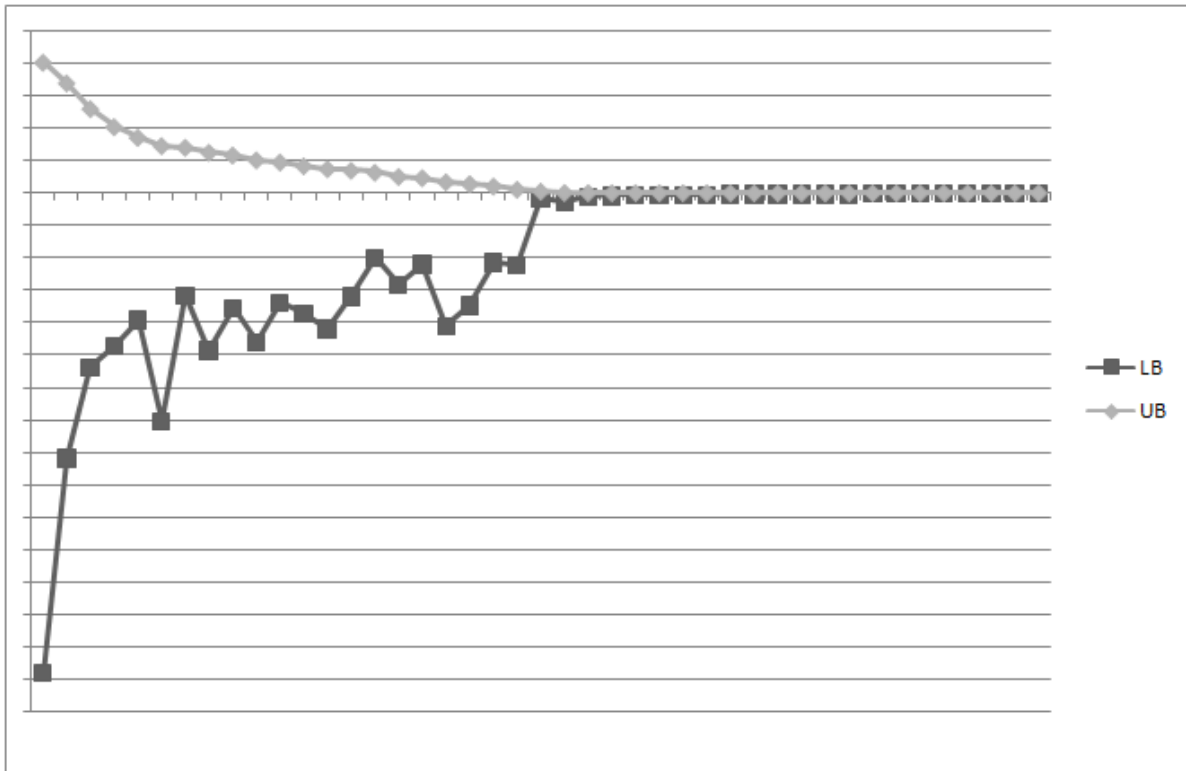


Abbildung 4.4: Entwicklung der oberen und unteren Schranken.

Abbildung 4.4 zeigt die Entwicklungen der oberen ($UB = z_{RMP}^*$) und unteren ($LB = z_{RMP}^* + \sum_{o \in O} \bar{c}_o^*$) Schranken für das Rangierproblem im Wurzelknoten. Dabei werden als Instanz ein paar Tage des fünfmonatigen Datensatzes betrachtet. Während die obere Schranke monoton fällt, weist die untere Schranke keine Monotonie auf (**Yo-Yo Effekt**). Dennoch können wir diese Grenzen benutzen, um die Qualität der aktuellen Lösung durch Berechnung der Optimalitätsgap zu bewerten und haben die Möglichkeit den Prozess zu stoppen, wenn eine voreingestellte Qualität erreicht wird. Im Folgenden beschreiben wir, wie wir die Schranken einsetzen können, um den Lösungsprozess zu beschleunigen.

Early Branching:

Zunächst ist es möglich den Pricing-Prozess zu beenden, falls $|(UB - LB)/LB| < \epsilon$

für $\epsilon < 1$ gilt. D.h wir brechen das Lösen des aktuellen Knotens ab und beginnen frühzeitig mit dem Branching. Dies hat zur Folge, dass wir nicht wissen, ob weitere Variablen existieren, die das (MP) verbessern könnten. Wir lösen das Master-Problem nicht optimal und daher ist der optimale Zielfunktionswert des (RMP) keine gültige untere Schranke für den optimalen Wert des Problems. Da wir in jedem Pricing-Schritt das (RMP) und das Subproblem optimal lösen, erhalten wir immer die gültige untere Schranke $LB = z_{RMP}^* + \sum_{o \in O} \bar{c}_o^*$. Somit können wir die untere Schranke dieses Knotens auf diesen Wert aktualisieren. Das Farkas Pricing wird wie gewohnt durchgeführt und nicht abgebrochen, da wir sonst keine zulässige Lösung für das aktuelle Master-Problem hätten und auch nicht wüssten, ob solch eine Lösung existiert. Im Folgenden fassen wir diese Technik zusammen.

Algorithmus: Early Branching

Input: Parameter ϵ , $0 < \epsilon < 1$.

- Löse das (RMP). Erhalte die untere Schranke LB und die obere Schranke UB ;
- Falls $|(UB - LB)/LB| < \epsilon$ gilt, stoppe die Pricing-Prozedur und branche;
- Wiederhole bis keine Variablen mit negativen reduzierten Kosten existieren;

Early Branching ergibt im Allgemeinen eine langsamere Erhöhung der dualen Schranke. Dies hat zur Folge, dass der Branch-and-Price Baum schneller bearbeitet wird und die Knoten, die im Baum tiefer gelegen sind, früher untersucht werden. Da diese Knoten den Problemen entsprechen, für die es wahrscheinlicher ist, dass diese eine ganzzahlige Lösung besitzen (da z.B. einige Variablen auf ganzzahlige Werte fixiert sind), kann dies zu einer zulässigen Lösung führen, die im Lösungsprozess früher gefunden wird. Doch man sollte sich vorab folgendes klarmachen: Falsche, also schlechte bzw. unnötige Verzweigungen können die Größe des Suchbaums vervielfachen. Verzweigt man nämlich frühzeitig auf einer aktuell noch gebrochenen Entscheidungsvariablen, welche jedoch in einer exakten Lösung des entsprechenden Master-Problems ganzzahlig ist, so resultiert daraus ein überflüssiger Kind-Knoten. Eine derartige Entscheidung kann auf tieferen Ebenen des Suchbaums erneut getätigt werden. Die Entscheidung zu verzweigen sollte also erst dann getroffen werden, wenn man mit hoher Wahrscheinlichkeit davon ausgehen kann, dass die zugrunde liegende Verzweigungsvariable auch tatsächlich gebrochen ist.

Tabelle 4.2: Ergebnisse für verschiedene Planungshorizonte unter Anwendung von Early Branching. Für jeden Planungshorizont sind die Anzahl der Instanzen in der Datei und die durchschnittliche Problemgröße für jede Instanz dargestellt. Die Tabelle zeigt die durchschnittliche Zahl der extra Roll-Ins und die durchschnittliche Ausführungszeit.

		<i>Planungshorizont (Tage)</i>			
		2	3	4	5
	<i>Anzahl der Instanzen</i>	75	50	37	30
	<i>Durchschn. Anzahl der Züge</i>	48.7	73.0	97.7	121.7
	<i>Durchschn. Anzahl der Gruppen</i>	244.9	367.3	492.0	612.2
EF	Durchsch. extra Roll-Ins	10.1	18.2	27.6	39.0
	Durchsch. Zeit (s)	1.4	12.3	65.5	142.9
$\epsilon = 0.00$	Optimalität bewiesen	75	50	37	30
EF + Early Branching	Durchsch. extra Roll-Ins	10.1	18.2	29.0	40.4
	Durchsch. Zeit (s)	1.5	30.0	172.5	411.0
$\epsilon = 0.01$	Optimalität bewiesen	75	50	35	22
EF + Early Branching	Durchsch. extra Roll-Ins	10.1	18.2	29.3	39.0
	Durchsch. Zeit (s)	1.4	34.6	181.5	425.0
$\epsilon = 0.03$	Optimalität bewiesen	75	50	35	22
EF + Early Branching	Durchsch. extra Roll-Ins	10.1	18.2	29.3	24.0
	Durchsch. Zeit (s)	1.5	45.7	159.3	424.0
$\epsilon = 0.05$	Optimalität bewiesen	75	50	35	23

Der einzige und entscheidende Parameter ϵ wird mit den Werten 0.01 (= 1%), 0.03 (= 3%) und 0.05 (= 5%) für das Rangierproblem variiert. Bei der Durchführung der Test-Rechnungen werden dieselben Instanzen wie bei der Untersuchung in Abschnitt 4.4 verwendet. Es stehen wieder maximal 20 Minuten (1200s) pro Instanz zur Verfügung. Wenn eine heuristische Lösung gefunden wird, wird diese als Startlösung verwendet. Falls für eine Instanz keine zulässige Lösung gefunden wird, wird diese von der Berechnung der durchschnittlichen extra Roll-In Anzahl ausgeschlossen. Die Tabelle 4.2 fasst die Ergebnisse der Test-Rechnungen zusammen.

Wir können sehen, dass durch Early Branching keine Verbesserung für das Rangierproblem erzielt werden kann. Die durchschnittlichen Rechenzeiten sind für kleine Werte von ϵ und ohne Early Branching am geringsten. Bei größeren Werten steigt die Rechenzeit und die Anzahl der optimal gelösten Instanzen sinkt (vgl. Kapitel 7).

Wir wählen die 9. Instanz des 3 Tage Planungshorizonts als Test-Instanz, um anhand dieser die Problematik des Early Branchings beispielhaft zu erklären. Ohne

Early Branching besteht der Branch-and-Price-Suchbaum aus 51 Knoten, welche in einer Zeit von 53.7s abgearbeitet werden. Bei Steigerung von ϵ (von 1%, 3% bis 5%) wachsen die Zahl der Knoten (858, 1018 und 2623) und die Rechenzeit von 444.2s bis zu 936.7s. Die mehrfachen, zu früh getroffenen Entscheidungen, auf eigentlich ganzzahlige Werte zu verzweigen, führen zu einem großen Suchbaum und steigenden Rechenzeiten.

Early Branching trägt bei dem Rangierproblem nicht zur Beschleunigung des Verfahrens bei. Es können nur marginale Unterschiede in den Rechenzeiten festgestellt werden. Zum einen existieren Instanzen, bei denen die Technik Early Branching deutlich kürzere Rechenzeiten bewirkt. Zum anderen existieren jedoch auch Instanzen, die erheblich größere Rechenzeiten aufweisen. Daher kompensieren sich die Zeiten, so dass letztendlich diese Technik keine Verbesserung bewirkt. Zu große Werte von ϵ führen sogar teilweise zu extrem vielen falschen Verzweigungsentscheidungen, aus denen insgesamt weniger optimal gelöste Instanzen resultieren.

Early Termination:

Early Termination ist eine abgeschwächte Version von Early Branching. Wenn bekannt ist, dass der Zielfunktionswert jeder zulässigen Lösung ganzzahlig ist, stoppen wir den Pricing-Prozess, falls weitere Verbesserungen der dualen Schranke des Knotens keine Vorteile bzgl. des Branchings liefern. Im Folgenden beschreiben wir, wie dies festgestellt werden kann.

Sei z_{RMP}^* der optimale Zielfunktionswert des (RMP)s und LB die aktuelle duale Schranke des aktiven Knotens. Durch das optimale Lösen des Master-Problems, erhalten wir eine untere Schranke z_{MP}^* für das Problem und insgesamt gilt die uns bekannte Beziehung

$$LB \leq z_{MP}^* \leq z_{RMP}^*.$$

Falls $\lceil LB \rceil \geq z_{RMP}^*$ gilt, dann folgt $\lceil LB \rceil = \lceil z_{MP}^* \rceil$. Dies bedeutet, dass die beste zulässige Lösung für das Problem einen Zielfunktionswert von mindestens $\lceil LB \rceil$ hat, so dass die untere Schranke des Knotens auf $\lceil LB \rceil$ gesetzt werden kann. Das optimale Lösen des Master-Problems kann nicht zu einer besseren dualen Schranke führen und daher brechen wir den Lösungsprozess des (MP)s ab. Falls eine Lösung mit Wert $\lceil LB \rceil$ bzw. besser gefunden wird, kann dieser Knoten terminiert werden.

Algorithmus: Early Termination

- Löse das (RMP). Erhalte die untere Schranke LB und die obere Schranke z_{RMP}^* ;
 - Falls $\lceil LB \rceil \geq z_{RMP}^*$ gilt, stoppe die Pricing-Prozedur;
 - Falls eine Lösung mit Wert $\lceil LB \rceil$ bzw. besser gefunden wird, terminiere diesen Knoten;
 - Wiederhole bis keine Variablen mit negativen reduzierten Kosten existieren;
-

Das Spaltengenerierungsverfahren besitzt die negative Eigenschaft **Tailing-Off**. Dies bedeutet, dass es nur kleine Fortschritte pro Iteration gibt, je näher wir uns auf das Optimum bewegen. Dieses Phänomen tritt insbesondere bei großen und entarteten Problemen auf. Denn, wenn das (RMP) entartet ist, existieren alternative duale optimale Lösungen und das Subproblem muss möglicherweise mehrere Spalten (mehrere Iterationen) erzeugen, um eine echte Verbesserung der primalen Lösung des (RMP)s zu erreichen. Durch Verwendung von Early Termination kann dieser Effekt reduziert werden.

Tabelle 4.3: Ergebnisse für verschiedene Planungshorizonte unter Anwendung von Early Termination. Für jeden Planungshorizont sind die Anzahl der Instanzen in der Datei und die durchschnittliche Problemgröße für jede Instanz dargestellt. Die Tabelle zeigt die durchschnittliche Anzahl der extra Roll-Ins und die durchschnittliche Ausführungszeit.

		<i>Planungshorizont (Tage)</i>			
		2	3	4	5
	<i>Anzahl der Instanzen</i>	75	50	37	30
	<i>Durchschn. Anzahl der Züge</i>	48.7	73.0	97.7	121.7
	<i>Durchschn. Anzahl der Gruppen</i>	244.9	367.3	492.0	612.2
EF +	Durchsch. extra Roll-Ins	10.1	18.2	27.6	39.0
	Durchsch. Zeit (s)	2.0	28.7	70.1	192.1
Early Termination	Optimalität bewiesen	75	50	37	30
EF	Durchsch. extra Roll-Ins	10.1	18.2	27.6	39.0
	Durchsch. Zeit (s)	2.0	17.5	76.4	168.2
	Optimalität bewiesen	75	50	37	30

4 Ein Spaltengenerierungsansatz für das Rangierproblem

Da das Rangierproblem nur ganzzahlige Variablen mit ganzzahligen Zielkoeffizienten besitzt, ist der Zielfunktionswert immer ganzzahlig. In der Tabelle 4.3 sind die Ergebnisse zusammengefasst, die durch Anwendung von Early Termination auf das Rangierproblem entstanden sind.

Wir sehen, dass Early Termination im Durchschnitt keine Beschleunigung bewirkt. In der nachfolgenden Tabelle wird dargestellt, wie oft Early Termination eintritt und ob diese Technik eine positive oder negative Beschleunigung erzeugt. Dabei bedeutet eine negative Zahl eine Beschleunigung um diesen Betrag und eine positive Zahl bezeichnet, dass die Laufzeit um diesen Betrag gestiegen ist.

Tabelle 4.4: Ergebnisse unter Anwendung von Early Termination. Enthalten sind die Anzahl positiver und negativer Beschleunigungen, die durch Early Termination hervorgerufen werden und die daraus resultierenden verkürzten bzw. verlängerten Laufzeiten.

	<i>Beschleunigung</i>		<i>keine Beschleunigung</i>		<i>ges. Beschleunigung</i>
	Anzahl (#)	Zeit (s)	Anzahl (#)	Zeit (s)	Zeit (s)
2 Tage	4	-11.7	6	+14.4	+2.7
3 Tage	6	-53.2	9	+619.6	+566.4
4 Tage	10	-895.6	7	+661.0	-234.6
5 Tage	9	-1128.5	9	1845.6	+717.1

Da unter Anwendung von Early Termination die Erhöhung der dualen Schranke eines Knotens nicht exakt berechnet wird, steigt die durchschnittliche Anzahl der Knoten. Dies führt bei unserem Problem dazu, dass Early Termination keine Beschleunigung bewirkt, obwohl bei einzelnen Instanzen eine erhebliche Verkürzung der Laufzeiten zu beobachten ist. Diese Beschleunigungen weisen jedoch keine Regelmäßigkeit auf. Ob die Technik einen positiven Effekt hat oder nicht, hängt weder von der Anzahl der Züge noch von den Gruppen in einer Instanz ab (siehe Anhang).

Eine weitere Ursache für die langsame Konvergenz von Column-Generierung ist das instabile Verhalten der dualen Variablen. Eine duale Lösung kann weit von der vorherigen entfernt sein (**Bang-Bang Effekt**). Auch wenn eine gute Annäherung der dualen Variablen in einer Iteration erhalten wird, kann die duale Lösung bei der darauffolgenden Iteration beliebig weit davon entfernt sein. Mit anderen Worten gesagt, ist der Spaltengenerierungsansatz nicht in der Lage die Tatsache auszunutzen, dass bereits eine gute duale Lösung erreicht wurde, um die nachfolgenden Berechnungen zu beschleunigen. Dies ist als **Instabilität** des Ansatzes bekannt, welches die

Hauptursache für die langsame Konvergenzrate ist. Zusammengefasst ergeben sich folgende mögliche Nachteile des Spaltengenerierungsverfahrens:

- Langsame Konvergenz (Tailing-Off Effekt).
- Die ersten Iterationen erzeugen irrelevante Spalten und duale Schranken (Heading-In Effekt).
- Entartung im Primalen und somit mehrere optimal Lösungen im Dualen → Ergebniss des (RMP)s bleibt für eine Weile konstant (Plateau Effekt).
- Instabilität der dualen Lösungen, die von einem extremen Wert zum anderen springen (Bang-Bang Effekt).
- Die duale Schranke konvergiert nicht monoton (Yo-Yo Effekt).

Die Abbildung 4.5 soll die oben eingeführten Effekte verdeutlichen.

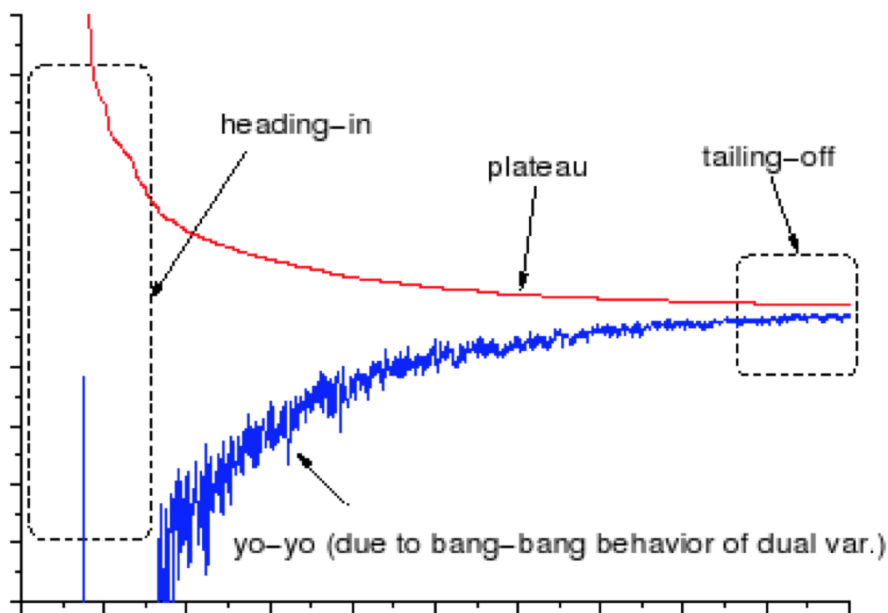


Abbildung 4.5: Bildliche Darstellung der oben eingeführten Effekte

Die **Stabilisierung** der dualen Variablen versucht den Bang-Bang Effekt zu reduzieren. Es sei darauf hingewiesen, dass in dem Fall, in dem die Stabilisierung erfolgreich ist, unabhängig davon welche Methode verwendet wird, häufig eine Verringerung der Spaltengenerierungsiterationen beobachtet wird. Doch auf der anderen Seite wird das Lösen der Pricing-Probleme schwieriger.

