#### Technische Universität Berlin

Fakultät IV: Elektrotechnik und Informatik Institut für Softwaretechnik und Theoretische Informatik Algorithmik und Komplexitätstheorie (AKT)



# Algorithmen und Experimente für das Electric Vehicle Scheduling-Problem

### Bachelorarbeit

von Jonas Schulte-Mattler

zur Erlangung des Grades "Bachelor of Science" (B. Sc.) im Studiengang Computer Science (Informatik)

Erstgutachter: Prof. Dr. Rolf Niedermeier Zweitgutachter: Prof. Dr.-Ing. Dietmar Göhlich Betreuer: Leon Kellerhals, Dominic Jefferies,

Prof. Dr. Rolf Niedermeier

Quellen und H	Hilfsmittel angefertigt ha	be.
Die selbststän	dige und eigenhändige A	ausfertigung versichert an Eides statt
Berlin, den	——————————————————————————————————————	Unterschrift

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten

### Zusammenfassung

Die Luftverschmutzung in Städten und der Klimawandel erzeugen großes Interesse an elektrischen Bussen, die lokal emissionsfrei sind und eine Reduzierung von Treibhausgasen versprechen. Da elektrische Busse eine beschränkte Reichweite haben, können sie nicht dieselben Fahrten ausführen wie Dieselbusse. Wollen wir so wenige elektrische Fahrzeuge wie möglich für bestehende Fahrpläne einsetzen, stehen wir vor einem  $\mathcal{NP}$ -schweren Problem. Wir geben zwei neue Lösungsansätze an, die genau die Beschaffenheiten von batteriebetriebenen Bussen berücksichtigen. Wir formulieren ein ILP, mit welchem nach Datenvorbehandlung Instanzen mit knapp 4000 Fahrgastfahrten in Berlin optimal gelöst werden konnten. Weiter geben wir eine Heuristik an, die auf den getesteten Instanzen in Sekunden teils optimale Lösungen und höchstens 6% schlechtere Lösungen als das Optimum liefert.

#### **Abstract**

Air pollution in cities and climate change arise the interest in electric buses that operate emission free and promise a reduction of green house gas emissions. Since electric vehicles have limited driving range and recharging is time consuming, battery electric buses cannot execute the same schedules as traditional buses. Finding the minimum number of electric buses to operate given timetables results in an  $\mathcal{NP}$ -hard problem. To solve this, we propose two novel approaches that consider the characteristics of battery electric vehicles. We give an Integer Linear Program that we could solve using data reduction for an instance with almost 4000 trips in Berlin optimally. Thus we developed a greedy-algorithm that finds within seconds either optimal solutions or solutions that are at most 6% worse than the optimum.

# Inhaltsverzeichnis

1	Einführung	5
2	Notation und Grundlagen	7
3	Problemformulierung und Modell3.1Vehicle Scheduling Problem3.2Electric Vehicle Scheduling Problem	<b>9</b> 10
4	Komplexität	14
5	Integer Linear Program 5.1 ILP für das VSP	16 16 18
6	Greedy-Algorithmus	23
7	Experimente	27
8	Fazit	32
GI	ossar	32
Lit	teratur	34

# 1 Einführung

Großstädte weltweit elektrifizieren ihre Busflotten [8]. In zehn Jahren sollen die Hälfte aller Busse in Europa mit Batterie betriebene Elektrobusse sein [14]. Wegen der beschränkten Reichweite der Fahrzeuge und Ladezeiten der Batterie können elektrische Busse nicht dieselben Fahrten ausführen wie bislang Dieselbusse [6]. Dadurch stehen wir vor einem neuen Problem der Umlaufplanung von elektrischen Fahrzeugen, dem ELECTRIC VEHICLE SCHEDULING PROBLEM (EVSP). Gegeben sind eine Reihe von Fahrgastfahrten. Die Aufgabe ist einen Umlaufplan zu erstellen, sodass jeder Fahrgastfahrt ein elektrisches Fahrzeug zugewiesen wird. Es gibt verschiedene Optimierungskriterien.

In dieser Arbeit suchen wir Algorithmen für das EVSP mit dem Ziel, die minimale Anzahl benötigter batteriebetriebener Busse zu finden.

Verwandte Arbeiten. Einen Überblick über Lösungsansätze zur Umlaufplanung von Fahrzeugen, dem Vehicle Scheduling, geben Bunte und Kliewer [3]. Auch das EVSP wurde bereits in einigen Varianten studiert.

Paul und Yamada [9] behandeln das Problem der Umlaufplanung von elektrischen Bussen in Kombination mit Dieselbussen. Ihr Ziel ist die elektrisch gefahrene Strecke zu maximieren. Sassi und Oulamara [11] haben das gleichen Ziel und minimieren noch dazu die Ladekosten. Adler [1] behandelt das Problem der Umlaufplanung von Fahrzeugen mit alternativem Treibstoff. Diese Fahrzeuge haben eine begrenzte Reichweite und können an bestimmten Ladestationen in einem festen Zeitraum volltanken. Reuer, Kliewer und Wolbeck [10] nehmen an, dass die Batterie eines elektrischen Fahrzeugs innerhalb einer konstanten Zeit unabhängig vom Ladezustand vollständig geladen werden kann. Van Kooten Niekerk, van den Akker und Hoogeveen [12] berücksichtigen, dass die Batterie zum Teil geladen werden kann. Dabei nehmen sie an: Befindet sich an einer Endhaltestelle eine Ladestation, kann vor oder nach einer Fahrgastfahrt, die an dieser Haltestelle beginnt oder endet, geladen werden. Wen u. a. [13] nehmen an, dass vor und nach jeder Fahrgastfahrt an allen Ladestationen, auch an mehreren hintereinander, beliebig lange geladen werden kann.