Im Folgenden stellen wir drei Stabilisierungsmöglichkeiten vor. Die ersten zwei Ansätze modifizieren das (RMP), während der letzte mit Hilfe eines Parameters versucht den Bang-Bang Effekt zu reduzieren.

1. Polyedrische Strafterme / Flexible Box:

Eine Möglichkeit, dem instabilen Verhalten der dualen Variablen entgegenzuwirken, ist, das Master-Problem durch Hinzufügen von beschränkten Überschuss- und Schlupfvariablen zu "stören". Resultierend erhalten wir das modifizierte Master-Problem (\widehat{MP})

(\widehat{MP})

$$\begin{aligned} z_{\widehat{MP}}^* &= \min \mathbf{c}^T \boldsymbol{\lambda} - \delta_-^T \mathbf{y}_- + \delta_+^T \mathbf{y}_+ \\ \text{s.t.} \quad & \mathbf{A} \boldsymbol{\lambda} - \mathbf{y}_- + \mathbf{y}_+ = \mathbf{b} \\ & \mathbf{y}_- \leq \boldsymbol{\epsilon}_- \\ & \mathbf{y}_+ \leq \boldsymbol{\epsilon}_+ \\ & \boldsymbol{\lambda}, \mathbf{y}_-, \mathbf{y}_+ \geq \mathbf{0}, \end{aligned}$$

mit dem zugehörigen dualen Problem (\widehat{DMP})

(\widehat{DMP})

$$z_{\widehat{DMP}}^* = \max \boldsymbol{\pi}^T \mathbf{b} - \boldsymbol{\epsilon}_-^T \mathbf{w}_- - \boldsymbol{\epsilon}_+^T \mathbf{w}_+ \tag{4.21}$$

$$\text{s.t.} \quad \boldsymbol{\pi}^T \mathbf{A} \leq \mathbf{c} \tag{4.22}$$

$$-\boldsymbol{\pi} - \mathbf{w}_- \leq -\boldsymbol{\delta}_- \tag{4.23}$$

$$\boldsymbol{\pi} - \mathbf{w}_+ \leq \boldsymbol{\delta}_+ \tag{4.24}$$

$$\mathbf{w}_-, \mathbf{w}_+ \geq \mathbf{0} \tag{4.25}$$

$$\boldsymbol{\pi} \in \mathbb{R}^n. \tag{4.26}$$

In dem primalen Problem (\widehat{MP}) sind $\mathbf{y}_-, \mathbf{y}_+$ Vektoren von Schlupf- und Überschussvariablen mit oberen Schranken $\boldsymbol{\epsilon}_-$ bzw. $\boldsymbol{\epsilon}_+$. Diese Variablen werden in der Zielfunktion durch Vektoren $\boldsymbol{\delta}_-$ bzw. $\boldsymbol{\delta}_+$ bestraft. Die Nebenbedingungen (4.23) und (4.24) von (\widehat{DMP}) können wir zu

$$\boldsymbol{\delta}_- - \mathbf{w}_- \leq \boldsymbol{\pi} \leq \boldsymbol{\delta}_+ + \mathbf{w}_+$$

umschreiben. Dies kann so interpretiert werden, dass bei dem Verlassen der Box $[\boldsymbol{\delta}_-, \boldsymbol{\delta}_+]$ die dualen Variablen $\boldsymbol{\pi}$ entsprechend dem Abstand zur Box mit $\boldsymbol{\epsilon}_-, \boldsymbol{\epsilon}_+$ pro Einheit bestraft werden. Denn, wenn die dualen Variablen die Box verlassen, nehmen

\mathbf{w}_- und \mathbf{w}_+ positive Werte an. Auch wenn in dem obigen Modell das gewählte Intervall weit von einer optimalen dualen Lösung von (MP) entfernt ist, können wir, indem wir die Parameter ϵ_- und ϵ_+ klein genug wählen, ermöglichen, dass (\widehat{MP}) optimale primale und duale Lösungen besitzt, die nahe an den optimalen Lösungen von (MP) sind. Wir bezeichnen mit \mathbf{x}^* , $\boldsymbol{\pi}^*$, $(\widehat{\mathbf{x}}^*, \mathbf{y}_-^*, \mathbf{y}_+^*)$ und $(\widehat{\boldsymbol{\pi}}^*, \mathbf{w}_-^*, \mathbf{w}_+^*)$ die optimalen Lösungen von (MP), (DMP), (\widehat{MP}) und (\widehat{DMP}) . Dann gilt $(MP) = (\widehat{MP})$ (d.h. $\mathbf{y}_-^* = \mathbf{y}_+^* = 0$), wenn eine der folgenden Bedingungen erfüllt ist

1. $\epsilon_- = \epsilon_+ = 0$,
2. $\delta_- < \widehat{\boldsymbol{\pi}}^* < \delta_+$.

Die 2. Bedingung folgt aus dem Satz vom komplementären Schlupf.

(4.2) Satz

Sei $\boldsymbol{\lambda}$ eine primal zulässige Lösung von (MP) und $\boldsymbol{\pi}$ eine dual zulässige Lösung von (DMP). $\boldsymbol{\lambda}$ und $\boldsymbol{\pi}$ sind genau dann optimal, wenn

- $\boldsymbol{\lambda} = \mathbf{0} \vee \boldsymbol{\pi}^T \mathbf{A} = \mathbf{c}$,
- $\boldsymbol{\pi} = \mathbf{0} \vee \mathbf{A} \boldsymbol{\lambda} = \mathbf{b}$.

ist. Zusätzlich gilt

$$z_{\widehat{MP}}^* \leq (\widehat{\boldsymbol{\pi}}^*)^T \mathbf{b} \leq z_{MP}^*.$$

Diese Ungleichung zeigt, dass (\widehat{MP}) eine Relaxation von (MP) ist und dass $(\widehat{\boldsymbol{\pi}}^*)^T \mathbf{b}$ eine bessere untere Schranke für z_{MP}^* liefert als $z_{\widehat{MP}}^*$.

Zur Stabilisierung des Column-Generation-Verfahrens wird (\widehat{MP}) anstelle von (MP) gelöst, wobei im Laufe des Verfahrens durch Anpassen von ϵ_- , ϵ_+ , δ_- und δ_+ mindestens eine der oberen Bedingungen erfüllt wird.

2. Boxstep Methode:

Die Boxstep Methode ist ein Spezialfall der oben vorgestellten Stabilisierung. In der Boxstep Methode gilt $\epsilon_- = \epsilon_+ = \infty$ und somit können die dualen Variablen nicht außerhalb der festgelegten Box liegen. Mit der Festlegung von Unter- und Obergrenzen werden die dualen Variablen gezwungen in einer Box um die bisherige duale Lösung zu liegen. Diese Lösung wird als das **Zentrum** der Stabilisierung bezeichnet. Das so beschränkte (RMP) wird neu optimiert. Wenn die neue optimale Lösung den Rand dieser Box erreicht, haben wir eine Richtung, in die die Box verschoben werden sollte. Andernfalls wird das Optimum im Inneren der Box erreicht.

4 Ein Spaltengenerierungsansatz für das Rangierproblem

Das durch die Boxstep Methode modifizierte Master-Problem (\overline{MP}) sieht wie folgt aus

(\overline{MP})

$$\begin{aligned} z_{\overline{MP}}^* &= \min \mathbf{c}^T \boldsymbol{\lambda} - \delta_-^T \mathbf{y}_- + \delta_+^T \mathbf{y}_+ \\ \text{s.t.} \quad & \mathbf{A} \boldsymbol{\lambda} - \mathbf{y}_- + \mathbf{y}_+ = \mathbf{b} \\ & \boldsymbol{\lambda}, \mathbf{y}_-, \mathbf{y}_+ \geq \mathbf{0}. \end{aligned}$$

Da wir im Dualen zu jeder Variable eine Restriktion erhalten, sieht das zugehörige duale Problem (\overline{DMP}) folgendermaßen aus

(\overline{DMP})

$$z_{\overline{DMP}}^* = \max \boldsymbol{\pi}^T \mathbf{b} \tag{4.27}$$

$$\text{s.t.} \quad \boldsymbol{\pi}^T \mathbf{A} \leq \mathbf{c} \tag{4.28}$$

$$-\boldsymbol{\pi} \leq -\delta_- \tag{4.29}$$

$$\boldsymbol{\pi} \leq \delta_+ \tag{4.30}$$

$$\boldsymbol{\pi} \in \mathbb{R}^n. \tag{4.31}$$

Weil die Nebenbedingungen (4.29) und (4.30) sich als

$$\delta_- \leq \boldsymbol{\pi} \leq \delta_+$$

schreiben lassen, erhalten wir die gewünschte Box um die dualen Variablen.

Eine Möglichkeit die Boxgrenzen zu bestimmen, geschieht mit Hilfe der Graphik 4.5. Wir können beispielsweise Iteration i wählen, falls der Sprung von $i - 1$ auf i "gut" war. Verbesserungen (Anstiege) der LB-Werte können als Zeichen genommen werden, dass π_i näher an einer optimalen Lösung von (DMP) liegt. Als Schranken für die dualen Variablen haben wir dann $\tau + / -$ die dualen Variablen der Iteration i , wobei τ ein beliebiger positiver Vektor ist, der die Länge des Intervalls angibt. So gelten $\delta_- = \pi_i - \tau$ und $\delta_+ = \pi_i + \tau$.

Alternativ können die optimalen Langrangemultiplikatoren verwendet werden, um die optimalen Boxgrenzen festzulegen (siehe: [11]).

Da die beiden oben eingeführten Methoden das (RMP) modifizieren, ergeben sich zwei wesentliche Nachteile:

1. Es müssen viele Parameter bestimmt werden und entschieden werden, wann das Stabilisierungszentrum aktualisiert wird.

2. Durch "stören" des (MP)s werden zusätzliche Variablen hinzugefügt. Dadurch vergrößert sich das Problem.

3. Smoothing Methode:

Die Smoothing Methode ist ein weiterer Ansatz, der zu Vermeidung von großen Schritten in dem dualen Raum eingesetzt wird und keine Änderung an dem (RMP) benötigt. Zudem ist diese sehr leicht in der Handhabung. Es muss nur ein einziger Parameter gewählt werden und dieser bleibt über den ganzen Spaltengenerierungsprozess hinweg konstant. Denn die Methode betrachtet die Konvexkombination aus der aktuellen dualen Lösung π_{akt} und einer früheren dualen Lösung π_{akt-1} , d.h. das Pricing-Problem wird mit $\pi_{konv} = \alpha\pi_{akt} + (1 - \alpha)\pi_{akt-1}$ für $0 < \alpha \leq 1$ aufgerufen anstatt mit π_{akt} . Sei φ die Variable mit minimalen reduzierten Kosten bzgl. π_{konv} , die während der Pricing-Prozedur gefunden wird. Wenn φ bzgl. π_{akt} negative reduzierte Kosten hat, wird es dem (RMP) in der nächsten Iteration hinzugefügt. Außerdem ist π_{konv} , falls $LB_{konvex} > LB_{akt-1}$ gilt, ein verbessernder dualer Vektor und dieser wird das neue Zentrum der Stabilisierung ($\pi_{akt-1} \leftarrow \pi_{konvex}$). Nachfolgend sehen wir eine algorithmische Beschreibung der Technik.

Algorithmus: Column-Generation mit der Smoothing Methode erweitert

Input: Parameter α , $0 < \alpha \leq 1$.

- Garantiere die Zulässigkeit des (RMP)s;
- $\pi_{akt-1} = 0$;
- Wiederhole
 - Löse das (RMP) und erhalte den Vektor π_{akt} ;
 - $\pi_{konvex} \leftarrow \alpha\pi_{akt} + (1 - \alpha)\pi_{akt-1}$;
 - Löse das Pricing-Problem mit π_{konvex} . Erhalte die untere Schranke LB_{konvex} bzgl. π_{konvex} und die Variable mit negativen reduzierten Kosten φ ;
 - Falls $LB_{konvex} > LB_{akt-1}$, dann $\pi_{akt-1} \leftarrow \pi_{konvex}$;
 - Falls φ bzgl. π_{akt} negative reduzierte Kosten hat, dann füge φ dem (RMP) hinzu;
- Bis $z_{RMP}^* - LB_{akt-1} < \epsilon$

Nun muss noch gezeigt werden, dass dieser Algorithmus nach endlich vielen Iterationen terminiert und eine optimale Lösung des (MP) liefert.

(4.3) Lemma

Falls die Lösung des Pricing-Problems mit dem Vektor π_{konvex} keine Spalten mit negativen reduzierten Kosten bzgl. π_{akt} liefert, dann gilt

$$LB_{konvex} \geq LB_{akt-1} + \alpha(z_{RMP}^* - LB_{akt-1}).$$

Beweis

Siehe [13].

Ein **Misprice** passiert dann, wenn Spalten mit negativen reduzierten Kosten in Bezug auf den Vektor π_{akt} existieren, jedoch das Pricing-Problem, welches mit dem Vektor π_{konv} gelöst wird, keine solche Spalte liefert. Lemma (4.3) besagt, dass ein Misprice keine Zeitverschwendung ist, sondern ganz im Gegenteil die Garantie dafür liefert, dass LB_{konvex} eine neue beste untere Schranke ist, welche LB_{akt-1} verbessert. Genauer gesagt heißt dies, dass ein Misprice die Gap $z_{RMP}^* - LB_{akt-1}$ um mindestens einen Faktor von $1/(1 - \alpha)$ verringern muss (der Algorithmus terminiert, falls diese Gap hinreichend klein ist). Nach einem Misprice gilt $\pi_{akt-1} \leftarrow \pi_{konvex}$, daher wird der nächste Pricing-Schritt, obwohl sich π_{akt} nicht verändert hat (da dem (RMP) keine neue Spalte hinzugefügt wurde), mit einem unterschiedlichen Vektor π_{konvex} durchgeführt.

(4.4) Theorem

Die Smoothing Methode findet nach endlich vielen Iterationen immer eine optimale zulässige Lösung für das (MP).

Beweis

Durch Verwendung von Lemma (4.3). Siehe [13].

Wir weisen an dieser Stelle darauf hin, dass es möglich ist, dass die Lösung des (RMP)s bereits eine optimale Lösung des (MP)s ist, jedoch die Gap $z_{RMP}^* - LB_{akt-1}$ noch positiv ist. Da die Smoothing Methode das Pricing-Problem nicht mit dem Vektor π_{akt} löst, kann es diese Optimalität nicht beweisen. Weil allerdings keine Spalten mit negativen reduzierten Kosten in Bezug auf π_{akt} existieren, sichert Lemma (4.3), dass jeder nachfolgende Pricing-Schritt mit dem Vektor π_{konvex} die Gap um mindestens einen konstanten Faktor reduziert. Daher konvergiert die Gap exponentiell schnell gegen null, was die Optimalität der Lösung des (RMP)s beweist. Als zusätzliche Bemerkung können wir festhalten, dass der Wert von der Konstanten α klein gehalten werden sollte, um den gewünschten Stabilisierungseffekt zu erreichen. Denn kleine Zahlenwerte für α führen dazu, dass der Vektor π_{konvex} sich nicht weit von dem aktuellen Stabilitätszentrum π_{akt-1} entfernt. Wir setzen daher α auf 0.1.

Im Folgenden wenden wir die Smoothing Methode auf das Rangierproblem an. Die Methode wird erst eingesetzt, wenn $|(z_{RMP}^* - LB_{akt-1}) / LB_{akt-1}| < 1.00$ gilt und ausgeschaltet, falls $|(z_{RMP}^* - LB_{akt-1}) / LB_{akt-1}| < 0.01$ ist. Denn auf Grund des Heading-In Effektes macht es keinen Sinn, sofort am Anfang des Spaltengenerierungsprozesses mit der Stabilisierung zu beginnen, sowie es am Ende keine Auswirkung hat, die Konvexkombination der dualen Variablen zu betrachten.

Die Tabelle 4.5 fasst die Resultate der Test-Rechnungen zusammen, wobei $\epsilon = 1$ gilt.

Tabelle 4.5: Ergebnisse für verschiedene Planungshorizonte unter Anwendung der Smoothing Methode. Für jeden Planungshorizont sind die Anzahl der Instanzen in der Datei und die durchschnittliche Problemgröße für jede Instanz dargestellt. Die Tabelle zeigt die durchschnittliche Anzahl der extra Roll-Ins, die durchschnittliche Ausführungszeit und die durchschnittliche Zahl der Iterationen.

		Planungshorizont (Tage)			
		2	3	4	5
<i>Anzahl der Instanzen</i>		75	50	37	30
<i>Durchschn. Anzahl der Züge</i>		48.7	73.0	97.7	121.7
<i>Durchschn. Anzahl der Gruppen</i>		244.9	367.3	492.0	612.2
EF + Smooth. Methode	Durchsch. extra Roll-Ins	10.1	18.2	27.6	28.8
	Durchsch. Zeit (s)	2.3	32.0	109.7	352.6
	Durchsch. Iterationen	14.5	65.5	148.6	344.2
EF	Durchsch. extra Roll-Ins	10.1	18.2	27.6	39.0
	Durchsch. Zeit (s)	2.0	17.5	76.4	168.2
	Durchsch. Iterationen	10.6	55.6	195.3	281.8

Da wie oben erwähnt, in dem Fall, in dem die Stabilisierung erfolgreich ist, häufig eine Verringerung der Spaltengenerierungsiterationen beobachtet wird, stellen wir in der Tabelle 4.6 die jeweilige Anzahl der Iterationen im Vergleich dar. Eine negative Zahl bedeutet dabei, dass die Anzahl der Iterationen um diesen Betrag gesunken ist, wobei eine positive Zahl genau das Gegenteil besagt.

Wir sehen, dass durch die Smoothing Methode ebenfalls keine regelmäßige Beschleunigung erreicht werden kann (siehe Kapitel 7).

Insgesamt können wir festhalten, dass durch die drei Techniken Early Branching, Early Termination und der Smoothing Methode keine Beschleunigung des Lösungsprozesses erreicht werden konnte. Der wesentliche Grund dafür ist, dass im Wurzelknoten bereits der optimale Zielfunktionswert gefunden wird und die weiteren Knoten nur gebraucht werden, um die Ganzzahligkeit zu erreichen.

Tabelle 4.6: Ergebnisse unter Anwendung der Smoothing Methode. Enthalten sind die Anzahl der Iterationen unter Anwendung der Smoothing Methode im Vergleich ohne diese und wie oft die Methode zum Einsatz gekommen ist.

	Anzahl (#)	Iterationen (#)	Anzahl (#)	Iterationen (#)	Iterationen (#)
2 Tage	7	-130	5	+45	-85
3 Tage	9	-492	6	+984	+492
4 Tage	10	-2693	7	+966	-1727
5 Tage	7	-1816	10	+3692	+1876

4.6 Symmetriefreie Formulierung

Symmetrien treten in natürlicher Weise in ganzzahligen Programmen auf. Häufig gibt es keinen Grund spezielle Objekte von anderen zu unterscheiden. Zum Beispiel ist die Linien-Nummer eines Busses für die Qualität des Linienplans unerheblich. Algorithmen, die solche Symmetrien ignorieren, finden häufig die optimale Lösung nicht schnell genug. Symmetrien in ganzzahligen Problemen verlangsamen deren Lösung auf drei Weisen: Der Lösungsraum ist "unnötig groß", die LP-Relaxierungen liefern üblicherweise schlechte duale Schranken und die linearen Programme tendieren dazu schwer lösbar zu sein.

Einer der wenigen systematischen, algorithmischen Ansätze, Symmetrien in ganzzahligen Programmen zu behandeln, stammt von Margot, der Branch-and-Cut Algorithmen (basierend auf Schreier-Sims-Tabellen) entwickelt hat, die nur einen Repräsentanten jedes Lösungsobits erzeugen (siehe: [14] - [17]). Der Orbital Branching-Ansatz, den Linderoth et al. vorgestellt haben, berechnet in jedem Branch-and-Bound-Knoten Symmetriegruppen und wurde zur Lösung von Set-Partitioning- und Set-Covering-Problemen eingesetzt (siehe: [18] und [19]). Einen weiteren Ansatz finden wir in [20]-[23]. Dort werden Methoden entwickelt, die auf der Identifikation von Ungleichungen fußen, die ein Polyeder beschreiben, das mindestens eine Lösung in jedem Orbit enthält.

Auch das Rangierproblem erweist sich als symmetrisch, da die Zugsequenzen auf äquivalenten Gleisen vertauscht werden können. Die meisten Sequenzzuordnungen sind für mehrere Rangiergleise gültig, nämlich für alle Gleise, die länger sind als der längste Zug in der Sequenz. Ein möglicher Ansatz die Symmetrie zu reduzieren ist, die Variablen der (EF) aus Kapitel 4.1 zu aggregieren und eine komplexere

Branching-Regel zu betrachten. In diesem Unterkapitel stellen wir die Theorie dieses Vorgehens dar. Dabei wird das Hall Theorem eine wichtige Rolle spielen.

(4.5) Theorem

Sei $G = (V_1 \cup V_2, E)$ ein bipartiter Graph. Es existiert genau dann ein Matching $|M| = |V_1|$, falls für alle $S \subseteq V_1$ die Hall Bedingung

$$|\Gamma(S)| \geq |S|$$

erfüllt ist, wobei wir mit $\Gamma(S)$ die Nachbarschaft der Knotenmenge S bezeichnen. Wenn $|M| = |V_1| = |V_2|$ gilt, besitzt der Graph sogar ein perfektes Matching.

Das erweiterte Programm aus Kapitel 4.1 ist auf Grund des expliziten Matchings der Sequenzen auf die Gleise symmetrisch. Wir möchten die Tatsache, dass die meisten Zugzuordnungen für mehrere Rangiergleise gültig sind, in unserem Modell berücksichtigen. Daher entscheiden wir zuerst welche Sequenzen wir benutzen und verschieben die Entscheidung, welche Sequenz welchem Gleis zugeordnet wird. Wir müssen nur sicherstellen, dass die Sequenzen, die wir wählen, am Ende noch zu den verfügbaren Gleisen gematcht werden können. Dies können wir mit Hilfe des Hall Theorems erreichen.

In unserem Fall sind V_1 die Menge der gewählten Sequenzen, V_2 die Menge der Gleise, und $(g, o) \in E$ genau dann, wenn $s(g) \leq s(o)$ gilt, wobei $s(g)$ die Länge der Sequenz g bezeichnet, welches gleich der Länge des längsten Zuges ist, der in g auftritt. Schließlich bezeichnet $s(o)$ wie oben die Länge des Gleises o . Im Folgenden gilt weiterhin o.B.d.A., dass die Gleise fallend bzgl. ihrer Längen geordnet sind. Das heißt, dass auf das erste Gleis alle Sequenzen passen. Wir partitionieren die Sequenzen und die Gleise in die in Kapitel 2.2 eingeführten Buckets, sodass jede Sequenz auf jedes Gleis in dem selben Bucket passt. Die Länge eines Buckets entspricht dabei der Länge des längsten Gleises in dem Bucket. Sei B die Menge aller Bucketlängen. Die symmetriefreie Formulierung (S-EF) für das Rangierproblem sieht wie folgt aus

(S-EF)

$$\min \quad \sum_{g \in G} c(g) \cdot x_g \quad (4.32)$$

$$\text{s.t.} \quad \sum_{\substack{g \in G \\ r \in g}} x_g \geq 1 \quad r \in R \quad (4.33)$$

$$\sum_{g \in G} s_p(g) \cdot x_g \leq s^{\text{mix}} \quad p \in P \quad (4.34)$$

$$\sum_{g \in G} x_g \leq |\{o : s(o) \geq s\}| \quad s \in B \quad (4.35)$$

$$x \in \{0, 1\}. \quad (4.36)$$

Die ersten beiden Nebenbedingungen entsprechen denen aus der erweiterten Formulierung. Die Restriktion (4.35) folgt aus dem Hall Theorem. Diese stellt für alle Bucketlängen sicher, dass die Anzahl der Sequenzen in einem Bucket nicht größer ist als die Zahl der Gleise, auf die diese jeweils passen.

Die Lösungen des Problems werden eine Anzahl von Sequenzen sein, die dann zu den Rangiergleisen gematcht werden können. Wir werden im Folgenden zeigen, dass solch ein Matching immer existiert.

(4.6) Theorem

Eine Lösung des (S-EF) kann immer zu den Rangiergleisen gematcht werden.

Beweis

Wir möchten folgende Aussage zeigen:

Die Ungleichungen (4.35) sind erfüllt $\Leftrightarrow \exists$ ein Matching.

\Leftarrow : Folgt direkt aus Hall's Theorem.

\Rightarrow : Per vollständiger Induktion nach i , wobei

- $B_1, \dots, B_i \hat{=}$ Buckets der Länge $s_1 \geq, \dots, \geq s_{B_i}$.
- Bucket B_i besitzt g_{n_i} Sequenzen.
- $\Gamma(B_1), \dots, \Gamma(B_i) \hat{=}$ Schienen im jeweiligen Bucket, zu denen die g_{n_i} Sequenzen zugewiesen werden können.

(IA): Für $i = 1$ gilt:

$$|\Gamma(B_1)| \geq g_{n_1} \Rightarrow \exists \text{ Matching,}$$

denn wähle z.B. das größte Gleis für g_1 , das zweitgrößte für g_2, \dots , das n_1 größte für g_{n_1} . (IS): Für $i \rightarrow i + 1$

Zu Zeigen:

$$|\Gamma(B_{i+1})| \geq \sum_{j=1}^{i+1} g_{n_j} \Rightarrow \exists \text{ Matching.}$$

Nach (IA) sind für B_1, \dots, B_i höchstens $\sum_{j=1}^i g_{n_j}$ Schienen verbraucht. Somit existieren noch mindestens n_{i+1} Schienen für B_{i+1} . \square

Für das Modell (S-EF) könnte das Branching mit Hilfe des Ryan-Foster Branchings realisiert werden. Diese alternative Branching-Regel brancht direkt auf den Master-Variablen.

Die Grundlage für eine effektive Branching-Regel, welche erstmals von Ryan und Foster angewandt wurde, ist die in dem folgenden Satz zusammengefasste Beobachtung.

(4.7) Satz

Es sei $\mathbf{A} = (a_{ij})$ eine $\{0, 1\}$ -Matrix und λ eine Basislösung des Systems $\mathbf{A}\lambda = \mathbf{1}$, $\lambda \geq \mathbf{0}$. Ist λ nicht ganzzahlig (d.h. mindestens eine Komponente λ_k nicht ganzzahlig), so existieren zwei Zeilen s und t mit folgender Eigenschaft:

$$0 < \sum_{t:a_{rt}=1, a_{st}=1} \lambda_t < 1.$$

Das heißt, dass (MP) ist als Set-Partitioning-Formulierung modelliert. Zusammengefasst sieht das (MP) wie folgt aus

$$\begin{aligned} \min & \sum_{j \in J} c_j \lambda_j \\ \text{s.t.} & \sum_{j \in J} \mathbf{a}_j \lambda_j = \mathbf{1} \\ & \lambda_j \geq 0, j \in J. \end{aligned}$$

Eine Lösung λ des (RMP) ist damit genau dann ganzzahlig, wenn für jedes Paar von Zeilen r und s die Summe $\sum_{t:a_{rt}=1, a_{st}=1} \lambda_t$ entweder gleich null oder gleich eins ist. Im Branching wird nun nach einem solchen nicht-ganzzahligen $\sum_{t:a_{rt}=1, a_{st}=1} \lambda_t$ verzweigt. Das (RMP) enthält dann nur noch Spalten, in denen nicht gleichzeitig in der r -ten und s -ten Zeile eine Eins steht. Hingegen bewirkt $\sum_{t:a_{rt}=1, a_{st}=1} \lambda_t = 1$, dass das (RMP) dann nur noch aus Spalten besteht, in denen in der r -ten und s -ten Zeile beide Werte gleich null oder gleich eins sind. Beide Zweige führen auf kleinere Set-Partitioning-Probleme.

Für das Pricing können die Bedingungen $\sum_{t:a_{rt}=1, a_{st}=1} \lambda_t = 0$ oder $\sum_{t:a_{rt}=1, a_{st}=1} \lambda_t = 1$ oft ohne größere Schwierigkeiten berücksichtigt werden. Nehmen wir dazu an, dass die Auswahl einer abzudeckenden Zeile r durch eine Entscheidungsvariable $x_r \in \{0, 1\}$ modelliert wird. Dann bedeutet $\sum_{t:a_{rt}=1, a_{st}=1} \lambda_t = 0$, dass entweder x_r oder x_s (oder beide) gleich null sein müssen. Dies kann im Pricing durch die zusätzliche Bedingung $x_r + x_s \leq 1$ berücksichtigt werden. Der alternativ Zweig $\sum_{t:a_{rt}=1, a_{st}=1} \lambda_t = 1$ impliziert, dass x_r und x_s identisch sein müssen, d.h. $x_r = x_s = 0$ oder $x_r = x_s = 1$.

Nun möchten wir diese Branching-Strategie von Ryan und Foster auf das Rangierproblem anwenden. Da diese Regel direkt auf den Master-Variablen brancht, ohne einen unbalancierten Branching-Baum zu erzeugen, brauchen wir die Hilfsvariablen y_{r0} nicht. Ohne diese Variablen erhalten wir folgende Formulierung (ef):

(ef)

$$\min \sum_{\substack{o \in O \\ g \in G(o)}} c(g) \cdot x_{go} \quad (4.37)$$

$$\text{s.t.} \quad \sum_{o \in O} \sum_{\substack{g \in G(o) \\ r \in g}} x_{go} \geq 1 \quad r \in R \quad (4.38)$$

$$\sum_{g \in G(o)} x_{go} \leq 1 \quad o \in O \quad (4.39)$$

$$\sum_{\substack{o \in O \\ g \in G(o)}} s_p(g) \cdot x_{go} \leq s^{\text{mix}} \quad p \in P \quad (4.40)$$

$$x \in \{0, 1\}. \quad (4.41)$$

Für das Rangierproblem können wir die Ryan-Foster-Bedingung im Pricing leicht berücksichtigen. Da das Pricing-Problem als längste Wege Problem in einem azyklischen Graphen gelöst wird, kann die Branching-Bedingung wie folgt umgesetzt werden.

- $\sum_{rs} = 0$: Im Pricing würde dies bedeuten, dass die Kante von Zug r zu s auf dem längsten Weg nicht enthalten werden darf. Dies können wir realisieren, indem wir dieser Kante in dem gerichteten Graphen ein sehr kleines Gewicht zuordnen, sodass diese Kante in der optimalen Lösung des Subproblems nicht gewählt wird.
- $\sum_{rs} = 1$: Die Übersetzung dieser Bedingung im Pricing ist, dass die Kante zwischen den Zügen r und s auf dem längsten Weg enthalten sein muss. Dies können wir realisieren, indem wir die Knoten, die die Züge r und s repräsentieren, zu einem Knoten zusammenfassen.

Dadurch bleibt uns die Struktur, das Subproblem als längste Wege Problem zu lösen, erhalten.

5 Eine arc-indexed Formulierung für das Rangierproblem

In diesem Kapitel werden wir eine letzte Formulierung für das Rangierproblem angeben. Diese Formulierung erweist sich als die sogenannte originale Formulierung in der Dantzig-Wolfe Dekomposition. Schließlich werden wir auch diesen Ansatz auf den 192 Probleminstanzen testen und die Ergebnisse mit der anfänglich vorgestellten erweiterten Formulierung vergleichen.

5.1 Dantzig-Wolfe Dekomposition

Der einfachsten Form der Spaltengenerierung begegnet man in der Dantzig-Wolfe Dekomposition. Die Dantzig-Wolfe Dekomposition für lineare Programme kann auf MIPs auf zwei verschiedene Weisen übertragen werden:

- Der Konvexifizierungsansatz.
- Der Diskretisierungsansatz.

Wir werden hier nur den ersten Ansatz einführen, denn das Rangierproblem ist ein binäres Programm und für diesen Spezialfall fallen der Konvexifizierungs- und Diskretisierungsansatz zusammen (siehe [24]).

Wir betrachten das folgende IP, welches wir als **originale Formulierung** bezeichnen.

(DW-OF)

$$z_{DW-OF}^* = \min \mathbf{c}^T \mathbf{x} \tag{5.1}$$

$$\text{s.t.} \quad \mathbf{Ax} \geq \mathbf{b} \tag{5.2}$$

$$\mathbf{Dx} \geq \mathbf{d} \tag{5.3}$$

$$\mathbf{x} \in \mathbb{Z}^n. \tag{5.4}$$

Wir führen die Menge X ein, welche die Bedingungen (5.3) und (5.4) enthält, d.h.

$$X = \{\mathbf{x} \in \mathbb{Z}^n \mid \mathbf{D}\mathbf{x} \geq \mathbf{d}\}.$$

Mit Hilfe dieser Menge erhalten wir die **kompakte Formulierung**.

(DW-KF)

$$z_{DW-KF}^* = \min \mathbf{c}^T \mathbf{x} \tag{5.5}$$

$$\text{s.t.} \quad \mathbf{A}\mathbf{x} \geq \mathbf{b} \tag{5.6}$$

$$\mathbf{x} \in X. \tag{5.7}$$

Das Theorem von Minkowski und Weyl ermöglicht uns, jedes $\mathbf{x} \in X$ als Konvexkombination von Extrempunkten $\{\mathbf{x}_p\}_{p \in P}$ plus eine nicht-negative Kombination von Extremalstrahlen $\{\mathbf{x}_r\}_{r \in R}$ darzustellen.

(5.1) Theorem

Genau dann ist eine Teilmenge $P \subseteq \mathbb{R}^n$ ein Polyeder, wenn es endliche Mengen $V, W \subseteq \mathbb{R}^n$ mit der Eigenschaft

$$P = \text{conv}(V) + \text{cone}(W)$$

gibt, wobei wir mit *conv* die Konvexkombination und mit *cone* die Kegelkombination bezeichnen.

Somit gilt für jedes $\mathbf{x} \in X$

$$\mathbf{x} = \sum_{p \in P} \mathbf{x}_p \cdot \lambda_p + \sum_{r \in R} \mathbf{x}_r \cdot \lambda_r, \tag{5.8}$$

$$\sum_{p \in P} \lambda_p = 1,$$

$$\boldsymbol{\lambda} \in \mathbb{R}_+^{|P| \times |R|}.$$

Nun ersetzen wir \mathbf{x} in der kompakten Formulierung mit (5.8) und setzen $c_j = \mathbf{c}^T \mathbf{x}_{p_j}$ und $\mathbf{a}_j = \mathbf{A}\mathbf{x}_{p_j}$, $j \in P \cup R$. Dadurch erhalten wir die **erweiterte Formulierung**

(DW-EF)

$$z_{DW-EF}^* = \min \quad \sum_{p \in P} c_p \cdot \lambda_p + \sum_{r \in R} c_r \cdot \lambda_r \quad (5.9)$$

$$\text{s.t} \quad \sum_{p \in P} \mathbf{a}_p \cdot \lambda_p + \sum_{r \in R} \mathbf{a}_r \cdot \lambda_r \geq \mathbf{b} \quad (5.10)$$

$$\sum_{p \in P} \lambda_p = 1 \quad (5.11)$$

$$\boldsymbol{\lambda} \geq \mathbf{0} \quad (5.12)$$

$$\mathbf{x} = \sum_{p \in P} \mathbf{x}_p \cdot \lambda_p + \sum_{r \in R} \mathbf{x}_r \cdot \lambda_r \quad (5.13)$$

$$\mathbf{x} \in \mathbb{Z}^n. \quad (5.14)$$

Wenn wir die LP-Relaxation der erweiterten Formulierung lösen möchten, relaxieren wir die Ganzzahligkeitsbedingung und entfernen daher die Nebenbedingung (5.14). Da die Vektoren \mathbf{x} nur benutzt werden, um die Ganzzahligkeit zu erzwingen, können wir nach Entfernung von (5.14) auch die Restriktion (5.13) weglassen. Somit erhalten wir das **Master-Problem** (DW-MP)

(DW-MP)

$$z_{DW-MP}^* = \min \quad \sum_{p \in P} c_p \cdot \lambda_p + \sum_{r \in R} c_r \cdot \lambda_r \quad (5.15)$$

$$\text{s.t} \quad \sum_{p \in P} \mathbf{a}_p \cdot \lambda_p + \sum_{r \in R} \mathbf{a}_r \cdot \lambda_r \geq \mathbf{b} \quad (5.16)$$

$$\sum_{p \in P} \lambda_p = 1 \quad (5.17)$$

$$\boldsymbol{\lambda} \geq \mathbf{0}. \quad (5.18)$$

$$(5.19)$$

D.h. durch Anwenden der Dantzig-Wolfe Dekomposition auf eine originale Formulierung erhalten wir die uns aus Kapitel 4.1 bekannte erweiterte Formulierung, welche wir auf Grund der exponentiellen Anzahl der Variablen mit Spaltengenerierung lösen.

5.2 Die originale Formulierung

Da immer eine originale Formulierung existiert (siehe: [25]), werden wir nun zu der erweiterten Formulierung (ef) aus Kapitel 4.6 eine originale Formulierung aufstellen.

5 Eine arc-indexed Formulierung für das Rangierproblem

Dabei ist jede Variable, die wir benutzen, durch Paare von Zügen $r \prec r'$ und einem Gleis o indiziert (**arc-indexed**):

$$x_{rr'}^o = \begin{cases} 1, & \text{wenn } r \prec r' \text{ auf } o, \\ 0, & \text{sonst.} \end{cases}$$

$$x_{ur'}^o = \begin{cases} 1, & \text{wenn } r' \text{ der erste Zug auf } o \text{ ist,} \\ 0, & \text{sonst.} \end{cases}$$

$$x_{r'v}^o = \begin{cases} 1, & \text{wenn } r' \text{ der letzte Zug auf } o \text{ ist,} \\ 0, & \text{sonst.} \end{cases}$$

Zusätzlich brauchen wir noch die folgenden zwei Mengen, um das Modell aufzustellen:

$$A_{r'}^o = \{r \in R \mid r' \prec r \text{ in einem zulässigen Teilschedule auf Gleis } o\},$$

$$B_{r'}^o = \{r \in R \mid r \prec r' \text{ in einem zulässigen Teilschedule auf Gleis } o\}.$$

Somit erhalten wir die originale Formulierung (OF) für das Rangierproblem

(OF)

$$\min \sum_{o \in O} \sum_{r, r': r \prec r'} c(r, r') \cdot x_{rr'}^o \quad (5.20)$$

$$\text{s.t.} \quad \sum_{o \in O} \sum_{r \in B_{r'}^o \cup \{u\}} x_{rr'}^o = 1 \quad r' \in R \quad (5.21)$$

$$\sum_{r' \in R} x_{ur'}^o \leq 1 \quad o \in O \quad (5.22)$$

$$\sum_{o \in O} \sum_{r, r': r \prec r'} s_p(r, r') \cdot x_{rr'}^o \leq s^{\text{mix}} \quad p \in P \quad (5.23)$$

$$\sum_{r \in B_{r'}^o \cup \{u\}} x_{rr'}^o = \sum_{r \in A_{r'}^o \cup \{v\}} x_{r'r}^o \quad r' \in R, o \in O \quad (5.24)$$

$$x \in \{0, 1\}. \quad (5.25)$$

Die Zielfunktion (5.20) minimiert die Anzahl der extra Roll-Ins. Die Nebenbedingungen (5.21) und (5.22) stellen jeweils sicher, dass jeder Zug genau einmal einem

Gleis zugeordnet wird und dass jedes Gleis höchstens einmal benutzt wird. Die Ungleichungen (5.23) sichern, dass nicht mehr als die vorhandene mixing Kapazität gebraucht wird. Die Restriktionen (5.24) garantieren, dass die Zuordnungen von Zügen zu den Schienen wohldefiniert sind. Diese Bedingungen haben die gleiche Bedeutung wie die Flusserhaltungsbedingungen in vielen Netzwerkflussproblemen. Die Nebenbedingungen (5.24) und (5.25) gewährleisten, dass der Teilschedule auf jedem Gleis zulässig ist.