**Eigener Beitrag.** In dieser Arbeit präsentieren wir ein Modell für das Problem der Umlaufplanung von elektrischen Bussen. Es berücksichtigt genau die Beschaffenheiten von batteriebetriebenen Bussen:

- Die Ladedauer hängt vom Ladezustand zu Beginn eines Ladevorgangs ab,
- die Batterie muss nicht voll geladen werden,
- es kann an bestimmten Ladestationen geladen werden.

#### 1 Einführung

Unser Modell verallgemeinert die Modelle der Arbeiten [1, 10, 12] und bildet damit die relevantesten praktischen Anforderungen ab. Wir zeigen, dass das EVSP  $\mathcal{NP}$ -schwer ist und wahrscheinlich nicht mit einem Faktor kleiner als  $^{3}/_{2}$  in polynomieller Zeit approximiert werden kann. Wir präsentieren zwei Lösungsansätze, um die minimale Anzahl benötigter Busse zu finden. Wir formulieren ein Integer Linear Program, mit welchem nach Datenvorbehandlung eine Instanz mit knapp 4000 Fahrgastfahrten des öffentlichen Nahverkehrs in Berlin optimal gelöst werden konnte. Außerdem geben wir einen Greedy-Algorithmus an, der in  $\mathcal{O}(n^{2}m)$  Zeit für n Fahrgastfahrten und m Ladestationen eine gültige Lösung liefert. Zwar garantiert der Algorithmus keine Güte der Lösung, liefert aber in unseren Experimenten Lösungen, die höchstens 6% schlechter sind als das Optimum, und für manche Instanzen sogar eine optimale Lösung. Der Algorithmus löst in 16s die Instanz mit knapp 4000 Fahrgastfahrten.

Aufbau der Arbeit. In Kapitel 2 führen wir unsere Notation für Mengen und Graphen ein und geben Grundlagen zur Komplexität und Approximationsalgorithmen, die wir in dieser Arbeit verwenden. In Kapitel 3 definieren wir unser Modell für die Umlaufplanung von batteriebetriebenen Bussen und formulieren das Problem. Die Härte des Problems untersuchen wir in Kapitel 4. Dann geben wir in Kapitel 5 ein Integer Linear Program und in Kapitel 6 einen Greedy-Algorithmus zur Lösung an. Wir bewerten unsere Lösungsansätze experimentell mit Daten des öffentlichen Personennahverkehrs aus Berlin in Kapitel 7. Abschließend fassen wir in Kapitel 8 die Inhalte und Ergebnisse der Arbeit zusammen und diskutieren weitere Forschung.

# 2 Notation und Grundlagen

In diesem Kapitel geben wir Grundlagen der Komplexität und Approximationsalgorithmen und führen Standardnotation für Mengen und Graphen ein, die wir in dieser Arbeit verwenden. Mit N bezeichnen wir die Menge der natürlichen Zahlen ohne Null.

**Mengen.** Für die Menge der natürlichen Zahlen bis n schreiben wir [n], sprich  $[n] := \{i \in \mathbb{N} \mid 1 \leq i \leq n\}$ ; für die Menge der natürlichen Zahlen von n bis m schreiben wir [n, m], sprich  $[n, m] := \{i \in \mathbb{N} \mid n \leq i \leq m\}$ .

**Graphentheorie.** Sei G = (V, E) ein gerichteter Graph, wobei V die Knotenmenge und  $E \subseteq \{(v, w) \mid v, w \in V\}$  die Kantenmenge bezeichnet. Wir nennen  $N^-(v)$  die eingehenden Nachbarn von  $v \in V$ , das heißt  $N^-(v) := \{w \in V \mid (w, v) \in E\}$  und  $N^+(v)$  die ausgehenden Nachbarn von  $v \in V$ , das heißt  $N^+(v) := \{w \in V \mid (v, w) \in E\}$ . Ein Weg ist eine Folge von Knoten  $(v_1, ..., v_n)$  aus V, wobei alle aufeinander folgenden Knotenpaare durch eine Kante verbunden sind, sprich  $\forall i \in [n-1]$  gilt  $(v_i, v_{i+1}) \in E$ . Einen  $Weg (v_1, ..., v_n)$  bezeichnen wir als Pfad, wenn alle Knoten des Wegs unterschiedlich sind, sprich  $\forall i \in [n]$  gilt,  $i \neq j \rightarrow v_i \neq v_j$ . Wir nennen einen  $Pfad (s, v_1, ..., v_n, t)$ , der an Knoten s beginnt und an Knoten t endet, einen s-t-Pfad.

Parametrisierte Komplexität. Ein Problem ist eine formale Sprache  $L\subseteq \Sigma^*$  über einem endlichen Alphabet  $\Sigma$ . Eine Instanz  $I\in \Sigma^*$  ist eine Ja-Instanz von Problem L genau dann, wenn  $I\in L$  ist. Ein parametrisiertes Problem P ist eine Teilmenge  $P\subseteq \{(x,k)\in \Sigma^*\times \mathbb{N}\}$  mit dem Parameter k. Ein parametrisiertes Problem ist in FPT, wenn man für jede Eingabe (x,k) in  $f(k)\cdot |x|^{\mathcal{O}(1)}$  Zeit entscheiden kann, ob (x,k) in P liegt. Dabei ist die Funktion f berechenbar und hängt nur von k ab.