Im weiteren werden wir zeigen, dass wir durch Anwenden der Dantzig-Wolfe Dekomposition auf die (OF) die (ef) erhalten und somit ein äquivalentes Modell für das Rangierproblem aufgestellt haben.

(5.2) Definition

Zwei binäre Programme BP und BP' heißen **äquivalent**, falls es zu jeder zulässigen Lösung x von BP eine zulässige Lösung x' von BP' mit gleichem Zielfunktionswert gibt und umgekehrt.

Beim Anwenden der Dantzig-Wolfe Dekomposition zerlegen wir die Formulierung (OF) in das Master-Problem bestehend aus den Nebenbedingungen (5.21), (5.22) und (5.23). Die $|O|$ Pricing-Probleme, welche wir bekannterweise als längste Wege Probleme lösen, erhalten zusätzlich die Restriktion (5.24). Es ist leicht einzusehen, dass bei festem o jedes $x_{rr'}^o$ im zulässigen Bereich des o -ten Subproblems einem zulässigen Teilschedule auf Gleis o entspricht.

Sei wie gehabt $G(o)$ die Menge aller zulässigen Sequenzen auf dem Gleis o . Für jedes Zugpaar r, r' mit $r \prec r'$ sei $\alpha_{rr',g}^o = 1$, falls die Sequenz g die Züge r, r' enthält und null sonst. Zusätzlich benutzen wir die uns bereits bekannten Variablen x_{go} , welche modellieren, ob die Sequenz g auf dem Gleis o benutzt wird oder nicht.

Da jedes $x_{rr'}^o \in X = \left\{ x_{rr'}^o \in \{0, 1\} \mid \sum_{r \in B_{r'}^o \cup \{u\}} x_{rr'}^o = \sum_{r \in A_{r'}^o \cup \{v\}} x_{r'r}^o, r' \in R \right\}$ sich als Konvexkombination der Extrempunkte von X beschreiben lässt, gilt

$$x_{rr'}^o = \sum_{g \in G(o)} x_{go} \cdot \alpha_{rr',g}^o.$$

Wegen der Beschränktheit der Menge X haben wir keine Extremalstrahlen. Durch Einsetzen dieser Beschreibung von $x_{rr'}^o$ in die obige originale Formulierung und unter Berücksichtigung der Beziehungen

$$\begin{aligned}
 c(g) &= \sum_{r,r':r \prec r'} c(r,r') \alpha_{rr',g}^o \\
 &= \sum_{(r,r') \in g} c(r,r'), \\
 s_p(g) &= \sum_{r,r':r \prec r'} s_p(r,r') \alpha_{rr',g}^o \\
 &= \sum_{(r,r') \in g} s_p(r,r'),
 \end{aligned}$$

erhalten wir als äquivalentes Programm

$$\min \sum_{\substack{o \in O \\ g \in G(o)}} c(g) \cdot x_{go} \tag{5.26}$$

$$\text{s.t. } \underbrace{\sum_{o \in O} \sum_{r \in B_r^o \cup \{u\}} \sum_{\substack{g \in G(o) \\ r' \in g}} x_{go} \alpha_{rr',g}^o}_{\sum_{o \in O} \sum_{\substack{g \in G(o) \\ r' \in g}} x_{go}} \geq 1 \quad r' \in R \tag{5.27}$$

$$\underbrace{\sum_{r' \in R} \sum_{g \in G(o)} x_{go} \alpha_{ur',g}^o}_{\sum_{g \in G(o)} x_{go}} \leq 1 \quad o \in O \tag{5.28}$$

$$\sum_{\substack{o \in O \\ g \in G(o)}} s_p(g) \cdot x_{go} \leq s^{\text{mix}} \quad p \in P \tag{5.29}$$

$$x \in \{0, 1\}. \tag{5.30}$$

Auf Grund der Bedingung (5.28) sind die Konvexitätsbedingungen $\sum_{p \in P} \lambda_p = 1$ redundant. Insgesamt ist somit das obige Modell äquivalent zu der erweiterten Formulierung (ef).

5.3 Integralitätseigenschaft des Subproblems

Generell liegt der Vorteil beim Einsatz eines Column-Generation-Verfahrens in erster Linie im Gewinnen einer guten Schranke. Die aus dem Master-Problem gewonnene Schranke ist nämlich in der Regel besser als die mittels einer LP-Relaxation der

originalen Formulierung gewonnene. Den Vorteil einer solchen schärferen Schranke erkaufte man sich damit, dass die Lösung des Pricing-Problems mehr Aufwand erfordert, als die Lösung eines linearen Programms. Diese Eigenschaft lässt sich auch theoretisch begründen.

(5.3) Definition

Es seien LP und LP' die LP-Relaxationen zweier äquivalenter Formulierungen BP und BP'. Dann heißt LP **mindestens genauso stark** wie LP', falls es zu jeder zulässigen (fraktionalen) Lösung x von LP auch eine zulässige (fraktionale) Lösung x' von LP' mit gleichem Zielfunktionswert gibt.

(5.4) Definition

Seien Z und Z' jeweils die zulässigen Bereiche von LP und LP'. LP heißt **stärker** als LP', falls es zusätzlich ein Element $x' \in Z'$ ohne korrespondierendes Element $x \in Z$ mit demselben Zielfunktionswert gibt.

Im Folgenden analysieren wir die Stärke der Schranke, die wir aus der originalen Formulierung für das Rangierproblem erhalten. Per Konstruktion sind das originale Programm (OF) und die erweiterte Formulierung (ef) äquivalent. Die Korrespondenz zwischen den Elementen wird durch

$$x_{rr'}^0 = \sum_{g \in G(o)} x_{g0} \cdot \alpha_{rr',g}^0$$

hergestellt. Dies zeigt auch, dass die LP-Relaxation der (ef) mindestens so stark ist wie die LP-Relaxation der originalen Formulierung.

Die Betrachtung zeigt ferner, dass das (RMP) exakt das Problem

$$\min \sum_{o \in O} \sum_{r,r': r \prec r'} c(r,r') \cdot x_{rr'}^0 \tag{5.31}$$

$$\text{s.t.} \quad \sum_{o \in O} \sum_{r \in B_r^o \cup \{u\}} x_{rr'}^0 = 1 \quad r' \in R \tag{5.32}$$

$$\sum_{r' \in R} x_{ur'}^0 \leq 1 \quad o \in O \tag{5.33}$$

$$\sum_{o \in O} \sum_{r,r': r \prec r'} s_p(r,r') \cdot x_{rr'}^0 \leq s^{\text{mix}} \quad p \in P \tag{5.34}$$

$$x \in \text{conv}(X) \tag{5.35}$$

löst. Notwendig dafür, dass die LP-Relaxation des Master-Problems (echt) stärker ist als die LP-Relaxation der (OF), ist das echte Enthaltensein von $conv(X)$ im zulässigen Bereich der LP-Relaxation, also

$$conv(X) \subsetneq \left\{ x \in \mathbb{R}_+ \mid \sum_{r \in B_r^o \cup \{u\}} x_{rr'}^o = \sum_{r \in A_r^o \cup \{v\}} x_{rr'}^o, r' \in R \right\}.$$

Bei dem Pricing-Problem des Rangierproblems gilt die Gleichheit

$$conv(X) = \left\{ x \in \mathbb{R}_+ \mid \sum_{r \in B_r^o \cup \{u\}} x_{rr'}^o = \sum_{r \in A_r^o \cup \{v\}} x_{rr'}^o, r' \in R \right\}.$$

Somit besitzt dieses die sogenannte **Integralitätseigenschaft**. Denn diese Eigenschaft bedeutet, dass die Lösungen des Pricing-Problems immer ganzzahlig sind, auch wenn es als lineares Programm gelöst wird. Dies ist genau der Fall für das längste Wege Problem und daher erhalten wir den Wert der LP-Relaxation des Master-Problems. Also ist die LP-Relaxation der (ef) nicht stärker als die LP-Relaxation der (OF).

Nun kann es aber trotzdem sein, dass auf Grund der pseudo-polynomiell großen Anzahl der Variablen in (OF) das direkte Lösen des Problems nicht effizient möglich ist. Dies werden wir im nächsten Unterkapitel untersuchen.

5.4 Ergebnisse

Nun testen wir das in 5.2 eingeführte, kompakte Modell auf den 192 Probleminstanzen und vergleichen die Ergebnisse mit der erweiterten Formulierung. Diesmal verwenden wir eine gefundene, heuristische Lösung nicht als Startlösung. Für jede Test-Instanz stehen wieder 20 Minuten zur Verfügung. Die Ergebnisse werden in der Tabelle 5.1 dargestellt.

Zu Beobachten ist, dass die (OF) bei größeren Instanzen eine kürzere Laufzeit hat. Obwohl die (EF) bei sieben Instanzen innerhalb der 20min keine zulässige Lösung finden kann, löst die (OF) alle 192 Test-Instanzen optimal.

Tabelle 5.1: Ergebnisse für verschiedene Planungshorizonte für (OF) und (EF). Für jeden Planungshorizont sind die Anzahl der Instanzen in der Datei und die durchschnittliche Problemgröße für jede Instanz dargestellt. Die Tabelle zeigt die durchschnittliche Anzahl der extra Roll-Ins und die durchschnittliche Ausführungszeit.

		<i>Planungshorizont (Tage)</i>			
		2	3	4	5
<i>Anzahl der Instanzen</i>		75	50	37	30
<i>Durchschn. Anzahl der Züge</i>		48.7	73.0	97.7	121.7
<i>Durchschn. Anzahl der Gruppen</i>		244.9	367.3	492.0	612.2
OF	Durchsch. extra Roll-Ins	10.1	18.2	27.6	39.0
	Durchsch. Zeit (s)	13.5	36.9	134.4	207.7
	Optimalität bewiesen	75	50	37	30
EF	Durchsch. extra Roll-Ins	10.1	18.2	35.4	72.8
	Durchsch. Zeit (s)	7.2	50.7	134.5	454.3
	Optimalität bewiesen	75	50	36	24

Die (EF) braucht mehrere Tage für das Lösen von größeren Instanzen, wie z.B. die eine Woche bzw. zwei Wochen Daten. Daher testen wir die originale Formulierung auf diesen größeren Instanzen. Für die eine Woche Instanz benötigt die (OF) circa 23 Minuten und die zwei Wochen Datei wird in ungefähr 12 Stunden gelöst.

6 Fazit und Ausblick

In dieser Arbeit haben wir anfänglich die Ergebnisse aus den Papern [1] und [2] präsentiert. Dort wurde für die Lösung des Rangierproblems neben heuristischen Ansätzen auch ein binäres Modell, die direkte Formulierung, aufgestellt. Doch da sich diese Ansätze als nicht effizient ergaben, haben wir in dieser Arbeit auch den Spaltengenerierungsansatz des Papers [3] für das Rangierproblem eingeführt. Für das Pricing-Problem wurde gezeigt, dass dieses als längste Wege Problem in einem gerichteten, azyklischen Graphen gelöst werden kann. Es wurde eine Branching-Regel aufgestellt, die leicht in das Pricing-Problem, durch Modifizieren des Graphen, integriert werden konnte. Bei den Tests, die auf der Datei von Hallsberg durchgeführt wurden, erwies die (EF) eine starke duale Schranke. Mit diesem Ansatz konnten wir optimale Lösungen für alle Probleminstanzen in einer angemessenen Zeit finden.

Weiterhin haben wir uns mit Techniken zur Beschleunigung des Lösungsverfahrens beschäftigt. Erfolge und Misserfolge dieser Techniken wurden dargestellt und erklärt. Zuerst haben wir das Prinzip des stabilized Column-Generation theoretisch dargestellt. Da die Spaltengenerierung im Primalen der Zeilengenerierung im Dualen entspricht, kann durch eine Stabilisierung der dualen Variablen in manchen Anwendungen eine deutlich schnellere Konvergenz erreicht werden. Stabilisierte duale Variablen können nämlich einerseits zur Beschleunigung des Pricings und andererseits zur Reduzierung der Zahl der Iterationen bei der Lösung des Master-Problems beitragen. Da diese Stabilisierungsansätze eine Modifizierung des (RMP)s erfordern, haben wir die Smoothing Methode eingeführt und diese auf das Rangierproblem angewandt. Schließlich haben wir noch mit den Techniken Early Termination und Early Branching versucht dem Tailing-Off und Bang-Bang Effekt entgegenzuwirken.

Letztenendes haben wir eine originale Formulierung aufgestellt und untersucht, ob dieser Ansatz bessere Ergebnisse liefert. Zu beobachten war, dass dieser Ansatz bei den größeren Instanzen eine viel geringere Laufzeit erwies. Mit Hilfe dieser Formulierung gelang es uns, die eine Woche und zwei Wochen Test-Instanzen zu lösen.

Obwohl einige vielversprechende Ergebnisse präsentiert wurden, gibt es noch viele offene Fragen, die in der zukünftigen Forschung berücksichtigt werden müssen.

Wie in Abschnitt 4 zu sehen war, konnten alle Probleminstanzen in einer angemessenen Zeit optimal gelöst werden. Trotzdem nimmt das optimale Lösen von größeren Instanzen zu viel Zeit in Anspruch. Wir glauben, dass dies vor allem an der Symmetrie in der Aufgabenstellung liegt. Die in dem Unterkapitel 4.6 vorgestellten theoretischen Ansätze müssten in der Zukunft noch auf das Rangierproblem umgesetzt werden und die daraus erhaltenen Ergebnisse gedeutet werden. Auch die originale Formulierung kann weiterentwickelt werden. Denn für die größeren Instanzen stellte sich heraus, dass diese Formulierung schneller als die erweiterte Formulierung ist. In der Zukunft könnte man eine Kombination aus der (EF) und der (OF) benutzen, um das Rangierproblem effizient zu lösen.

In realen Anwendungen sind die genauen Wagonlisten und Zeiten aller eingehenden und ausgehenden Züge normalerweise zeitlich nicht weit im Voraus bekannt. Daher muss die Planung flexibel sein, und der Rangierzeitplan sollte regelmäßig aktualisiert werden. Dies deutet auf mehrere mögliche weitere Forschungsrichtungen:

- Ändern des Rangierzeitplans könnte sich als eine komplizierte Operation erweisen, denn die Zusammenstellung einiger Züge könnte bereits begonnen haben. Deshalb ist es wünschenswert, den ursprünglichen Zeitplan so zu konstruieren, dass dieser leicht wiederhergestellt werden kann. Forschungen aus dem Gebiet der wiederherstellbaren Robustheit könnten hier eingesetzt werden.
- Erstellen des Rangierzeitplans, ohne dass wir alle Informationen über die Zukunft haben, fügt dem Problem eine online Komponente hinzu. Es wäre interessant zu sehen, ob das Problem als ein Online Problem behandelt werden kann.

7 Anhang

7.1 Ergebnisse der Heuristik++, (DF) und (EF)

Tabelle 7.1: Ergebnisse für einen Planungshorizont von 2 Tagen. Enthalten ist die Anzahl der extra Roll-Ins (ER), die Laufzeit und der Optimalitätsgap. \bar{x} ist das arithmetische Mittel für alle zulässige Instanzen. Ein – bedeutet, dass keine zulässige Lösung gefunden wurde.

	<i>Instanz</i>		<i>Heuristik++</i>		<i>DF</i>			<i>EF</i>		
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	Gap (%)	ER (#)	Zeit (s)	Gap (%)
1	32	90	0	0.0	0	0.4	0.0	0	0.0	0.0
2	73	380	37	0.2	36	1201.8	100.0	35	19.5	0.0
3	67	368	9	0.1	5	1202.9	100.0	5	7.6	0.0
4	30	132	0	0.0	0	0.4	0.0	0	0.0	0.0
5	35	121	0	0.0	0	0.4	0.0	0	0.0	0.0
6	58	235	0	0.1	0	1.1	0.0	0	0.1	0.0
7	14	46	0	0.0	0	0.1	0.0	0	0.0	0.0
8	7	9	0	0.0	0	0.1	0.0	0	0.0	0.0
9	48	161	0	0.0	0	0.6	0.0	0	0.1	0.0
10	43	169	0	0.0	0	0.7	0.0	0	0.1	0.0
11	1	1	0	0.0	0	0.0	0.0	0	0.0	0.0
12	51	155	0	0.0	0	0.7	0.0	0	0.1	0.0
13	61	236	0	0.0	0	1.0	0.0	0	0.1	0.0
14	25	96	0	0.0	0	0.2	0.0	0	0.0	0.0
15	28	91	0	0.0	0	0.4	0.0	0	0.0	0.0
16	77	366	20	0.1	20	1201.6	100.0	20	3.4	0.0
17	59	344	12	0.1	11	1201.2	100.0	11	3.5	0.0
18	34	147	0	0.0	0	0.4	0.0	0	0.0	0.0
19	51	250	2	0.0	2	19.6	0.0	2	0.3	0.0
20	78	450	38	0.1	28	1201.7	100.0	28	10.5	0.0
21	42	225	0	0.0	0	0.6	0.0	0	0.0	0.0
22	29	83	0	0.0	0	0.3	0.0	0	0.0	0.0
23	56	262	0	0.0	0	0.9	0.0	0	0.1	0.0

Tabelle 7.1: Ergebnisse für einen Planungshorizont von 2 Tagen (Fortsetzung von der vorhergehenden Seite).

	<i>Instanz</i>		<i>Heuristik++</i>		<i>DF</i>			<i>EF</i>		
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	Gap (%)	ER (#)	Zeit (s)	Gap (%)
24	51	222	0	0.0	0	0.8	0.0	0	0.1	0.0
25	28	142	0	0.0	0	0.3	0.0	0	0.0	0.0
26	47	191	0	0.0	0	0.7	0.0	0	0.1	0.0
27	75	381	3	0.1	3	1202.0	100.0	3	1.7	0.0
28	48	221	0	0.0	0	0.5	0.0	0	0.1	0.0
29	26	117	0	0.0	0	0.4	0.0	0	0.0	0.0
30	70	398	14	0.1	12	1201.2	100.0	12	1.9	0.0
31	59	307	0	0.0	0	1.0	0.0	0	0.1	0.0
32	26	97	0	0.0	0	0.2	0.0	0	0.0	0.0
33	49	218	2	0.0	0	1.2	100.0	0	0.4	0.0
34	78	448	42	0.1	28	1201.5	100.0	28	2.5	0.0
35	48	237	0	0.0	0	0.8	0.0	0	0.1	0.0
36	23	109	0	0.0	0	0.3	0.0	0	0.0	0.0
37	71	336	0	0.0	0	1.2	0.0	0	0.1	0.0
38	74	385	15	0.1	11	1201.5	100.0	11	11.7	0.0
39	37	144	0	0.0	0	0.4	0.0	0	0.0	0.0
40	46	237	0	0.0	0	0.8	0.0	0	0.1	0.0
41	69	432	119	0.1	77	1203.7	2.6	77	2.0	0.0
42	39	211	0	0.0	0	0.5	0.0	0	0.0	0.0
43	24	79	0	0.0	0	0.2	0.0	0	0.0	0.0
44	79	411	35	0.1	29	1201.5	100.0	29	11.6	0.0
45	62	341	2	0.0	2	12.9	100.0	2	0.7	0.0
46	34	166	0	0.0	0	0.4	0.0	0	0.0	0.0
47	60	282	5	0.1	4	1201.0	100.0	4	0.7	0.0
48	74	501	40	0.1	24	1201.5	100.0	24	11.1	0.0
49	57	331	6	0.1	1	13.6	0.0	1	0.7	0.0
50	35	152	0	0.0	0	0.5	0.0	0	0.0	0.0
51	74	429	30	0.1	24	1201.5	100.0	24	5.2	0.0
52	68	405	22	0.1	11	1204.5	100.0	11	3.9	0.0
53	36	175	0	0.0	0	0.4	0.0	0	0.0	0.0
54	45	211	0	0.0	0	0.7	0.0	0	0.0	0.0
55	60	347	28	0.1	27	1201.1	100.0	27	3.7	0.0
56	49	305	0	0.0	0	0.7	0.0	0	0.0	0.0
57	36	162	0	0.0	0	0.5	0.0	0	0.0	0.0
58	66	372	5	0.1	5	1258.1	100.0	5	1.2	0.0
59	65	384	23	0.1	17	1201.3	100.0	17	1.9	0.0

7.1 Ergebnisse der Heuristik++, (DF) und (EF)

Tabelle 7.1: Ergebnisse für einen Planungshorizont von 2 Tagen (Fortsetzung von der vorhergehenden Seite).

	<i>Instanz</i>		<i>Heuristik++</i>		<i>DF</i>			<i>EF</i>		
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	Gap (%)	ER (#)	Zeit (s)	Gap (%)
60	39	153	0	0.0	0	0.4	0.0	0	0.1	0.0
61	51	270	–	0.1	134	8.4	0.7	134	1.7	0.0
62	76	440	36	0.1	26	1201.4	100.0	26	3.3	0.0
63	54	317	0	0.0	0	0.8	0.0	0	0.1	0.0
64	29	111	0	0.0	0	0.4	0.0	0	0.0	0.0
65	80	445	43	0.1	40	1202.5	100.0	40	14.7	0.0
66	45	266	0	0.0	0	0.7	0.0	0	0.1	0.0
67	3	5	0	0.0	0	0.0	0.0	0	0.0	0.0
68	3	5	0	0.0	0	0.0	0.0	0	0.0	0.0
69	90	469	19	0.1	19	1201.7	100.0	19	4.9	0.0
70	53	325	0	0.0	0	0.7	0.0	0	0.1	0.0
71	32	119	0	0.0	0	0.5	0.0	0	0.0	0.0
72	80	480	–	0.3	162	1201.4	14.8	162	20.0	0.0
73	61	364	2	0.0	2	465.1	0.0	2	0.8	0.0
74	31	132	0	0.0	0	0.3	0.0	0	0.0	0.0
75	38	161	0	0.0	0	0.5	0.0	0	0.0	0.0
\bar{x}	48.7	244.9	8.3	0.0	10.1	360.6	29.6	10.1	2.0	0.0

Tabelle 7.2: Ergebnisse für einen Planungshorizont von 3 Tagen. Enthalten ist die Anzahl der extra Roll-Ins (ER), die Laufzeit und der Optimalitätsgap. \bar{x} ist das arithmetische Mittel für alle zulässige Instanzen. Ein – bedeutet, dass keine zulässige Lösung gefunden wurde.

	<i>Instanz</i>		<i>Heuristik++</i>		<i>DF</i>			<i>EF</i>		
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	Gap (%)	ER (#)	Zeit (s)	Gap (%)
1	65	257	0	0.1	0	1.1	0.0	0	0.1	0.0
2	107	581	57	0.1	52	1202.6	100.0	44	57.6	0.0
3	35	148	0	0.0	0	0.4	0.0	0	0.0	0.0
4	88	340	0	0.1	0	1.8	0.0	0	0.2	0.0
5	14	46	0	0.0	0	0.1	0.0	0	0.0	0.0
6	55	170	0	0.0	0	0.7	0.0	0	0.1	0.0
7	44	170	0	0.0	0	0.7	0.0	0	0.1	0.0

Tabelle 7.2: Ergebnisse für einen Planungshorizont von 3 Tagen (Fortsetzung von der vorhergehenden Seite).

	<i>Instance</i>		<i>Heuristik++</i>		<i>DF</i>			<i>EF</i>		
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	Gap (%)	ER (#)	Zeit (s)	Gap (%)
8	51	155	0	0.0	0	0.7	0.0	0	0.1	0.0
9	70	268	0	0.1	0	1.2	0.0	0	0.1	0.0
10	44	155	0	0.0	0	0.5	0.0	0	0.1	0.0
11	113	575	62	0.2	58	1203.1	100.0	58	53.1	0.0
12	57	282	0	0.0	0	0.9	0.0	0	0.1	0.0
13	91	485	22	0.1	20	1201.8	100.0	20	51.1	0.0
14	80	440	6	0.1	3	1204.2	100.0	3	6.6	0.0
15	63	259	0	0.0	0	1.1	0.0	0	0.1	0.0
16	73	308	0	0.0	0	1.2	0.0	0	0.1	0.0
17	53	229	0	0.0	0	0.7	0.0	0	0.1	0.0
18	97	485	5	0.1	5	1205.9	100.0	5	3.8	0.0
19	66	296	0	0.0	0	0.9	0.0	0	0.1	0.0
20	78	440	14	0.1	12	1201.4	100.0	12	2.9	0.0
21	75	371	0	0.1	0	1.3	0.0	0	0.1	0.0
22	59	251	0	0.0	0	1.1	0.0	0	0.1	0.0
23	106	594	63	0.1	48	1202.3	100.0	47	43.9	0.0
24	43	200	0	0.0	0	0.5	0.0	0	0.1	0.0
25	108	538	4	0.1	3	1239.9	100.0	3	27.9	0.0
26	74	327	0	0.1	0	1.2	0.0	0	0.1	0.0
27	82	467	2	0.1	2	1208.1	100.0	2	1.4	0.0
28	72	413	–	0.2	75	1.8	0.0	75	3.4	0.0
29	62	259	0	0.0	0	1.4	0.0	0	0.1	0.0
30	103	572	30	0.1	28	1201.9	100.0	27	62.4	0.0
31	50	234	0	0.0	0	0.6	0.0	0	0.1	0.0
32	118	715	93	0.1	76	1202.8	100.0	68	95.4	0.0
33	78	420	6	0.1	1	13.2	100.0	0	18.3	0.0
34	88	492	47	0.1	38	1203.1	100.0	38	6.2	0.0
35	84	503	22	0.1	11	1209.8	100.0	11	9.5	0.0
36	65	288	0	0.0	0	1.1	0.0	0	0.1	0.0
37	94	546	195	0.2	68	1203.5	100.0	61	19.5	0.0
38	51	268	0	0.0	0	0.7	0.0	0	0.1	0.0
39	111	640	66	0.1	58	1202.4	100.0	57	310.8	0.0
40	59	269	0	0.0	0	0.7	0.0	0	0.1	0.0
41	95	513	–	0.3	154	1201.9	13.6	154	28.3	0.0
42	86	514	19	0.1	18	1201.6	100.0	18	16.7	0.0
43	70	319	6	0.1	6	1201.8	100.0	6	1.3	0.0

Tabelle 7.2: Ergebnisse für einen Planungshorizont von 3 Tagen (Fortsetzung von der vorhergehenden Seite).

	<i>Instance</i>		<i>Heuristik++</i>		<i>DF</i>			<i>EF</i>		
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	Gap (%)	ER (#)	Zeit (s)	Gap (%)
44	84	503	13	0.1	12	1201.5	100.0	12	25.2	0.0
45	5	9	0	0.0	0	0.1	0.0	0	0.0	0.0
46	91	470	19	0.1	19	1201.7	100.0	19	4.9	0.0
47	71	391	0	0.0	0	1.0	0.0	0	0.1	0.0
48	94	533	–	0.3	170	1201.7	18.8	168	20.7	0.0
49	74	413	4	0.1	4	1204.2	100.0	4	2.1	0.0
50	56	244	0	0.1	0	0.9	0.0	0	0.1	0.0
\bar{x}	73.0	367.3	16.1	0.1	18.8	530.9	42.6	18.2	17.5	0.0

Tabelle 7.3: Ergebnisse für einen Planungshorizont von 4 Tagen. Enthalten ist die Anzahl der extra Roll-Ins (ER), die Laufzeit und der Optimalitätsgap. \bar{x} ist das arithmetische Mittel für alle zulässige Instanzen. Ein – bedeutet, dass keine zulässige Lösung gefunden wurde.

	<i>Instanz</i>		<i>Heuristik++</i>		<i>DF</i>			<i>EF</i>		
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	Gap (%)	ER (#)	Zeit (s)	Gap (%)
1	105	470	40	0.2	36	1203.1	100.0	36	59.9	0.0
2	97	500	9	0.1	5	1206.9	100.0	5	7.8	0.0
3	93	356	0	0.1	0	1.8	0.0	0	0.2	0.0
4	21	55	0	0.0	0	0.2	0.0	0	0.0	0.0
5	91	330	0	0.1	0	1.5	0.0	0	0.2	0.0
6	52	156	0	0.0	0	0.7	0.0	0	0.1	0.0
7	86	332	0	0.1	0	1.4	0.0	0	0.1	0.0
8	105	457	29	0.2	28	1202.1	100.0	28	79.3	0.0
9	93	491	12	0.1	11	1201.5	100.0	11	25.8	0.0
10	129	700	55	0.2	53	1208.2	100.0	51	362.1	0.0
11	71	308	0	0.1	0	1.0	0.0	0	0.1	0.0
12	107	484	0	0.1	0	2.0	0.0	0	0.2	0.0
13	75	333	0	0.0	0	1.0	0.0	0	0.1	0.0
14	123	602	3	0.1	3	1202.8	100.0	3	7.2	0.0
15	96	515	14	0.1	12	1201.7	100.0	12	66.9	0.0
16	85	404	0	0.1	0	1.4	0.0	0	0.1	0.0
17	127	666	–	0.3	117	1203.2	100.0	85	280.1	0.0
18	71	346	0	0.1	0	1.0	0.0	0	0.1	0.0
19	145	721	18	0.2	15	1203.1	100.0	14	355.0	0.0
20	83	381	0	0.1	0	1.4	0.0	0	0.1	0.0
21	108	643	163	0.3	81	1201.7	7.4	81	35.9	0.0
22	103	490	35	0.1	33	1202.3	100.0	33	80.6	0.0
23	96	507	2	0.1	2	12.6	100.0	2	3.7	0.0
24	134	783	109	0.2	85	1202.8	100.0	73	90.8	0.0
25	92	483	6	0.1	1	17.6	100.0	1	18.7	0.0
26	142	834	88	0.2	82	1203.0	100.0	75	138.9	0.0
27	81	386	0	0.1	0	1.2	0.0	0	0.1	0.0
28	109	652	182	0.2	74	1202.4	100.0	62	263.1	0.0
29	102	534	5	0.1	5	1205.6	100.0	5	5.8	0.0
30	104	537	23	0.1	17	1201.8	100.0	17	48.2	0.0
31	127	710	–	0.4	189	1202.4	29.6	180	139.6	0.0
32	83	428	0	0.1	0	1.4	0.0	0	0.1	0.0
33	125	711	56	0.1	55	1202.5	100.0	52	357.7	0.0
34	6	10	0	0.0	0	0.1	0.0	0	0.0	0.0
35	143	794	30	0.1	22	1201.8	100.0	22	335.8	0.0
36	112	599	–	0.3	172	1202.0	19.8	168	59.1	0.0
37	92	496	4	0.1	4	1204.6	100.0	4	4.0	0.0
\bar{x}	97.7	492.0	26.0	0.1	79.8	684.1	55.6	27.6	76.4	0.0

7.1 Ergebnisse der Heuristik++, (DF) und (EF)

Tabelle 7.4: Ergebnisse für einen Planungshorizont von 5 Tagen. Enthalten ist die Anzahl der extra Roll-Ins (ER), die Laufzeit und der Optimalitätsgap. \bar{x} ist das arithmetische Mittel für alle zulässige Instanzen. Ein – bedeutet, dass keine zulässige Lösung gefunden wurde.

	<i>Instanz</i>		<i>Heuristik++</i>		<i>DF</i>			<i>EF</i>		
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	Gap (%)	ER (#)	Zeit (s)	Gap (%)
1	139	646	112	0.3	77	1204.4	100.0	68	128.6	0.0
2	98	445	0	0.1	0	1.4	0.0	0	0.2	0.0
3	72	281	0	0.1	0	1.3	0.0	0	0.1	0.0
4	98	339	0	0.1	0	1.6	0.0	0	0.2	0.0
5	80	263	0	0.1	0	1.2	0.0	0	0.1	0.0
6	86	316	0	0.1	0	1.2	0.0	0	0.2	0.0
7	157	816	72	0.3	68	1202.9	100.0	68	368.9	0.0
8	142	741	55	0.2	52	1203.6	100.0	51	116.5	0.0
9	105	484	0	0.1	0	1.6	0.0	0	0.2	0.0
10	101	450	0	0.1	0	1.5	0.0	0	0.2	0.0
11	142	636	6	0.1	6	1211.9	100.0	6	16.8	0.0
12	124	672	13	0.1	12	1202.0	100.0	12	95.2	0.0
13	96	433	0	0.1	0	1.6	0.0	0	0.1	0.0
14	164	874	–	0.4	–	1200.0	–	104	293.1	0.0
15	131	647	5	0.1	3	1202.0	100.0	3	328.7	0.0
16	120	564	0	0.1	0	2.1	0.0	0	0.3	0.0
17	110	645	163	0.3	81	1202.1	7.4	81	60.2	0.0
18	163	829	65	0.2	65	1203.8	100.0	61	362.6	0.0
19	132	702	56	0.1	36	1203.0	100.0	35	123.4	0.0
20	128	730	40	0.1	37	1202.4	100.0	35	274.9	0.0
21	158	932	88	0.3	82	1204.2	100.0	76	223.0	0.0
22	125	635	28	0.1	27	1202.3	100.0	27	93.3	0.0
23	116	624	0	0.1	0	2.0	0.0	0	0.2	0.0
24	139	752	66	0.1	62	1203.2	100.0	61	506.1	0.0
25	158	884	–	0.6	–	1200.0	–	204	377.0	0.0
26	132	699	53	0.1	51	1204.0	100.0	51	393.9	0.0
27	50	275	0	0.0	0	0.7	0.0	0	0.1	0.0
28	144	795	30	0.1	22	1201.9	100.0	22	739.8	0.0
29	146	808	–	0.4	–	1200.0	–	204	540.8	0.0
30	96	448	0	0.1	0	1.4	0.0	0	0.2	0.0
\bar{x}	121.7	612.2	31.6	0.2	25.2	722.4	52.1	39.0	168.2	0.0

7.2 Ergebnisse der (EF) unter Anwendung der Beschleunigungstechniken

Tabelle 7.5: Ergebnisse für einen Planungshorizont von 2 Tagen unter Anwendung von Early Branching. Enthalten ist die Anzahl der extra Roll-Ins (ER) und die Laufzeit. \bar{x} ist das arithmetische Mittel für alle zulässige Instanzen. Ein – bedeutet, dass keine zulässige Lösung gefunden wurde.

	<i>Instanz</i>		<i>E.B.($\epsilon = 0.0$)</i>		<i>E.B.($\epsilon = 0.01$)</i>		<i>E.B.($\epsilon = 0.03$)</i>		<i>E.B.($\epsilon = 0.05$)</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
1	32	90	0	0.0	0	0.0	0	0.0	0	0.0
2	73	380	35	9.0	35	12.4	35	12.0	35	9.2
3	67	368	5	3.5	5	3.9	5	3.7	5	4.6
4	30	132	0	0.0	0	0.0	0	0.0	0	0.0
5	35	121	0	0.0	0	0.0	0	0.0	0	0.0
6	58	235	0	0.1	0	0.1	0	0.1	0	0.1
7	14	46	0	0.0	0	0.0	0	0.0	0	0.0
8	7	9	0	0.0	0	0.0	0	0.0	0	0.0
9	48	161	0	0.1	0	0.1	0	0.1	0	0.1
10	43	169	0	0.1	0	0.1	0	0.1	0	0.1
11	1	1	0	0.0	0	0.0	0	0.0	0	0.0
12	51	155	0	0.1	0	0.1	0	0.1	0	0.1
13	61	236	0	0.1	0	0.1	0	0.1	0	0.1
14	25	96	0	0.0	0	0.0	0	0.0	0	0.0
15	28	91	0	0.0	0	0.0	0	0.0	0	0.0
16	77	366	20	3.5	20	3.5	20	3.5	20	3.5
17	59	344	11	1.3	11	1.3	11	1.3	11	1.3
18	34	147	0	0.0	0	0.0	0	0.0	0	0.0
19	51	250	2	0.3	2	0.3	2	0.3	2	0.3
20	78	450	28	5.9	28	5.9	28	5.9	28	5.9
21	42	225	0	0.1	0	0.1	0	0.0	0	0.0
22	29	83	0	0.0	0	0.0	0	0.0	0	0.0
23	56	262	0	0.1	0	0.1	0	0.1	0	0.1
24	51	222	0	0.1	0	0.1	0	0.1	0	0.1
25	28	142	0	0.0	0	0.0	0	0.0	0	0.0
26	47	191	0	0.1	0	0.1	0	0.1	0	0.1
27	75	381	3	1.7	3	1.7	3	1.7	3	1.7
28	48	221	0	0.1	0	0.1	0	0.1	0	0.1
29	26	117	0	0.0	0	0.0	0	0.0	0	0.0

7.2 Ergebnisse der (EF) unter Anwendung der Beschleunigungstechniken

Tabelle 7.5: Ergebnisse für einen Planungshorizont von 2 Tagen (Fortsetzung von der vorhergehenden Seite).

	<i>Instanz</i>		<i>E.B.($\epsilon = 0.00$)</i>		<i>E.B.($\epsilon = 0.01$)</i>		<i>E.B.($\epsilon = 0.03$)</i>		<i>E.B.($\epsilon = 0.05$)</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
30	70	398	12	1.9	12	1.9	12	1.9	12	1.9
31	59	307	0	0.1	0	0.1	0	0.1	0	0.1
32	26	97	0	0.0	0	0.0	0	0.0	0	0.0
33	49	218	0	0.4	0	0.4	0	0.4	0	0.4
34	78	448	28	2.6	28	2.6	28	2.6	28	2.6
35	48	237	0	0.1	0	0.1	0	0.1	0	0.1
36	23	109	0	0.0	0	0.0	0	0.0	0	0.0
37	71	336	0	0.1	0	0.1	0	0.1	0	0.1
38	74	385	11	6.1	11	6.1	11	6.1	11	6.1
39	37	144	0	0.0	0	0.0	0	0.0	0	0.0
40	46	237	0	0.1	0	0.1	0	0.1	0	0.1
41	69	432	77	2.1	77	2.1	77	2.1	77	2.1
42	39	211	0	0.0	0	0.0	0	0.0	0	0.0
43	24	79	0	0.0	0	0.0	0	0.0	0	0.0
44	79	411	29	7.7	29	10.6	29	5.0	29	9.9
45	62	341	2	0.7	2	0.7	2	0.7	2	0.7
46	34	166	0	0.0	0	0.0	0	0.0	0	0.0
47	60	282	4	0.7	4	0.7	4	0.7	4	0.7
48	74	501	24	4.6	24	4.6	24	4.6	24	4.6
49	57	331	1	0.7	1	0.7	1	0.7	1	0.7
50	35	152	0	0.0	0	0.0	0	0.0	0	0.0
51	74	429	24	4.2	24	3.7	24	5.6	24	5.1
52	68	405	11	3.6	11	3.6	11	3.6	11	3.6
53	36	175	0	0.0	0	0.0	0	0.0	0	0.0
54	45	211	0	0.0	0	0.0	0	0.0	0	0.0
55	60	347	27	1.9	27	1.9	27	1.9	27	1.9
56	49	305	0	0.0	0	0.0	0	0.0	0	0.0
57	36	162	0	0.0	0	0.0	0	0.0	0	0.0
58	66	372	5	1.2	5	1.2	5	1.2	5	1.2
59	65	384	17	1.9	17	1.9	17	1.9	17	1.9
60	39	153	0	0.1	0	0.1	0	0.1	0	0.1
61	51	270	134	1.7	134	1.7	134	1.7	134	1.7
62	76	440	26	3.3	26	3.3	26	3.3	26	3.3
63	54	317	0	0.1	0	0.1	0	0.1	0	0.1
64	29	111	0	0.0	0	0.0	0	0.0	0	0.0
65	80	445	40	15.2	40	12.5	40	7.1	40	15.5

Tabelle 7.5: Ergebnisse für einen Planungshorizont von 2 Tagen (Fortsetzung von der vorhergehenden Seite).