Ein parametrisiertes Problem P ist W[1]-schwer, wenn es eine parametrisierte Reduktion von einem W[1]-schweren Problem Q auf P gibt. Eine parametrisierte Reduktion von einem parametrisierten Problem Q auf ein anderes parametrisiertes Problem P ist eine Funktion, die für zwei berechenbare Funktionen f und g sowie einer gegebenen Instanz (x,k) von Q in  $f(k) \cdot |x|^{\mathcal{O}(1)}$  Zeit eine Instanz (x',k') von P berechnet, sodass  $k' \leq g(k)$  und  $(x,k) \in Q$  genau dann, wenn  $(x',k') \in P$ .

**Approximationsalgorithmen.** Bei einem Optimierungsproblem suchen wir für ein Problem eine optimale Lösung. Die Güte einer Lösung  $\omega$  wird durch eine Zielfunktion  $f(\omega)$  bestimmt. Ein Minimierungsproblem ist ein Optimierungsproblem, bei dem wir nach der Lösung  $\omega^*$  mit dem kleinsten  $f(\omega^*)$  suchen.

### $2\ Notation\ und\ Grundlagen$

Eine  $\alpha$ -Approximation für ein Minimierungsproblem ist ein Algorithmus, der für alle Instanzen des Problems eine Lösung  $\omega$  findet, sodass gilt  $f(\omega) \leq \alpha f(\omega^*)$ .

# 3 Problemformulierung und Modell

Wir führen in diesem Kapitel ein Modell für das Problem der Umlaufplanung von elektrischen Fahrzeugen mit Batterie ein. Das Problem ist eine Erweiterung des klassischen Problems der Umlaufplanung von Fahrzeugen, dem VEHICLE SCHEDULING PROBLEM (VSP). Daher führen wir zuerst unser Modell und unsere Notation für das VSP ein, um im nächsten Abschnitt ein Modell und die Formulierung des Electric VSP anzugeben.

### 3.1 Vehicle Scheduling Problem

Zuerst definieren wir ein Modell für das Problem, danach geben wir Definitionen, die schließlich eine exakte Problemstellung ermöglichen und betrachten ein Beispiel.

Gegeben sind Fahrgastfahrten  $\mathcal{T}$ . Eine Fahrgastfahrt  $\tau_i = (d_i, a_i, s_i, e_i) \in \mathcal{X}^2 \times \mathbb{N}^2$  startet am Abfahrtsort  $d_i$ , endet am Ankunftsort  $a_i$  und findet im Zeitraum  $[s_i, e_i]$  statt. Orte nennen wir  $\mathcal{X}$ . Um von Ort x zu y zu fahren, benötigt ein Fahrzeug Fahrzeit T(x, y). Der Umlauf  $\psi \subseteq \mathcal{T}$  eines Fahrzeugs besteht aus einer Menge von Fahrgastfahrten. Ein Umlaufplan  $\Psi$  ist eine Menge von Umläufen.

Ein Fahrzeug kann nach einer Fahrgastfahrt  $\tau_i$  eine spätere Fahrt  $\tau_j$  ausführen, wenn das Fahrzeug nach der Ankunftszeit  $e_i$  von  $\tau_i$  noch vor der Abfahrtszeit  $s_j$  von  $\tau_j$  am Abfahrtsort  $d_j$  sein kann, also wenn sich der Zeitraum  $[s_i, e_i + T(a_i, d_j)]$  nicht mit dem Zeitraum  $[s_j, e_j]$ , in dem Fahrt  $\tau_j$  stattfindet, überlappt. Findet  $\tau_j$  vor  $\tau_i$  statt, so überlappen sich  $\tau_j$  und  $\tau_i$  zeitlich, wenn sich  $[s_j, e_j + T(a_j, d_i)]$  mit  $[s_i, e_i]$  überlappt.

**Definition 3.1.** Zwei Fahrgastfahrten  $\tau_i, \tau_j \in \mathcal{T}$  überlappen sich zeitlich, wenn  $[s_i, e_i + T(a_i, d_i)] \cap [s_i, e_i + T(a_i, d_i)] \neq \emptyset$  ist. Wir schreiben  $\tau_i \cap \tau_j \neq \emptyset$ .

**Definition 3.2.** Ein Umlauf  $\psi$  ist *möglich*, wenn sich keine zwei Fahrgastfahrten des Umlaufs zeitlich überlappen, also für alle  $\tau_i, \tau_j \in \psi$  gilt, dass  $\tau_i \cap \tau_j = \emptyset$  ist.

**Definition 3.3.** Ein gültiger Umlaufplan  $\Psi$  ist eine Menge möglicher Umläufe, sodass jede Fahrgastfahrt in genau einem Umlauf enthalten ist.

Unser Ziel ist es, die Fahrgastfahrten mit so wenigen Fahrzeugen wie möglich auszuführen. Damit ergibt sich folgendes Problem:

Vehicle Scheduling Problem

**Eingabe:** Fahrgastfahrten  $\mathcal{T}$  und die für die Fahrt zwischen zwei Orten benötigte Zeit  $T: \mathcal{X}^2 \to \mathbb{N}$ .

**Aufgabe:** Finde einen gültigen Umlaufplan  $\Psi$  mit der kleinsten Anzahl an Umläufen.

Betrachten wir ein

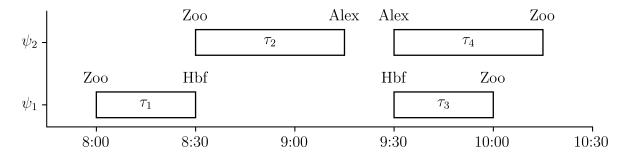


Abbildung 3.1: Ein optimaler Umlaufplan  $\Psi$  für unsere Beispielinstanz, der aus zwei Umläufen  $\psi_1$  und  $\psi_2$  besteht. Jedes umrandete Intervall ist eine Fahrgastfahrt. An den Enden des Intervalls steht der Abfahrts-und Ankunftsort.

**Beispiel.** Gegeben sind vier Fahrgastfahrten  $\mathcal{T}$ , die am Vormittag zwischen dem Zoologischen Garten (Zoo), dem Haupfbahnhof (Hbf) und dem Alexanderplatz (Alex) in Berlin stattfinden, siehe Tabelle 3.1. Die Fahrzeit T zwischen zwei Endhaltestellen x und y berechnen wir aus der Distanz D(x,y) und einer Durchschnittsgeschwindigkeit  $\overline{v}$ , sprich  $T(x,y) = D(x,y)/\overline{v}$ . Die Distanzen finden sich in Tabelle 3.2,  $\overline{v}$  sei gleich 10 km/h.

Wir bemerken, dass sich Fahrgastfahrten  $\tau_1$  und  $\tau_2$  zeitlich überlappen. Zwar beginnt  $\tau_2$ , wenn  $\tau_1$  endet, jedoch startet  $\tau_2$  nicht am Ankunftsort von  $\tau_1$ . Genauso überlappt sich  $\tau_2$  zeitlich mit  $\tau_3$ . Denn die Fahrt vom Ankunftsort von  $\tau_2$  zum Abfahrtsort von  $\tau_3$  dauert länger als 15 Minuten. Da sich aber weder  $\tau_2$  und  $\tau_4$  noch  $\tau_1$  und  $\tau_3$  überlappen, gibt es eine optimale Lösung  $\Psi = \{\psi_1, \psi_2\}$ , die aus zwei Umläufen  $\psi_1 = \{\tau_1, \tau_3\}$  und  $\psi_2 = \{\tau_2, \tau_4\}$  besteht, dargestellt in Abbildung 3.1.

Nachdem wir das Problem der Umlaufplanung herkömmlicher Busse kennen, definieren wir nun ein Modell für die Umlaufplanung von elektrischen Bussen mit Batterie.

### 3.2 Electric Vehicle Scheduling Problem

Wir nutzen die eingeführte Notation und führen weitere Parameter für ein Modell für das EVSP ein. Dann geben wir Definitionen, um das Problem formulieren zu können. Wir betrachten wieder ein Beispiel und schließen mit Beobachtungen von Ladevorgängen.

Ein elektrisches Fahrzeug verfügt über eine Batteriekapazität  $\beta$ , verbraucht für eine

Fahrgastfahrt	Abfahrtsort	Abfahrtszeit	Ankunftsort	Ankunftszeit
$ au_1$	Zoo	8:00	Hbf	8:30
$ au_2$	Zoo	8:30	Alex	9:15
$ au_3$	Hbf	9:30	Zoo	10:00
$ au_4$	Alex	9:30	Zoo	10:15

Tabelle 3.1: Ein Beispiel für vier Fahrgastfahrten in Berlin.

Fahrgastfahrt  $\tau$  Energie  $Q^P(\tau)$ , für eine Fahrt zwischen Ort x und y Energie Q(x,y) und kann an einer Ladestation  $c \in \mathcal{C} \subseteq \mathcal{X}$  geladen werden. Eine Ladefunktion  $Q^C(q, \Delta t)$  gibt den Ladezustand zwischen 0 und  $\beta$  an, nachdem im Zeitraum  $\Delta t$  ab Ladezustand  $q \in [0, \beta]$  geladen wurde. Jedes Fahrzeug startet und endet an einem Depot  $\delta$ .

Wir nehmen an, dass ein Fahrzeug vor der ersten Fahrgastfahrt, zwischen zwei Fahrgastfahrten und nach der letzten Fahrgastfahrt an einer Ladestation geladen werden kann. Ladevorgänge an mehreren Ladestationen hintereinander sind dann interessant, wenn für die Fahrt zwischen zwei Orten so viel Energie benötigt wird, dass mehrere Ladevorgänge nötig sind und mehrere Ladestationen auf dieser Strecke liegen. Das ist im öffentlichen Nahverkehr einer Stadt nicht der Fall und wir können diese Annahme treffen.

Einen Ladevorgang, der zwischen zwei Fahrgastfahrten  $\tau_i, \tau_j \in \mathcal{T}$  an Ladestation c stattfindet, nennen wir  $\zeta_{ij}^c \in \mathcal{T}^2 \times \mathcal{C}$ . Einen Ladevorgang, der vor der ersten Fahrgastfahrt  $\tau_i$  an Ladestation c stattfindet, nennen wir  $\zeta_{\delta i}^c \in \delta \times \mathcal{T} \times \mathcal{C}$ ; einen Ladevorgang, der nach einer letzten Fahrgastfahrt  $\tau_j$  an Ladestation c stattfindet, nennen wir  $\zeta_{j\delta}^c \in \mathcal{T} \times \delta \times \mathcal{C}$ . Jeder Ladevorgang findet so lange wie möglich statt oder so lange bis die Batterie vollständig geladen ist. Die Menge aller Ladevorgänge sei gleich Z.

Der Umlauf eines elektrischen Fahrzeugs  $\omega$  besteht aus einer Menge von Fahrgastfahrten und Ladevorgängen,  $\omega \subseteq \mathcal{T} \cup Z$ . Das Glossar gibt eine Übersicht über alle von uns eingeführten Größen.

Ein elektrisches Fahrzeug kann einen Umlauf nur dann absolvieren, wenn der Ladezustand des Fahrzeugs während des Umlaufs immer mindestens null ist. Außerdem kann der Ladezustand nie größer sein als die Batteriekapazität.