	<i>Instanz</i>		<i>E.B.($\epsilon = 0.00$)</i>		<i>E.B.($\epsilon = 0.01$)</i>		<i>E.B.($\epsilon = 0.03$)</i>		<i>E.B.($\epsilon = 0.05$)</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
66	45	266	0	0.1	0	0.1	0	0.1	0	0.1
67	3	5	0	0.0	0	0.0	0	0.0	0	0.0
68	3	5	0	0.0	0	0.0	0	0.0	0	0.0
69	90	469	19	5.0	19	5.0	19	5.0	19	5.0
70	53	325	0	0.1	0	0.1	0	0.1	0	0.1
71	32	119	0	0.0	0	0.0	0	0.0	0	0.0
72	80	480	162	11.3	162	16.4	162	16.4	162	16.4
73	61	364	2	0.8	2	0.8	2	0.8	2	0.8
74	31	132	0	0.0	0	0.0	0	0.0	0	0.0
75	38	161	0	0.0	0	0.0	0	0.0	0	0.0
\bar{x}	48.7	244.9	10.1	1.4	10.1	1.5	10.1	1.4	10.1	1.5

Tabelle 7.6: Ergebnisse für einen Planungshorizont von 3 Tagen unter Anwendung von Early Branching. Enthalten ist die Anzahl der extra Roll-Ins (ER) und die Laufzeit. \bar{x} ist das arithmetische Mittel für alle zulässige Instanzen. Ein – bedeutet, dass keine zulässige Lösung gefunden wurde.

	<i>Instanz</i>		<i>E.B.($\epsilon = 0.00$)</i>		<i>E.B.($\epsilon = 0.01$)</i>		<i>E.B.($\epsilon = 0.03$)</i>		<i>E.B.($\epsilon = 0.05$)</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
1	65	257	0	0.1	0	0.1	0	0.1	0	0.1
2	107	581	44	59.1	44	164.7	44	123.7	44	135.2
3	35	148	0	0.0	0	0.0	0	0.0	0	0.0
4	88	340	0	0.2	0	0.2	0	0.2	0	0.2
5	14	46	0	0.0	0	0.0	0	0.0	0	0.0
6	55	170	0	0.1	0	0.1	0	0.0	0	0.1
7	44	170	0	0.0	0	0.0	0	0.0	0	0.0
8	51	155	0	0.1	0	0.1	0	0.1	0	0.1
9	70	268	0	0.1	0	0.1	0	0.1	0	0.1
10	44	155	0	0.1	0	0.1	0	0.1	0	0.1
11	113	575	58	77.0	58	316.2	58	314.2	58	476.5
12	57	282	0	0.1	0	0.1	0	0.1	0	0.1
13	91	485	20	30.2	20	17.2	20	21.7	20	29.3

7.2 Ergebnisse der (EF) unter Anwendung der Beschleunigungstechniken

Tabelle 7.6: Ergebnisse für einen Planungshorizont von 3 Tagen (Fortsetzung von der vorhergehenden Seite).

	<i>Instance</i>		<i>E.B.($\epsilon = 0.00$)</i>		<i>E.B.($\epsilon = 0.01$)</i>		<i>E.B.($\epsilon = 0.03$)</i>		<i>E.B.($\epsilon = 0.05$)</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
14	80	440	3	6.7	3	6.3	3	6.3	3	6.3
15	63	259	0	0.1	0	0.1	0	0.1	0	0.1
16	73	308	0	0.1	0	0.1	0	0.1	0	0.1
17	53	229	0	0.1	0	0.1	0	0.1	0	0.1
18	97	485	5	3.9	5	3.9	5	3.9	5	3.9
19	66	296	0	0.1	0	0.1	0	0.1	0	0.1
20	78	440	12	3.0	12	3.0	12	3.0	12	3.0
21	75	371	0	0.1	0	0.1	0	0.1	0	0.1
22	59	251	0	0.1	0	0.1	0	0.1	0	0.1
23	106	594	47	26.3	47	118.5	47	128.7	47	172.2
24	43	200	0	0.1	0	0.1	0	0.1	0	0.1
25	108	538	3	29.0	3	21.8	3	16.6	3	21.7
26	74	327	0	0.1	0	0.1	0	0.1	0	0.1
27	82	467	2	1.4	2	1.4	2	1.4	2	1.4
28	72	413	75	3.4	75	3.4	75	3.4	75	3.4
29	62	259	0	0.1	0	0.1	0	0.1	0	0.1
30	103	572	27	65.4	27	50.2	27	214.0	27	154.6
31	50	234	0	0.1	0	0.1	0	0.1	0	0.1
32	118	715	68	53.7	68	444.2	68	542.5	68	936.7
33	78	420	0	6.6	0	3.8	0	3.8	0	3.8
34	88	492	38	6.3	38	6.3	38	6.3	38	6.3
35	84	503	11	8.7	11	9.1	11	9.2	11	11.9
36	65	288	0	0.1	0	0.1	0	0.1	0	0.1
37	94	546	61	21.5	61	19.9	61	29.0	61	20.4
38	51	268	0	0.1	0	0.1	0	0.1	0	0.1
39	111	640	57	153.6	57	208.0	57	195.8	57	184.0
40	59	269	0	0.1	0	0.1	0	0.1	0	0.1
41	95	513	154	18.6	154	31.6	154	24.3	154	34.4
42	86	514	18	6.3	18	5.1	18	5.2	18	5.4
43	70	319	6	1.3	6	1.3	6	1.3	6	1.3
44	84	503	12	9.4	12	12.8	12	8.8	12	6.4
45	5	9	0	0.0	0	0.0	0	0.0	0	0.0
46	91	470	19	4.9	19	4.9	19	4.9	19	4.9
47	71	391	0	0.1	0	0.1	0	0.1	0	0.1
48	94	533	168	14.5	168	40.0	168	60.0	168	57.3
49	74	413	4	2.1	4	2.1	4	2.1	4	2.1

Tabelle 7.6: Ergebnisse für einen Planungshorizont von 3 Tagen (Fortsetzung von der vorhergehenden Seite).

	<i>Instance</i>		<i>E.B.($\epsilon = 0.00$)</i>		<i>E.B.($\epsilon = 0.01$)</i>		<i>E.B.($\epsilon = 0.03$)</i>		<i>E.B.($\epsilon = 0.05$)</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
50	56	244	0	0.1	0	0.1	0	0.1	0	0.1
\bar{x}	73.0	367.3	18.2	12.3	18.2	30.0	18.2	34.6	18.2	45.7

7.2 Ergebnisse der (EF) unter Anwendung der Beschleunigungstechniken

Tabelle 7.7: Ergebnisse für einen Planungshorizont von 4 Tagen unter Anwendung von Early Branching.. Enthalten ist die Anzahl der extra Roll-Ins (ER) und die Laufzeit. \bar{x} ist das arithmetische Mittel für alle zulässige Instanzen. Ein – bedeutet, dass keine zulässige Lösung gefunden wurde.

	<i>Instanz</i>		<i>E.B.($\epsilon = 0.00$)</i>		<i>E.B.($\epsilon = 0.01$)</i>		<i>E.B.($\epsilon = 0.03$)</i>		<i>E.B.($\epsilon = 0.05$)</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER	Zeit
1	105	470	36	38.7	36	88.2	36	93.4	36	140.4
2	97	500	5	6.2	5	7.7	5	7.7	5	7.7
3	93	356	0	0.2	0	0.2	0	0.2	0	0.2
4	21	55	0	0.0	0	0.0	0	0.0	0	0.0
5	91	330	0	0.2	0	0.2	0	0.2	0	0.2
6	52	156	0	0.1	0	0.1	0	0.1	0	0.1
7	86	332	0	0.1	0	0.1	0	0.1	0	0.1
8	105	457	28	59.1	28	34.1	28	30.2	28	41.1
9	93	491	11	17.7	11	14.0	11	33.6	11	20.4
10	129	700	51	107.9	51	245.2	51	404.9	51	606.8
11	71	308	0	0.1	0	0.1	0	0.1	0	0.1
12	107	484	0	0.2	0	0.2	0	0.2	0	0.2
13	75	333	0	0.1	0	0.1	0	0.1	0	0.1
14	123	602	3	7.3	3	7.3	3	7.3	3	7.3
15	96	515	12	10.1	12	11.4	12	10.2	12	10.2
16	85	404	0	0.1	0	0.1	0	0.1	0	0.1
17	127	666	85	110.4	85	1159.3	85	733.8	85	798.4
18	71	346	0	0.1	0	0.1	0	0.1	0	0.1
19	145	721	14	266.8	14	68.5	14	78.2	14	55.3
20	83	381	0	0.2	0	0.2	0	0.2	0	0.2
21	108	643	81	34.3	81	356.5	81	609.2	81	89.0
22	103	490	33	36.6	33	66.1	33	13.3	33	33.5
23	96	507	2	3.7	2	3.7	2	3.7	2	3.7
24	134	783	73	74.8	73	70.9	73	648.6	73	77.8
25	92	483	1	6.7	1	6.7	1	6.7	1	14.3
26	142	834	75	306.2	77	1200.0	88	1200.0	88	1200.0
27	81	386	0	0.1	0	0.1	0	0.1	0	0.1
28	109	652	62	44.9	62	55.3	62	112.6	62	92.4
29	102	534	5	5.8	5	5.8	5	5.8	5	5.8
30	104	537	17	19.6	17	13.9	17	19.0	17	27.8
31	127	710	180	373.7	230	1200.0	230	1200.0	230	1200.0
32	83	428	0	0.1	0	0.1	0	0.1	0	0.1
33	125	711	52	332.8	52	340.3	52	390.8	52	480.1
34	6	10	0	0.0	0	0.0	0	0.0	0	0.0
35	143	794	22	512.9	22	1070.6	22	900.3	22	762.1
36	112	599	168	41.7	168	350.1	168	200.4	168	214.6
37	92	496	4	4.1	4	4.1	4	4.1	4	4.1
\bar{x}	97.7	492.0	27.6	65.5	29.0	172.5	29.3	181.5	29.3	159.3

Tabelle 7.8: Ergebnisse für einen Planungshorizont von 5 Tagen unter Anwendung von Early Branching. Enthalten ist die Anzahl der extra Roll-Ins (ER) und die Laufzeit. \bar{x} ist das arithmetische Mittel für alle zulässige Instanzen. Ein – bedeutet, dass keine zulässige Lösung gefunden wurde.

	<i>Instanz</i>		<i>E.B.($\epsilon = 0.00$)</i>		<i>E.B.($\epsilon = 0.01$)</i>		<i>E.B.($\epsilon = 0.03$)</i>		<i>E.B.($\epsilon = 0.05$)</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
1	139	646	68	485.8	112	1200.0	122	1200.0	112	1200.0
2	98	445	0	0.2	0	0.2	0	0.2	0	0.2
3	72	281	0	0.1	0	0.1	0	0.1	0	0.1
4	98	339	0	0.2	0	0.2	0	0.2	0	0.2
5	80	263	0	0.1	0	0.1	0	0.1	0	0.1
6	86	316	0	0.2	0	0.2	0	0.2	0	0.2
7	157	816	68	190.6	72	1200.0	72	1200.0	72	1200.0
8	142	741	51	139.5	51	676.6	51	861.2	51	828.5
9	105	484	0	0.2	0	0.2	0	0.2	0	0.2
10	101	450	0	0.2	0	0.2	0	0.2	0	0.2
11	142	636	6	16.8	6	16.8	6	16.8	6	16.8
12	124	672	12	86.7	12	39.9	12	56.7	12	32.1
13	96	433	0	0.1	0	0.1	0	0.1	0	0.1
14	164	874	104	194.3	–	1200.0	113	1200.1	–	1200.0
15	131	647	3	161.2	5	1200.0	5	1200.0	3	835.8
16	120	564	0	0.3	0	0.3	0	0.3	0	0.3
17	110	645	81	47.1	81	195.3	81	87.8	81	486.0
18	163	829	61	590.3	65	1200.0	65	1200.0	65	1200.0
19	132	702	35	40.3	35	768.9	35	419.6	35	598.9
20	128	730	35	250.6	35	168.4	35	370.0	35	134.4
21	158	932	76	390.8	88	1200.0	88	1200.0	88	1200.0
22	125	635	27	230.2	27	39.1	27	23.3	27	23.5
23	116	624	0	0.2	0	0.2	0	0.2	0	0.2
24	139	752	61	283.4	61	134.9	61	118.7	61	1335.5
25	158	884	204	377.0	209	1200.0	314	1200.0	–	1200.0
26	132	699	51	152.2	51	38.04	51	40.2	51	169.9
27	50	275	0	0.1	0	0.1	0	0.1	0	0.1
28	144	795	22	454.0	22	648.9	22	1180.3	22	1043.8
29	146	808	204	193.52	279	1200.0	–	1200.0	–	1200.0
30	96	448	0	0.2	0	0.2	0	0.2	0	0.2
\bar{x}	121.7	612.2	39.0	142.9	40.4	411.0	39.0	425.0	24.0	424.0

7.2 Ergebnisse der (EF) unter Anwendung der Beschleunigungstechniken

Tabelle 7.9: Ergebnisse für einen Planungshorizont von 2 Tagen unter Anwendung von Early Termination. Enthalten ist die Anzahl der extra Roll-Ins (ER) und die Laufzeit. \bar{x} ist das arithmetische Mittel für alle zulässige Instanzen. Ein – bedeutet, dass keine zulässige Lösung gefunden wurde.

	<i>Instanz</i>		<i>Early Termination</i>		<i>EF</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
1	32	90	0	0.0	0	0.0
2	73	380	35	17.8	35	19.5
3	67	368	5	5.6	5	7.6
4	30	132	0	0.0	0	0.0
5	35	121	0	0.0	0	0.0
6	58	235	0	0.1	0	0.1
7	14	46	0	0.0	0	0.0
8	7	9	0	0.0	0	0.0
9	48	161	0	0.1	0	0.1
10	43	169	0	0.0	0	0.1
11	1	1	0	0.0	0	0.0
12	51	155	0	0.1	0	0.1
13	61	236	0	0.1	0	0.1
14	25	96	0	0.0	0	0.0
15	28	91	0	0.0	0	0.0
16	77	366	20	3.4	20	3.4
17	59	344	11	4.4	11	3.5
18	34	147	0	0.0	0	0.0
19	51	250	2	0.3	2	0.3
20	78	450	28	12.7	28	10.5
21	42	225	0	0.0	0	0.0
22	29	83	0	0.0	0	0.0
23	56	262	0	0.1	0	0.1
24	51	222	0	0.1	0	0.1
25	28	142	0	0.0	0	0.0
26	47	191	0	0.1	0	0.1
27	75	381	3	1.7	3	1.7
28	48	221	0	0.1	0	0.1
29	26	117	0	0.0	0	0.0
30	70	398	12	1.9	12	1.9
31	59	307	0	0.1	0	0.1
32	26	97	0	0.0	0	0.0
33	49	218	0	0.4	0	0.4
34	78	448	28	2.5	28	2.5

Tabelle 7.9: Ergebnisse für einen Planungshorizont von 2 Tagen (Fortsetzung von der vorhergehenden Seite).

	<i>Instanz</i>		<i>Early Termination</i>		<i>EF</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
35	48	237	0	0.1	0	0.1
36	23	109	0	0.0	0	0.0
37	71	336	0	0.1	0	0.1
38	74	385	11	13.3	11	11.7
39	37	144	0	0.0	0	0.0
40	46	237	0	0.1	0	0.1
41	69	432	77	2.0	77	2.0
42	39	211	0	0.0	0	0.0
43	24	79	0	0.0	0	0.0
44	79	411	29	13.4	29	11.6
45	62	341	2	0.7	2	0.7
46	34	166	0	0.0	0	0.0
47	60	282	4	0.7	4	0.7
48	74	501	24	7.0	24	11.1
49	57	331	1	0.7	1	0.7
50	35	152	0	0.0	0	0.0
51	74	429	24	5.0	24	5.2
52	68	405	11	6.3	11	3.9
53	36	175	0	0.0	0	0.0
54	45	211	0	0.0	0	0.0
55	60	347	27	3.4	27	3.7
56	49	305	0	0.0	0	0.0
57	36	162	0	0.0	0	0.0
58	66	372	5	1.2	5	1.2
59	65	384	17	1.9	17	1.9
60	39	153	0	0.1	0	0.1
61	51	270	134	1.8	134	1.7
62	76	440	26	3.2	26	3.3
63	54	317	0	0.1	0	0.1
64	29	111	0	0.0	0	0.0
65	80	445	40	10.8	40	14.7
66	45	266	0	0.1	0	0.1
67	3	5	0	0.0	0	0.0
68	3	5	0	0.0	0	0.0
69	90	469	19	4.9	19	4.9
70	53	325	0	0.1	0	0.1

7.2 Ergebnisse der (EF) unter Anwendung der Beschleunigungstechniken

Tabelle 7.9: Ergebnisse für einen Planungshorizont von 2 Tagen (Fortsetzung von der vorhergehenden Seite).

	<i>Instanz</i>		<i>Early Termination</i>		<i>EF</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
71	32	119	0	0.0	0	0.0
72	80	480	162	21.6	162	20.0
73	61	364	2	0.8	2	0.8
74	31	132	0	0.0	0	0.0
75	38	161	0	0.0	0	0.0
\bar{x}	48.7	244.9	10.1	2.0	10.1	2.0

Tabelle 7.10: Ergebnisse für einen Planungshorizont von 3 Tagen unter Anwendung von Early Termination. Enthalten ist die Anzahl der extra Roll-Ins (ER) und die Laufzeit. \bar{x} ist das arithmetische Mittel für alle zulässige Instanzen. Ein – bedeutet, dass keine zulässige Lösung gefunden wurde.

	<i>Instanz</i>		<i>Early Termination</i>		<i>EF</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
1	65	257	0	0.1	0	0.1
2	107	581	44	59.6	44	57.6
3	35	148	0	0.0	0	0.0
4	88	340	0	0.2	0	0.2
5	14	46	0	0.0	0	0.0
6	55	170	0	0.1	0	0.1
7	44	170	0	0.1	0	0.1
8	51	155	0	0.1	0	0.1
9	70	268	0	0.1	0	0.1
10	44	155	0	0.1	0	0.1
11	113	575	58	174.6	58	53.1
12	57	282	0	0.1	0	0.1
13	91	485	20	68.8	20	51.1
14	80	440	3	6.3	3	6.6
15	63	259	0	0.1	0	0.1
16	73	308	0	0.1	0	0.1
17	53	229	0	0.1	0	0.1
18	97	485	5	3.8	5	3.8

Tabelle 7.10: Ergebnisse für einen Planungshorizont von 3 Tagen (Fortsetzung von der vorhergehenden Seite).