**Definition 3.4.** Der Umlauf  $\omega$  eines elektrischen Fahrzeugs ist *möglich*, wenn sich keine zwei Fahrgastfahrten zeitlich überlappen und der Ladezustand des Fahrzeugs immer mindestens null und höchstens gleich der Batteriekapazität ist.

**Definition 3.5.** Ein gültiger Umlaufplan  $\Omega$  ist eine Menge möglicher Umläufe von elektrischen Fahrzeugen, sodass jede Fahrgastfahrt in genau einem Umlauf enthalten ist.

Damit ergibt sich folgendes Problem:

Tabelle 3.2: Die Distanzen D zwischen den Orten unserer Beispielinstanz in Kilometern.

	Depot $\delta$	Zoo	Hbf	Alex
Depot $\delta$ Zoo	0	3	7	10
Zoo	3	0	5	7
Hbf	7	5	0	4
Alex	10	7	4	0

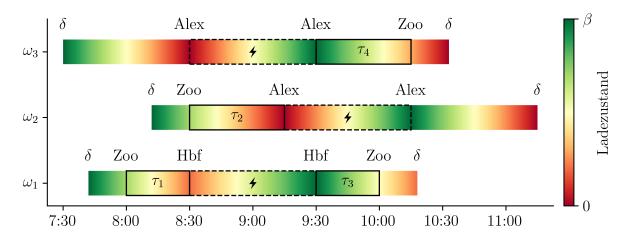


Abbildung 3.2: Ein optimaler Umlaufplan  $\Omega$  für unsere Beispielinstanz, der drei Umläufe  $\omega_1, \omega_2$  und  $\omega_3$  enthält. Jedes umrandete Intervall ist eine Fahrgastfahrt. An den Enden des Intervalls steht der Abfahrts-und Ankunftsort. Jedes gestrichelt umrandete Intervall ist ein Ladevorgang. Am Depot  $\delta$  startet und endet jeder Umlauf. Der Ladezustand der Fahrzeuge ist farblich kodiert.

ELECTRIC VEHICLE SCHEDULING PROBLEM

**Eingabe:** Depot  $\delta$ , Ladestationen  $\mathcal{C}$ , Fahrgastfahrten  $\mathcal{T}$ , die für eine Fahrgastfahrt benötigte Energie  $Q^P$ , die für eine Fahrt zwischen zwei Orten benötigte

Zeit T und Energie Q, eine Ladefunktion  $Q^C$  und Batteriekapazität  $\beta$ .

**Aufgabe:** Finde einen gültigen Umlaufplan  $\Omega$  mit der kleinsten Anzahl an Umläufen.

Die Entscheidungsvariante des EVSP hat die Eingabe des EVSP und eine natürliche Zahl k und fragt nach einem Umlaufplan  $\Omega$  mit höchstens k Umläufen.

Betrachten wir wieder ein

Beispiel. Wir erweitern die Beispielinstanz  $(\mathcal{T}, T)$  des VSP aus dem letzten Kapitel um die weiteren Größen  $(\delta, \beta, Q^P, Q, \mathcal{C}, Q^C)$  der Eingabe des EVSP. Die Energie Q, die ein Fahrzeug für eine Fahrt zwischen zwei Endhaltestellen x und y verbraucht, berechnen wir aus der Distanz D zwischen den Orten und einem durchschnittlichem Verbrauch  $\overline{q}$  von 1kWh/km, sprich  $Q(x,y) = D(x,y)\overline{q}$ . Die Distanzen sind in Tabelle 3.2 definiert. Wir nehmen an, dass für eine Fahrgastfahrt  $\tau_i$  genauso viel Energie benötigt wird wie für eine Leerfahrt zwischen dem Abfahrtsort  $d_i$  und Ankunftsort  $a_i$  von  $\tau_i$ , sprich  $Q^P(\tau_i) = Q(d_i, a_i)$ . Die Batteriekapazität  $\beta$  eines Fahrzeugs sei 10kWh. Am Hauptbahnhof (Hbf), am Alexanderplatz (Alex) und am Depot  $\delta$  befinde sich eine Ladestation,  $\mathcal{C} = \{\text{Hbf}, \text{Alex}, \delta\}$ . Wir nehmen an, die Batterie wird mit einer konstanten Leistung  $p_C$  von 10kW geladen. Also sei Ladefunktion  $Q^C(q, \Delta t) = \min(q + p_C \Delta t, \beta)$ .

Wir stellen fest, dass der Umlauf, der  $\tau_1$  und  $\tau_3$  enthält, auch von einem elektrischen Bus ausgeführt werden kann: Der Ladezustand nach  $\tau_1$  ist gleich 2kWh, da für die Fahrt vom Depot zum Abfahrtsort von  $\tau_1$  3kWh und für  $\tau_1$  5kWh benötigt werden. Zwischen  $\tau_1$ 

und  $\tau_3$  kann am Hbf 1h lang geladen werden. Danach ist die Batterie des Fahrzeugs wieder vollständig geladen und das Fahrzeug besitzt genug Energie, um  $\tau_3$  auszuführen und darauf ins Depot zu fahren. Also ist ein möglicher Umlauf eines elektrischen Fahrzeugs  $\omega_1 = \{\tau_1, \zeta_{1,3}^{\text{Hbf}}, \tau_3\}.$ 