	<i>Instance</i>		<i>Early Termination</i>		<i>EF</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
19	66	296	0	0.1	0	0.1
20	78	440	12	2.9	12	2.9
21	75	371	0	0.1	0	0.1
22	59	251	0	0.1	0	0.1
23	106	594	47	41.6	47	43.9
24	43	200	0	0.1	0	0.1
25	108	538	3	36.6	3	27.9
26	74	327	0	0.1	0	0.1
27	82	467	2	1.4	2	1.4
28	72	413	75	3.3	75	3.4
29	62	259	0	0.1	0	0.1
30	103	572	27	413.8	27	62.4
31	50	234	0	0.1	0	0.1
32	118	715	68	61.1	68	95.4
33	78	420	0	16.9	0	18.3
34	88	492	38	6.2	38	6.2
35	84	503	11	19.9	11	9.5
36	65	288	0	0.1	0	0.1
37	94	546	61	37.1	61	19.5
38	51	268	0	0.1	0	0.1
39	111	640	58	394.7	57	310.8
40	59	269	0	0.1	0	0.1
41	95	513	154	20.9	154	28.3
42	86	514	18	14.1	18	16.7
43	70	319	6	1.3	6	1.3
44	84	503	12	20.0	12	25.2
45	5	9	0	0.0	0	0.0
46	91	470	19	4.9	19	4.9
47	71	391	0	0.1	0	0.1
48	94	533	168	27.1	168	20.7
49	74	413	4	2.1	4	2.1
50	56	244	0	0.1	0	0.1
\bar{x}	73.0	367.3	18.2	28.7	18.2	17.5

7.2 Ergebnisse der (EF) unter Anwendung der Beschleunigungstechniken

Tabelle 7.11: Ergebnisse für einen Planungshorizont von 4 Tagen unter Verwendung von Early Termination. Enthalten ist die Anzahl der extra Roll-Ins (ER) und die Laufzeit. \bar{x} ist das arithmetische Mittel für alle zulässige Instanzen. Ein – bedeutet, dass keine zulässige Lösung gefunden wurde.

	<i>Instanz</i>		<i>Early Termination</i>		<i>EF</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
1	105	470	36	45.2	36	59.9
2	97	500	5	7.6	5	7.8
3	93	356	0	0.2	0	0.2
4	21	55	0	0.0	0	0.0
5	91	330	0	0.2	0	0.2
6	52	156	0	0.1	0	0.1
7	86	332	0	0.1	0	0.1
8	105	457	28	91.2	28	79.3
9	93	491	11	18.4	11	25.8
10	129	700	55	150.4	51	362.1
11	71	308	0	0.1	0	0.1
12	107	484	0	0.2	0	0.2
13	75	333	0	0.1	0	0.1
14	123	602	3	7.2	3	7.2
15	96	515	12	23.7	12	66.9
16	85	404	0	0.1	0	0.1
17	127	666	85	117.2	85	280.1
18	71	346	0	0.1	0	0.1
19	145	721	14	603.7	14	355.0
20	83	381	0	0.1	0	0.1
21	108	643	81	35.1	81	35.9
22	103	490	33	41.1	33	80.6
23	96	507	2	3.7	2	3.7
24	134	783	73	157.2	73	90.8
25	92	483	1	37.4	1	18.7
26	142	834	75	177.6	75	138.9
27	81	386	0	0.1	0	0.1
28	109	652	62	48.9	62	263.1
29	102	534	5	5.8	5	5.8
30	104	537	17	32.4	17	48.2
31	127	710	180	76.9	180	139.6
32	83	428	0	0.1	0	0.1
33	125	711	56	234.2	52	357.7
34	6	10	0	0.0	0	0.0
35	143	794	22	566.8	22	335.8
36	112	599	168	104.7	168	59.1
37	92	496	4	4.0	4	4.0
\bar{x}	97.7	492.0	27.6	70.1	27.6	76.4

Tabelle 7.12: Ergebnisse für einen Planungshorizont von 5 Tagen unter Verwendung von Early Termination. Enthalten ist die Anzahl der extra Roll-Ins (ER) und die Laufzeit. \bar{x} ist das arithmetische Mittel für alle zulässige Instanzen. Ein – bedeutet, dass keine zulässige Lösung gefunden wurde.

	<i>Instanz</i>		<i>Early Termination</i>		<i>EF</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
1	139	646	68	145.8	68	128.6
2	98	445	0	0.2	0	0.2
3	72	281	0	0.1	0	0.1
4	98	339	0	0.2	0	0.2
5	80	263	0	0.1	0	0.1
6	86	316	0	0.2	0	0.2
7	157	816	68	415.8	68	368.9
8	142	741	51	109.0	51	116.5
9	105	484	0	0.2	0	0.2
10	101	450	0	0.2	0	0.2
11	142	636	6	17.4	6	16.8
12	124	672	12	457.5	12	95.2
13	96	433	0	0.1	0	0.1
14	164	874	104	285.4	104	293.1
15	131	647	3	91.5	3	328.7
16	120	564	0	0.3	0	0.3
17	110	645	81	39.5	81	60.2
18	163	829	61	337.9	61	362.6
19	132	702	35	695.5	35	123.4
20	128	730	35	452.5	35	274.9
21	158	932	88	152.6	76	223.0
22	125	635	27	242.2	27	93.3
23	116	624	0	0.2	0	0.2
24	139	752	66	876.8	61	506.1
25	158	884	204	193.3	204	377.0
26	132	699	51	543.2	51	393.9
27	50	275	0	0.1	0	0.1
28	144	795	22	542.2	22	739.8
29	146	808	204	161.8	204	540.8
30	96	448	0	0.2	0	0.2
\bar{x}	121.7	612.2	39.0	192.1	39.0	168.2

7.2 Ergebnisse der (EF) unter Anwendung der Beschleunigungstechniken

Tabelle 7.13: Ergebnisse für einen Planungshorizont von 2 Tagen unter Anwendung der Smoothing Methode. Enthalten ist die Anzahl der extra Roll-Ins (ER), die Laufzeit und die Anzahl der benötigten Iterationen. \bar{x} ist das arithmetische Mittel für alle zulässige Instanzen. Ein – bedeutet, dass keine zulässige Lösung gefunden wurde.

	<i>Instanz</i>		<i>Smoothing Methode</i>			<i>EF</i>		
	Züge (#)	Gr. (#)	ER (#)	Iter. (#)	Zeit (s)	ER (#)	Iter. (#)	Zeit (s)
1	32	90	0	1	0.1	0	1	0.0
2	73	380	35	54	13.8	35	86	19.5
3	67	368	5	48	8.1	5	65	7.6
4	30	132	0	1	0.1	0	1	0.0
5	35	121	0	1	0.1	0	1	0.0
6	58	235	0	1	0.2	0	1	0.1
7	14	46	0	1	0.0	0	1	0.0
8	7	9	0	1	0.0	0	1	0.0
9	48	161	0	1	0.1	0	1	0.1
10	43	169	0	1	0.1	0	1	0.1
11	1	1	0	1	0.0	0	1	0.0
12	51	155	0	1	0.1	0	1	0.1
13	61	236	0	1	0.2	0	1	0.1
14	25	96	0	1	0.0	0	1	0.0
15	28	91	0	1	0.0	0	1	0.0
16	77	366	20	37	9.2	20	37	3.4
17	59	344	11	23	3.1	11	41	3.5
18	34	147	0	1	0.1	0	1	0.0
19	51	250	2	9	0.8	2	9	0.3
20	78	450	28	52	14.2	28	62	10.5
21	42	225	0	1	0.1	0	1	0.0
22	29	83	0	1	0.0	0	1	0.0
23	56	262	0	1	0.1	0	1	0.1
24	51	222	0	1	0.1	0	1	0.1
25	28	142	0	1	0.0	0	1	0.0
26	47	191	0	1	0.1	0	1	0.1
27	75	381	3	23	4.8	3	23	1.7
28	48	221	0	1	0.0	0	1	0.1
29	26	117	0	1	0.0	0	1	0.1
30	70	398	12	27	5.0	12	27	1.9
31	59	307	0	1	0.1	0	1	0.1
32	26	97	0	1	0.0	0	1	0.0
33	49	218	0	1	0.4	0	1	0.4

Tabelle 7.13: Ergebnisse für einen Planungshorizont von 2 Tagen (Fortsetzung von der vorhergehenden Seite).

	<i>Instanz</i>		<i>Smoothing Methode</i>			<i>EF</i>		
	Züge (#)	Gr. (#)	ER (#)	Iter. (#)	Zeit (s)	ER (#)	Iter. (#)	Zeit (s)
34	78	448	28	29	6.5	28	29	2.5
35	48	237	0	1	0.1	0	1	0.1
36	23	109	0	1	0.0	0	1	0.0
37	71	336	0	1	0.1	0	1	0.1
38	74	385	11	57	13.9	11	76	11.7
39	37	144	0	1	0.0	0	1	0.0
40	46	237	0	1	0.1	0	1	0.1
41	69	432	77	33	5.2	77	33	2.0
42	39	211	0	1	0.0	0	1	0.0
43	24	79	0	1	0.0	0	1	0.0
44	79	411	29	80	20.7	29	57	11.6
45	62	341	2	14	1.8	2	14	0.7
46	34	166	0	1	0.0	0	1	0.0
47	60	282	4	15	1.9	4	15	0.7
48	74	501	24	46	10.6	24	78	11.1
49	57	331	1	17	1.8	1	17	0.7
50	35	152	0	1	0.1	0	1	0.0
51	74	429	24	40	8.9	24	38	5.2
52	68	405	11	53	9.3	11	47	3.9
53	36	175	0	1	0.1	0	1	0.0
54	45	211	0	1	0.1	0	1	0.0
55	60	347	27	36	5.2	27	34	3.7
56	49	305	0	1	0.1	0	1	0.0
57	36	162	0	1	0.1	0	1	0.0
58	66	372	5	19	3.1	5	19	1.2
59	65	384	17	33	4.9	17	33	1.9
60	39	153	0	1	0.1	0	1	0.1
61	51	270	134	43	4.5	134	43	1.7
62	76	440	26	41	8.5	26	41	3.3
63	54	317	0	1	0.1	0	1	0.1
64	29	111	0	1	0.1	0	1	0.0
65	80	445	40	88	28.6	40	76	14.7
66	45	266	0	1	0.1	0	1	0.1
67	3	5	0	1	0.0	0	1	0.0
68	3	5	0	1	0.0	0	1	0.0
69	90	469	19	39	12.5	19	39	4.9

7.2 Ergebnisse der (EF) unter Anwendung der Beschleunigungstechniken

Tabelle 7.13: Ergebnisse für einen Planungshorizont von 2 Tagen (Fortsetzung von der vorhergehenden Seite).

	<i>Instanz</i>		<i>Smoothing Methode</i>			<i>EF</i>		
	Züge (#)	Gr. (#)	ER (#)	Iter. (#)	Zeit (s)	ER (#)	Iter. (#)	Zeit (s)
70	53	325	0	1	0.1	0	1	0.1
71	32	119	0	1	0.1	0	1	0.0
72	80	480	162	67	24.7	162	69	20.0
73	61	364	2	16	2.0	2	16	0.8
74	31	132	0	1	0.1	0	1	0.0
75	38	161	0	1	0.1	0	1	0.0
\bar{x}	48.7	244.9	10.1	14.5	2.3	10.1	10.6	2.0

Tabelle 7.14: Ergebnisse für einen Planungshorizont von 3 Tagen unter Anwendung der Smoothing Methode. Enthalten ist die Anzahl der extra Roll-Ins (ER), die Laufzeit und die Anzahl der benötigten Iterationen. \bar{x} ist das arithmetische Mittel für alle zulässige Instanzen. Ein – bedeutet, dass keine zulässige Lösung gefunden wurde.

	<i>Instanz</i>		<i>Smoothing Methode</i>			<i>EF</i>		
	Züge (#)	Gr. (#)	ER (#)	Iter. (#)	Zeit. (s)	ER (#)	Iter. (#)	Zeit (s)
1	65	257	0	1	0.2	0	1	0.1
2	107	581	44	320	209.4	44	202	57.6
3	35	148	0	1	0.1	0	1	0.0
4	88	340	0	1	0.2	0	1	0.2
5	14	46	0	1	0.0	0	1	0.0
6	55	170	0	1	0.1	0	1	0.1
7	44	170	0	1	0.1	0	1	0.1
8	51	155	0	1	0.1	0	1	0.1
9	70	268	0	1	0.1	0	1	0.1
10	44	155	0	1	0.2	0	1	0.1
11	113	575	58	316	160.3	58	147	53.1
12	57	282	0	1	0.1	0	1	0.1
13	91	485	20	102	46.7	20	164	51.1
14	80	440	3	37	12.3	3	37	6.6
15	63	259	0	1	0.2	0	1	0.1
16	73	308	0	1	0.2	0	1	0.1

Tabelle 7.14: Ergebnisse für einen Planungshorizont von 3 Tagen (Fortsetzung von der vorhergehenden Seite).

	<i>Instanz</i>		<i>Smoothing Methode</i>			<i>EF</i>		
	Züge (#)	Gr. (#)	ER (#)	Zeit (#)	Iter. (s)	ER (#)	Iter. (#)	Zeit (s)
17	53	229	0	1	0.1	0	1	0.1
18	97	485	5	29	10.8	5	29	3.8
19	66	296	0	1	0.1	0	1	0.1
20	78	440	12	32	7.6	12	32	2.9
21	75	371	0	1	0.1	0	1	0.1
22	59	251	0	1	0.1	0	1	0.1
23	106	594	47	112	56.5	47	148	43.9
24	43	200	0	1	0.1	0	1	0.1
25	108	538	3	268	139.9	3	80	27.9
26	74	327	0	0	0.3	0	1	0.1
27	82	467	2	16	3.9	2	16	1.4
28	72	413	75	40	8.7	75	40	3.4
29	62	259	0	1	0.1	0	1	0.1
30	103	572	27	577	277.6	27	219	62.4
31	50	234	0	1	0.1	0	1	0.1
32	118	715	68	214	114.7	68	261	95.4
33	78	420	1	28	7.6	1	63	18.3
34	88	492	38	52	15.8	38	52	6.2
35	84	503	11	82	23.5	11	57	9.5
36	65	288	0	1	0.1	0	1	0.1
37	94	546	61	220	97.8	61	94	19.5
38	51	268	0	1	0.1	0	1	0.1
39	111	640	57	425	269.2	57	601	310.8
40	59	269	0	1	0.1	0	1	0.1
41	95	513	154	109	42.4	154	135	28.3
42	86	514	18	49	14.8	18	83	16.7
43	70	319	6	19	3.5	6	19	1.3
44	84	503	12	60	19.6	12	131	25.2
45	5	9	0	1	0.1	0	1	0.0
46	91	470	19	39	12.9	19	39	4.9
47	71	391	0	1	0.2	0	1	0.1
48	94	533	168	76	33.5	168	81	20.7
49	74	413	4	27	5.8	4	27	2.1
50	56	244	0	1	0.1	0	1	0.1
\bar{x}	73.0	367.3	18.2	65.5	32.0	18.2	55.6	17.5

7.2 Ergebnisse der (EF) unter Anwendung der Beschleunigungstechniken

Tabelle 7.15: Ergebnisse für einen Planungshorizont von 4 Tagen unter Anwendung der Smoothing Methode. Enthalten ist die Anzahl der extra Roll-Ins (ER), die Laufzeit und die Anzahl der benötigten Iterationen. \bar{x} ist das arithmetische Mittel für alle zulässige Instanzen. Ein – bedeutet, dass keine zulässige Lösung gefunden wurde.

	<i>Instanz</i>		<i>Smoothing Methode</i>			<i>EF</i>		
	Züge (#)	Gr. (#)	ER (#)	Iter. (#)	Zeit (s)	ER (#)	Iter. (#)	Zeit (s)
1	105	470	36	220	131.1	36	200	59.9
2	97	500	5	54	21.2	5	54	7.8
3	93	356	0	1	0.4	0	1	0.2
4	21	55	0	1	0.0	0	1	0.0
5	91	330	0	1	0.4	0	1	0.2
6	52	156	0	1	0.1	0	1	0.1
7	86	332	0	1	0.4	0	1	0.1
8	105	457	28	334	184.5	28	267	79.3
9	93	491	11	85	43.6	11	85	25.8
10	129	700	51	371	254.3	51	855	362.1
11	71	308	0	1	0.2	0	1	0.1
12	107	484	0	1	0.3	0	1	0.2
13	75	333	0	1	0.5	0	1	0.1
14	123	602	3	34	21.0	3	34	7.2
15	96	515	12	50	21.5	12	220	66.9
16	85	404	0	1	0.3	0	1	0.1
17	127	666	85	204	150.9	85	560	280.1
18	71	346	0	1	0.2	0	1	0.1
19	145	721	14	366	369.0	14	644	355.0
20	83	381	0	1	0.3	0	1	0.1
21	108	643	81	196	77.2	81	140	35.9
22	103	490	33	67	35.6	3	248	80.6
23	96	507	2	27	10.4	2	27	3.7
24	134	783	73	228	171.1	73	229	90.8
25	92	483	1	95	44.9	1	78	18.7
26	142	834	76	295	261.6	76	326	139.6
27	81	386	0	1	0.3	0	1	0.1
28	109	652	62	240	113.7	62	1305	263.1
29	102	534	5	35	16.2	5	35	5.8
30	104	537	17	110	63.2	17	156	48.2
31	127	710	180	817	635.5	180	360	139.6
32	83	428	0	1	0.3	0	1	0.1
33	125	711	52	559	444.9	52	640	357.7
34	6	10	0	1	0.0	0	1	0.0
35	143	794	22	865	868.0	22	518	335.8
36	112	599	168	201	106.0	168	199	59.1
37	92	496	4	31	11.2	4	31	4.0
\bar{x}	97.7	492.0	27.6	148.6	109.7	27.6	195.3	76.4

Tabelle 7.16: Ergebnisse für einen Planungshorizont von 5 Tagen unter Anwendung der Smoothing Methode. Enthalten ist die Anzahl der extra Roll-Ins (ER), die Laufzeit und die Anzahl der benötigten Iterationen. \bar{x} ist das arithmetische Mittel für alle zulässige Instanzen. Ein – bedeutet, dass keine zulässige Lösung gefunden wurde.

	<i>Instanz</i>		<i>Smoothing Methode</i>			<i>EF</i>		
	Züge (#)	Gr. (#)	ER (#)	Iter. (#)	Zeit (s)	ER (#)	Iter. (#)	Zeit (s)
1	139	646	68	818	806.0	68	279	128.6
2	98	445	0	1	0.5	0	1	0.2
3	72	281	0	1	0.3	0	1	0.1
4	98	339	0	1	0.5	0	1	0.2
5	80	263	0	1	0.3	0	1	0.1
6	86	316	0	1	0.4	0	1	0.2
7	157	816	72	1116	1200.7	68	456	368.9
8	142	741	51	346	316.5	51	243	116.5
9	105	484	0	1	0.5	0	1	0.2
10	101	450	0	1	0.5	0	1	0.2
11	142	636	6	51	46.2	6	51	16.8
12	124	672	12	409	322.6	12	209	95.2
13	96	433	0	1	0.4	0	1	0.1
14	164	874	–	845	1200.8	104	424	293.1
15	131	647	3	409	345.3	3	634	328.7
16	120	564	0	1	0.7	0	1	0.3
17	110	645	81	195	96.8	81	243	60.2
18	163	829	61	496	593.5	61	505	362.6
19	132	702	35	791	860.0	35	244	123.4
20	128	730	35	1068	958.9	35	554	274.9
21	158	932	76	389	468.1	76	289	223.0
22	125	635	27	515	366.1	27	240	93.3
23	116	624	0	1	0.6	0	1	0.2
24	139	752	61	442	435.9	61	784	506.1
25	158	884	–	967	1200.5	204	634	377.0
26	132	699	51	362	267.9	51	752	393.9
27	50	275	0	1	0.1	0	1	0.1
28	144	795	22	803	820.3	22	954	739.8
29	146	808	204	296	267.4	204	947	540.8
30	96	448	0	1	0.4	0	1	0.2
\bar{x}	121.7	612.2	28.8	344.2	352.6	39.0	281.8	168.2

7.3 Ergebnisse der (OF) im Vergleich zur (EF)

Tabelle 7.17: Ergebnisse für einen Planungshorizont von 2 Tagen. Enthalten ist die Anzahl der extra Roll-Ins (ER) und die Laufzeit. \bar{x} ist das arithmetische Mittel für alle zulässige Instanzen. Ein – bedeutet, dass keine zulässige Lösung gefunden wurde.