Jedoch kann der Umlauf  $\psi_2 = \{\tau_2, \tau_4\}$ , der Teil der optimalen Lösung für das VSP ist, nicht von einem elektrischen Fahrzeug ausgeführt werden. Denn ein Fahrzeug kann zwischen  $\tau_2$  und  $\tau_4$  in 15 Minuten am Alexanderplatz 2.5kWh laden. Nach  $\tau_2$  ist der Ladezustand eines Fahrzeugs höchstens gleich 3kWh, da die Batteriekapazität 10kWh ist und für  $\tau_2$  7kWh verbraucht werden. Also ist der größtmögliche Ladezustand eines Fahrzeugs, nach  $\tau_2$  zu Beginn von  $\tau_4$  gleich 4.5kWh, für  $\tau_4$  werden allerdings 7kWh benötigt. Lädt ein Fahrzeug nach  $\tau_2$  am Alexanderplatz bis die Batterie voll geladen ist, kann das Fahrzeug zum Depot zurückkehren. Fährt ein anderes Fahrzeug so früh vom Depot zum Alexanderplatz, dem Abfahrtsort von  $\tau_4$ , dass die Batterie vor  $\tau_4$  voll geladen werden kann, so kann  $\tau_4$  ausgeführt und anschließend wieder ins Depot gefahren werden.

Also ist eine optimale Lösung der Umlaufplan  $\Omega = \{\omega_1, \omega_2, \omega_3\}$ , der aus den Umläufen  $\omega_1 = \{\tau_1, \zeta_{1,3}^{\text{Hbf}}, \tau_3\}$ ,  $\omega_2 = \{\tau_2, \zeta_{2,\delta}^{\text{Alex}}\}$  und  $\omega_3 = \{\zeta_{\delta,4}^{\text{Alex}}, \tau_4\}$  besteht, dargestellt in Abbildung 3.2.

**Beobachtungen.** Zwischen zwei Fahrten  $\tau_i$  und  $\tau_j$  ist an Ladestation c ein Ladevorgang  $m\ddot{o}glich$ , wenn die Ladedauer  $t^c_{ij}$  größer als null ist. Diese ist gleich der Abfahrtszeit  $s_j$  minus der Fahrtdauer  $T(c,d_j)$  von c zu Abfahrtsort  $d_j$  minus der Ankunftszeit  $e_i$  und minus der Fahrtdauer  $T(a_i,c)$  von Ankunftsort  $a_i$  zu c, sprich  $t^c_{ij}=s_j-T(c,d_j)-e_i-T(a_i,c)$ . Dann ist die Energiedifferenz  $Q^c_{ij}$  zwischen den Fahrten gleich dem negativen Energieverbrauch  $-Q(a_i,c)$  für die Fahrt von  $a_i$  zu c plus der geladenen Energie, die gleich der Ladedauer mal der Ladeleistung  $p_C$  ist, minus der Energie  $Q(c,d_j)$  für die Fahrt von c zu  $d_j$ . Also ist  $Q^c_{ij}=-Q(a_i,c)+p_Ct^c_{ij}-Q(c,d_j)$ . Ein Ladevorgang an Ladestation c lohnt sich, wenn  $Q^c_{ij}>-Q(a_i,d_j)$  ist. Wir bezeichnen die Menge der Ladestationen, an denen zwischen Fahrt  $\tau_i$  und  $\tau_j$  ein Ladevorgang möglich ist und sich lohnt, mit  $C_{ij}$ .

Vor der ersten Fahrgastfahrt  $\tau_i$ , die ein Fahrzeug ausführt, kann beliebig lange an einer Ladestation geladen werden. Wird für die Fahrt von einer Ladestation zum Abfahrtsort  $d_i$  einer Fahrt  $\tau_i$  weniger Energie benötigt als für die Fahrt vom Depot zu  $d_i$  und wird für die Fahrt vom Depot zur Ladestation höchstens so viel Energie benötigt wie die Batteriekapazität des Fahrzeugs, so wissen wir, dass es möglich und mindestens so gut ist, vor der Fahrt  $\tau_i$  zu dieser Ladestation zu fahren und dort den Akku voll zu laden als direkt vom Depot zu  $d_i$  zu fahren. Wir nennen  $\gamma_i$  den Ort aus der Menge der Ladestationen, die vom Depot erreichbar sind, und dem Depot, sodass von  $\gamma_i$  am wenigsten Energie zum Abfahrtsort  $d_i$  von  $\tau_i$  benötigt wird.

Nach der letzten Fahrt  $\tau_i$ , die ein Fahrzeug absolviert, kann wieder beliebig lange an einer Ladestation geladen werden bevor das Fahrzeug ins Depot zurückkehrt. Also ist es nie schlechter nach  $\tau_i$  zu einer Ladestation zu fahren als zum Depot, wenn für die Fahrt weniger Energie benötigt wird. Wir nennen  $\epsilon_i$  den Ort aus der Menge der Ladestationen und dem Depot, zu dem vom Ankunftsort von  $\tau_i$  am wenigsten Energie benötigt wird.

# 4 Komplexität

Sassi und Oulamara [11] haben bereits gezeigt, dass das EVSP schwach  $\mathcal{NP}$ -schwer ist. Wir zeigen in diesem Kapitel, dass die Entscheidungsvariante des EVSP bereits ohne Lademöglichkeit und sich zeitlich überlappende Fahrten selbst für unär kodierte Eingaben W[1]-schwer ist und damit stark  $\mathcal{NP}$ -vollständig ist. Dazu geben wir eine parametrisierte Reduktion von der Entscheidungsvariante von BIN PACKING an, die W[1]-schwer ist. Aus der Reduktion können wir weiter schlussfolgern, dass das EVSP wahrscheinlich nicht in polynomieller Zeit  $\alpha$ -approximiert werden kann für ein  $\alpha$  kleiner  $^{3}/_{2}$ .