	<i>Instanz</i>		<i>OF</i>		<i>EF ohne Startlsg.</i>		<i>EF</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
1	32	90	0	0.4	0	0.0	0	0.0
2	73	380	35	9.1	35	35.6	35	19.5
3	67	368	5	7.1	5	8.5	5	7.6
4	30	132	0	1.3	0	0.4	0	0.0
5	35	121	0	1.6	0	0.7	0	0.0
6	58	235	0	19.0	0	6.0	0	0.1
7	14	46	0	0.1	0	0.1	0	0.0
8	7	9	0	0.0	0	0.0	0	0.0
9	48	161	0	2.7	0	2.3	0	0.1
10	43	169	0	2.9	0	1.4	0	0.1
11	1	1	0	0.0	0	0.0	0	0.0
12	51	155	0	7.3	0	3.4	0	0.1
13	61	236	0	4.4	0	6.7	0	0.1
14	25	96	0	0.2	0	0.2	0	0.0
15	28	91	0	0.9	0	0.4	0	0.0
16	77	366	20	134.1	20	36.3	20	3.4
17	59	344	11	27.0	11	5.3	11	3.5
18	34	147	0	1.4	0	0.6	0	0.0
19	51	250	2	3.4	2	3.6	2	0.3
20	78	450	28	111.6	28	29.4	28	10.5
21	42	225	0	2.1	0	1.3	0	0.0
22	29	83	0	1.2	0	0.4	0	0.0
23	56	262	0	4.3	0	4.4	0	0.1
24	51	222	0	4.1	0	3.1	0	0.1
25	28	142	0	1.2	0	0.4	0	0.0
26	47	191	0	2.8	0	2.5	0	0.1
27	75	381	3	8.5	3	21.9	3	1.7
28	48	221	0	3.5	0	2.0	0	0.1
29	26	117	0	0.2	0	0.3	0	0.0
30	70	398	12	8.0	12	16.1	12	1.9
31	59	307	0	4.6	0	5.7	0	0.1

Tabelle 7.17: Ergebnisse für einen Planungshorizont von 2 Tagen (Fortsetzung von der vorhergehenden Seite).

	<i>Instanz</i>		<i>OF</i>		<i>EF ohne Startlsg.</i>		<i>EF</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
32	26	97	0	0.2	0	0.3	0	0.0
33	49	218	0	3.5	0	2.8	0	0.4
34	78	448	28	10.6	28	30.3	28	2.5
35	48	237	0	3.0	0	2.1	0	0.1
36	23	109	0	0.7	0	0.2	0	0.0
37	71	336	0	99.1	0	16.2	0	0.1
38	74	385	11	8.9	11	30.6	11	11.7
39	37	144	0	1.6	0	0.8	0	0.0
40	46	237	0	2.4	0	1.8	0	0.1
41	69	432	77	76.9	77	8.0	77	2.0
42	39	211	0	1.9	0	0.9	0	0.0
43	24	79	0	0.2	0	0.2	0	0.0
44	79	411	29	10.4	29	14.8	29	11.6
45	62	341	2	58.9	2	17.0	2	0.7
46	34	166	0	1.4	0	0.6	0	0.0
47	60	282	4	5.0	4	6.6	4	0.7
48	74	501	24	8.9	24	18.3	24	11.1
49	57	331	1	4.2	1	4.4	1	0.7
50	35	152	0	1.3	0	0.7	0	0.0
51	74	429	24	38.4	24	10.7	24	5.2
52	68	405	11	6.7	11	10.5	11	3.9
53	36	175	0	1.5	0	0.8	0	0.0
54	45	211	0	2.3	0	1.5	0	0.0
55	60	347	27	64.4	27	5.6	27	3.7
56	49	305	0	3.1	0	2.5	0	0.0
57	36	162	0	1.5	0	0.8	0	0.0
58	66	372	5	6.0	5	8.3	5	1.2
59	65	384	17	6.3	17	7.7	17	1.9
60	39	153	0	1.8	0	1.1	0	0.1
61	51	270	134	3.5	134	3.4	134	1.7
62	76	440	26	9.2	26	14.0	26	3.3
63	54	317	0	38.7	0	3.2	0	0.1
64	29	111	0	1.1	0	0.3	0	0.0
65	80	445	40	11.4	40	32.8	40	14.7
66	45	266	0	2.4	0	3.5	0	0.1
67	3	5	0	0.0	0	0.0	0	0.0

7.3 Ergebnisse der (OF) im Vergleich zur (EF)

Tabelle 7.17: Ergebnisse für einen Planungshorizont von 2 Tagen (Fortsetzung von der vorhergehenden Seite).

	<i>Instanz</i>		<i>OF</i>		<i>EF ohne Startlsg.</i>		<i>EF</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
68	3	5	0	0.1	0	0.0	0	0.0
69	90	469	19	15.4	19	32.2	19	4.9
70	53	325	0	3.6	0	3.0	0	0.1
71	32	119	0	1.3	0	0.6	0	0.0
72	80	480	162	109.0	162	34.4	162	20.0
73	61	364	2	5.2	2	6.7	2	0.8
74	31	132	0	0.7	0	0.5	0	0.0
75	38	161	0	1.1	0	0.8	0	0.0
\bar{x}	48.7	244.9	10.1	13.5	10.1	7.2	10.1	2.0

Tabelle 7.18: Ergebnisse für einen Planungshorizont von 3 Tagen. Enthalten ist die Anzahl der extra Roll-Ins (ER) und die Laufzeit. \bar{x} ist das arithmetische Mittel für alle zulässige Instanzen. Ein – bedeutet, dass keine zulässige Lösung gefunden wurde.

	<i>Instanz</i>		<i>OF</i>		<i>EF ohne Startlsg.</i>		<i>EF</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
1	65	257	0	80.1	0	7.8	0	0.1
2	107	581	44	35.5	44	133.5	44	57.6
3	35	148	0	1.5	0	0.8	0	0.0
4	88	340	0	17.2	0	33.9	0	0.2
5	14	46	0	0.1	0	0.1	0	0.0
6	55	170	0	3.7	0	4.0	0	0.1
7	44	170	0	2.1	0	1.5	0	0.1
8	51	155	0	10.7	0	3.1	0	0.1
9	70	268	0	82.8	0	11.8	0	0.1
10	44	155	0	2.2	0	1.7	0	0.1
11	113	575	58	193.9	58	474.5	58	53.1
12	57	282	0	4.7	0	4.4	0	0.1
13	91	485	20	16.5	20	55.4	20	51.1
14	80	440	3	11.1	3	32.5	3	6.6
15	63	259	0	37.2	0	6.4	0	0.1

Tabelle 7.18: Ergebnisse für einen Planungshorizont von 3 Tagen (Fortsetzung von der vorhergehenden Seite).

	<i>Instance</i>		<i>OF</i>		<i>EF ohne Startlsg.</i>		<i>EF</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
16	73	308	0	117.1	0	13.4	0	0.1
17	53	229	0	3.6	0	3.3	0	0.1
18	97	485	5	20.1	5	234.0	5	3.8
19	66	296	0	6.0	0	6.1	0	0.1
20	78	440	12	10.2	12	21.0	12	2.9
21	75	371	0	84.0	0	14.5	0	0.1
22	59	251	0	4.9	0	5.8	0	0.1
23	106	594	47	9.6	47	250.4	47	43.9
24	43	200	0	1.9	0	1.5	0	0.1
25	108	538	3	34.0	3	234.1	3	27.9
26	74	327	0	8.9	0	10.2	0	0.1
27	82	467	2	10.4	2	36.5	2	1.4
28	72	413	75	7.0	75	8.8	75	3.4
29	62	259	0	5.4	0	18.7	0	0.1
30	103	572	27	119.8	27	161.5	27	62.4
31	50	234	0	3.0	0	2.7	0	0.1
32	118	715	68	229.9	68	118.6	68	95.4
33	78	420	0	7.9	0	41.7	0	18.3
34	88	492	38	13.6	38	34.1	38	6.2
35	84	503	11	92.2	11	39.9	11	9.5
36	65	288	0	7.1	0	6.7	0	0.1
37	94	546	61	17.8	61	48.5	61	19.5
38	51	268	0	3.4	0	3.1	0	0.1
39	111	640	57	272.0	57	175.8	57	310.8
40	59	269	0	2.6	0	4.6	0	0.1
41	95	513	154	18.7	154	56.4	154	28.3
42	86	514	18	11.2	18	20.1	18	16.7
43	70	319	6	3.6	6	36.4	6	1.3
44	84	503	12	12.8	12	17.5	12	25.2
45	5	9	0	0.0	0	0.0	0	0.0
46	91	470	19	171.1	19	60.3	19	4.9
47	71	391	0	6.9	0	8.3	0	0.1
48	94	533	168	17.0	168	53.4	168	20.7
49	74	413	4	8.6	4	12.1	4	2.1
50	56	244	0	3.8	0	4.3	0	0.1
\bar{x}	73.0	367.3	18.2	36.9	18.2	50.7	18.2	17.5

7.3 Ergebnisse der (OF) im Vergleich zur (EF)

Tabelle 7.19: Ergebnisse für einen Planungshorizont von 4 Tagen. Enthalten ist die Anzahl der extra Roll-Ins (ER) und die Laufzeit. \bar{x} ist das arithmetische Mittel für alle zulässige Instanzen. Ein – bedeutet, dass keine zulässige Lösung gefunden wurde.

	<i>Instanz</i>		<i>OF</i>		<i>EF ohne Startlsg.</i>		<i>EF</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
1	105	470	36	26.5	36	212.5	36	59.9
2	97	500	5	22.0	5	84.7	5	7.8
3	93	356	0	217.1	0	75.0	0	0.2
4	21	55	0	0.1	0	0.2	0	0.0
5	91	330	0	204.2	0	58.7	0	0.2
6	52	156	0	14.4	0	3.2	0	0.1
7	86	332	0	168.1	0	57.3	0	0.1
8	105	457	28	30.7	28	56.0	28	79.2
9	93	491	11	144.0	11	49.9	11	25.8
10	129	700	51	517.8	51	192.1	51	362.1
11	71	308	0	6.8	0	18.5	0	0.1
12	107	484	0	378.2	0	58.1	0	0.2
13	75	333	0	106.5	0	30.6	0	0.1
14	123	602	3	39.8	3	163.4	3	7.2
15	96	515	12	185.1	12	129.1	12	66.9
16	85	404	0	11.0	0	17.0	0	0.1
17	127	666	85	366.2	85	272.4	85	280.1
18	71	346	0	6.8	0	8.2	0	0.1
19	145	721	14	378.2	302	1200.2	14	355.0
20	83	381	0	162.6	0	15.2	0	0.1
21	108	643	81	151.7	81	147.3	81	35.9
22	103	490	33	25.6	33	278.8	33	80.6
23	96	507	2	172.8	2	94.1	2	3.7
24	134	783	73	257.3	73	195.3	73	90.8
25	92	483	1	13.6	1	55.7	1	18.7
26	142	834	75	85.4	75	346.1	75	138.9
27	81	386	0	10.8	0	30.6	0	0.1
28	109	652	62	106.4	62	66.4	62	263.1
29	102	534	5	24.4	5	79.9	5	5.8
30	104	537	17	153.6	17	119.5	17	48.2
31	127	710	180	426.9	180	167.4	180	139.6
32	83	428	0	10.9	0	15.1	0	0.1
33	125	711	52	182.9	52	178.6	52	357.7
34	6	10	0	0.0	0	0.0	0	0.0
35	143	794	22	103.6	22	299.2	22	335.8
36	112	599	168	47.6	168	126.7	168	59.1
37	92	496	4	211.9	4	75.2	4	4.0
\bar{x}	97.7	492.0	27.6	134.4	35.4	134.5	27.6	76.4

Tabelle 7.20: Ergebnisse für einen Planungshorizont von 5 Tagen. Enthalten ist die Anzahl der extra Roll-Ins (ER) und die Laufzeit. \bar{x} ist das arithmetische Mittel für alle zulässige Instanzen. Ein – bedeutet, dass keine zulässige Lösung gefunden wurde.

	<i>Instanz</i>		<i>OF</i>		<i>EF ohne Startlsg.</i>		<i>EF</i>	
	Züge (#)	Gruppen (#)	ER (#)	Zeit (s)	ER (#)	Zeit (s)	ER (#)	Zeit (s)
1	139	646	68	95.4	68	218.7	68	128.6
2	98	445	0	16.7	0	44.2	0	0.2
3	72	281	0	7.5	0	14.0	0	0.1
4	98	339	0	21.4	0	60.1	0	0.2
5	80	263	0	9.9	0	19.5	0	0.1
6	86	316	0	11.0	0	17.0	0	0.2
7	157	816	68	564.8	653	1200.2	68	368.9
8	142	741	51	142.6	51	236.6	51	116.5
9	105	484	0	286.0	0	82.9	0	0.2
10	101	450	0	18.8	0	27.3	0	0.2
11	142	636	6	788.6	6	1143.5	6	16.8
12	124	672	12	65.8	12	102.7	12	95.2
13	96	433	0	15.8	0	35.0	0	0.1
14	164	874	104	244.6	164	1200.2	104	293.1
15	131	647	3	724.2	3	151.0	3	328.7
16	120	564	0	43.9	0	113.7	0	0.3
17	110	645	81	159.2	81	311.7	81	60.2
18	163	829	61	195.6	61	898.9	61	362.6
19	132	702	35	70.9	35	154.5	35	123.4
20	128	730	35	43.9	35	926.8	35	274.9
21	158	932	76	487.8	76	349.3	76	223.0
22	125	635	27	45.1	78	1200.1	27	93.3
23	116	624	0	30.6	0	100.9	0	0.2
24	139	752	61	482.6	143	1200.0	61	506.1
25	158	884	204	137.9	–	1200.0	204	377.0
26	132	699	51	379.7	51	645.3	51	393.9
27	50	275	0	2.9	0	3.8	0	0.1
28	144	795	22	645.2	22	745.6	22	739.8
29	146	808	204	473.1	644	1200.0	204	540.8
30	96	448	0	18.7	0	25.4	0	0.2
\bar{x}	121.7	612.2	39.0	207.7	72.8	454.3	39.0	168.2

Literaturverzeichnis

- [1] M.Bohlin, H.Flier, J.Maue and M.Mihalák: *Hump Yard Track Allocation with Temporary Car Storage*, In: The 4th International Seminar on Railway Operations Modelling and Analysis (RailRome), 2011.
- [2] M.Bohlin, H.Flier, J.Maue and M.Mihalák: *Track Allocation in Freight-Train Classification with Mixed Tracks*, In: 11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems, volume 20 OpenAccess Series in Informatics (OASICs), pages 38–51, Dagstuhl, Germany, 2011. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
- [3] M.Bohlin, F.Dahms, H. Flier and S.Gestrelus: *Optimal Freight Train Classification using Column Generation*, In: 12th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems, volume 25 OpenAccess Series in Informatics (OASICs), pages 10–22, Dagstuhl, Germany, 2012. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
- [4] V. Chvátal: *Linear Programming*, Freeman, New York, 1983.
- [5] G.L. Nemhauser, L.A. Wolsey: *Integer and Combinatorial Optimization*, Wiley, New York, 1988.
- [6] H.J. Zimmermann: *Operations Research Methoden und Modelle*, Vieweg, Braunschweig, 2. Auflage, 1992.
- [7] W. Domschke, A. Drexl: *Einführung in Operations Research*, Springer, Berlin, 1998.
- [8] M.R. Garey, D.S. Johnson: *Computers and Intractability*, Freeman, New York, 1979.
- [9] F. Bonomo, G. Durán, J. Marengo: *Exploring the complexity boundary between coloring and list-coloring*, Annals OR, 169(1):3-16, 2009.
- [10] T. Cormen, C. Leiserson, R. Rivest and C. Stein: *Introduction To Algorithms*, MIT Press, 3rd edition, 2009.
- [11] T. Grünert, St. Irnich: *Optimierung im Transport*, Shaker, Aachen, Band 1, 2005.
- [12] M.E.Lübbecke: *Column Generation*, In J.J.Cochran (Ed.), Encyclopedia of Operations Research and Management Science, John Wiley & Sons, Chichester, 2011.

- [13] A. Pessoa, E. Uchoa, M. Poggi de Arag ao, R. Rodrigues: *Algorithms over arc-time indexed formulations for single and parallel machine scheduling problems*, Report RPEP Vol. 8 no. 8, Universidade Federal Fluminense, 2008.
- [14] F. Margot: *Pruning by isomorphism in branch-and-cut*, *Math. Programming*, 94(1):71-90, 2002.
- [15] F. Margot: *Small covering designs by branch-and-cut*, *Mathematical Programming, Series B*, 94(2-3):207-220, 2003.
- [16] F. Margot: *Exploiting orbits in symmetric ILP*, *Mathematical Programming, Series B*, 98:21-23, 2003.
- [17] F. Margot: *Symmetric ILP: Coloring and small integers*, *Discrete Optimization*, 4:40-62, 2007.
- [18] J. Ostrowski, J. Linderoth, F. Rossi, St. Smiriglio: *Orbital Branching*, In: Proc. of the 12th Integer Programming and Combinatorial Optimization conference (IPCO), M. Fischetti and D. Williamson (Eds.), volume 4513 of LNCS, Springer-Verlag, 2007.
- [19] J. Ostrowski, J. Linderoth, F. Rossi, St. Smiriglio: *Constraint Orbital Branching*, In: Proc. of the 13th Integer Programming and Combinatorial Optimization conference (IPCO), 2008. to appear.
- [20] V. Kaibel, M. Peinhardt, M. Pfetsch: *Orbitopal Fixing*, In: Proc. of the 12th Integer Programming and Combinatorial Optimization conference (IPCO), Matteo Fischetti and David Williamson (Eds.), volume 4513 of LNCS, pp. 74-88, Springer-Verlag, 2007.
- [21] V. Kaibel, M. Peinhardt, M. Pfetsch: *Packing and Partitioning Orbitopes*, *Mathematical Programming*, 114(1):1-36, 2008.
- [22] T. Berthold, M. Pfetsch: *Detecting Orbitopal Symmetries*, In: Operations Research Proceedings 2008, pp. 433-438, 2009.
- [23] T. Achterberg, T. Berthold, S. Heinz, T. Koch, K. Wolter: *Constraint Integer Programming: Techniques and Applications*, Forschungsbericht 08-43, ZIB, 2008.
- [24] J.Desrosiers, M.E.Lübbecke: *Selected topics in column generation*, *Oper. Res.* 53(6):1007-1023, 2005.
- [25] D. Villeneuve: *Logiciel de Génération de Colonnes* PhD thesis, École Polytechnique de Montréal, 1999.
- [26] F. Vanderbeck, L.A. Wolsey: *An exact algorithm for IP column generation*, *Operations Research Letters* 19 151-159, 1996.

Versicherung an Eides statt

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmitteln benutzt habe.

Aachen, den 28.09.2012