Bei BIN PACKING sollen verschieden große Gegenstände in so wenige Behälter einer festen Größe wie möglich "gepackt" werden:

BIN PACKING

**Eingabe:** Gegenstände  $\mathcal{I} = [n]$ , je  $i \in \mathcal{I}$  der Größe  $w_i \in \mathbb{N}$  und Behältergröße  $B \in \mathbb{N}$ .

**Aufgabe:** Finde eine Aufteilung der Gegenstände auf die kleinste Anzahl von paarweise disjunkten Teilmengen  $\mathcal{I}_1, ..., \mathcal{I}_k$ , sodass  $\sum_{j \in \mathcal{I}_i} w_j \leq B$  für alle  $i \in [k]$ .

Die Entscheidungsvariante von BIN PACKING hat die gleiche Eingabe und eine natürliche Zahl k und fragt nach einer Aufteilung auf höchstens k disjunkte Teilmengen.

**Theorem 4.1.** Die Entscheidungsvariante des EVSP ist W[1]-schwer für Parameter k, der Anzahl der benötigten Fahrzeuge, selbst wenn die Eingabe unär kodiert ist.

Beweis. Wir geben eine parametrisierte Reduktion von der Entscheidungsvariante von BIN PACKING an, die W[1]-schwer für Parameter k ist, selbst wenn die Eingabe unär kodiert ist [5].

Für eine gegebene Instanz von BIN PACKING konstruieren wir eine Instanz des EVSP: Wir betrachten einen Ort  $\delta$ , der sowohl das Depot der Fahrzeuge, als auch der Abfahrtsund Ankunftsort aller Fahrgastfahrten ist. Für jeden gegebenen Gegenstand  $i \in \mathcal{I}$  existiert eine Fahrgastfahrt  $\tau_i = (\delta, \delta, s_i, e_i)$ , die an  $\delta$  zur Zeit  $s_i$  startet und an  $\delta$  zur Zeit  $e_i$  endet. Alle Fahrgastfahrten finden hintereinander statt, sprich  $s_i = e_{i-1} \ \forall i \in [2, n]$ . Für eine Fahrt von  $\delta$  zu  $\delta$  wird keine Zeit oder Energie benötigt. Nun ist die für Fahrgastfahrt  $\tau_i$  benötigte Energie gleich der Größe des Gegenstandes i, sprich  $Q^P(\tau_i) = w_i$ , die Batteriekapazität  $\beta$  der Fahrzeuge gleich der Größe eines Behälters B und es gibt keine Möglichkeit, die Batterie zu laden; die Menge der Ladestationen ist leer.

Eine Konstruktion dieser Instanz ist offensichtlich in linearer Zeit möglich. Wir zeigen nun, dass eine Instanz der Entscheidungsvariante von BIN PACKING genau dann eine Ja-Instanz ist, wenn die konstruierte Instanz der Entscheidungsvariante des EVSP eine Ja-Instanz ist:

#### 4 Komplexität

Ist eine Instanz der Entscheidungsvariante von BIN PACKING eine Ja-Instanz, so kann ein Fahrzeug i alle Fahrgastfahrten  $\tau_j$  aller  $j \in \mathcal{I}_i$  ausführen. Denn keine zwei Fahrgastfahrten überlappen sich zeitlich und alle Fahrgastfahrten sind energetisch möglich, da die benötigte Energie für alle Fahrgastfahrten kleiner ist als die Batteriekapazität eines Fahrzeugs.

Ist eine Instanz der Entscheidungsvariante des EVSP eine Ja-Instanz, so sind alle einem Fahrzeug zugeordneten Fahrten energetisch möglich. Denn die Summe der benötigten Energie aller einem Fahrzeug zugeordneten Fahrten ist höchstens gleich B, da die Menge der Ladestationen leer ist. Sei  $\mathcal{I}_i$  die Menge der Indizes j jeder dem Fahrzeug i zugeordneten Fahrt  $\tau_j$ . Dann ist  $\sum_{j\in\mathcal{I}_i}w_j\leq B$  für alle  $i\in[k]$ . Ferner wird jede Fahrt von genau einem Fahrzeug erfüllt. Somit ist  $\bigcup_{i=1}^k\mathcal{I}_i=\mathcal{I}$  und alle Teilmengen sind paarweise disjunkt.

Die Anzahl der benötigten Fahrzeuge k' ist gleich der Anzahl der Behälter k.

Korollar 4.2. Die Entscheidungsvariante des EVSP ist stark  $\mathcal{NP}$ -vollständig.

Aus der Reduktion können wir weiter folgern, dass das EVSP wahrscheinlich nicht in polynomieller Zeit mit einem Faktor kleiner als <sup>3</sup>/<sub>2</sub> approximiert werden kann, da dies für BIN PACKING gilt [15].

**Korollar 4.3.** Das EVSP kann nicht in polynomieller Zeit  $\alpha$ -approximiert werden mit  $\alpha < 3/2$ , es sei denn  $\mathcal{P} = \mathcal{NP}$ .

Beweis. Sei x eine Instanz von BIN PACKING und x' die wie in Theorem 4.1 konstruierte Instanz des EVSP von x, dann ist die Anzahl der Fahrzeuge k', auf die wir die Fahrgastfahrten aufteilen können, gleich der Anzahl der Behälter k, auf die wir die Gegenstände aufteilen können. Die Konstruktion der Probleminstanz des EVSP geschieht in linearer Zeit. Nehmen wir also an, das EVSP wäre in polynomieller Zeit  $\alpha$ -approximierbar mit  $\alpha < 3/2$ , dann wäre genauso BIN PACKING in polynomieller Zeit  $\alpha$ -approximierbar und damit wäre  $\mathcal{P} = \mathcal{N}\mathcal{P}$ .

# 5 Integer Linear Program

Wir geben ein Integer Linear Program (ILP) für das EVSP an. Da aktuelle ILP-Solver schnell gute Lösungen finden, erhalten wir nicht nur eine gute Lösungsmethode, sondern können mit einer optimalen Lösung des ILP die Qualität und Laufzeit anderer Algorithmen messen.

Zuerst geben wir ein ILP für das VSP an. Anschließend erweitern wir dieses ILP, um Instanzen des EVSP zu lösen. Dazu nehmen wir eine konstante Ladeleistung an. Abschließend präsentieren wir eine Datenreduktionsregel, mit der wir die Anzahl zu betrachtender Ladestationen und damit die Anzahl der Variablen und Bedingungen unseres ILP verringern.

### 5.1 ILP für das VSP

Wir geben eine ILP Formulierung an, die eine Variante des Min-Cost-Flow-Problems ist. Dazu führen wir für eine Instanz  $(\mathcal{T}, T)$  des VSP einen gerichteten Graphen G = (V, E) ein. Für jede Fahrgastfahrt  $\tau \in \mathcal{T}$  fügen wir einen gleichnamigen Knoten zu V hinzu. Zusätzlich fügen wir einen Depot-Knoten  $\delta^+$  hinzu, an dem alle Fahrzeuge starten und einen Knoten  $\delta^-$ , an dem alle Fahrzeuge enden. Denn jedes Fahrzeug startet am Depot, absolviert eine Menge an Fahrten und endet wieder am Depot. Im Graphen ist daher der Startknoten am Depot  $\delta^+$  mit allen Fahrten durch eine Kante verbunden und jede Fahrt mit dem Endknoten  $\delta^-$ . Die Menge dieser Kanten nennen wir  $E_\delta$ . Kann ein Fahrzeug nach einer Fahrt  $\tau_i$  eine andere Fahrt  $\tau_j$  absolvieren, sind die Knoten  $\tau_i$  und  $\tau_j$  im Graph durch eine Kante verbunden. Also wenn  $\tau_i$  vor  $\tau_j$  stattfindet, sprich  $s_i \leq s_j$  ist und sich beide Fahrten nicht überlappen, sprich  $\tau_i \cap \tau_j = \emptyset$  ist, existiert eine Kante  $(\tau_i, \tau_j)$ . Die Menge dieser Kanten nennen wir  $E_{\mathcal{T}}$ . Zusammen mit  $E_\delta$  bilden sie die Menge aller Kanten E. Abbildung 5.1 zeigt den Graphen für unsere Beispielinstanz aus Abschnitt 3.1.

Wir führen für jede Kante  $(\delta^+, \tau_i) \in E_\delta$  eine binäre Variable  $x_{\delta^+\tau_i} \in \{0, 1\}$  ein, die genau dann gleich eins ist, wenn ein Fahrzeug vom Depot zur Fahrt  $\tau_i$  ausrückt, für jede Kante  $(\tau_i, \tau_j) \in E_{\mathcal{T}}$  eine binäre Variable  $x_{\tau_i \tau_j} \in \{0, 1\}$  ein, die genau dann gleich eins ist, wenn ein Fahrzeug nach Fahrt  $\tau_i$  Fahrt  $\tau_j$  ausführt und für jede Kante  $(\tau_i, \delta^-) \in E_\delta$  eine binäre Variable  $x_{\tau_i \delta^-} \in \{0, 1\}$  ein, die genau dann gleich eins ist, wenn ein Fahrzeug nach  $\tau_i$  zum Depot zurückkehrt.

Die Anzahl aller benötigten Fahrzeuge ist die Summe aller vom Depot ausrückenden Fahrzeuge und damit gleich der Summe der Variablen  $x_{\delta^+\tau_i}$  über alle Nachbarknoten  $\tau_i \in N^+(\delta^+)$  vom Startknoten  $\delta^+$ . Ziel ist es, die minimale Anzahl an Fahrzeugen zu

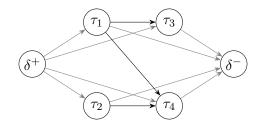


Abbildung 5.1: Graph für die Beispielinstanz aus Abschnitt 3.1

finden. Damit lautet die Zielfunktion

$$\min \sum_{\tau_i \in N^+(\delta^+)} x_{\delta^+ \tau_i}. \tag{5.1}$$

Wird eine Fahrt  $\tau_j$  von genau einem Fahrzeug bedient, ist die Summe der Variablen  $x_{v\tau_j}$  über alle eingehenden Nachbarknoten  $v \in N^-(\tau_j)$  gleich eins. Führt ein Fahrzeug nach einer Fahrt  $\tau_j$  eine andere Fahrt aus oder kehrt ins Depot zurück, ist die Summe der Variablen  $x_{\tau_j w}$  über alle ausgehenden Nachbarknoten  $w \in N^+(\tau_j)$  auch gleich eins. Nachdem ein Fahrzeug eine Fahrt ausgeführt hat, absolviert es entweder eine andere Fahrt oder kehrt ins Depot zurück. Jede Fahrt muss von genau einem Fahrzeug bedient werden. Also muss gelten, dass

$$\sum_{v \in N^{-}(\tau_{j})} x_{v\tau_{j}} = \sum_{w \in N^{+}(\tau_{j})} x_{\tau_{j}w} = 1 \qquad \forall \tau_{j} \in \mathcal{T}.$$
 (5.2)

Lemma 5.1. Gleichungen (5.1) und (5.2) sind eine ILP-Formulierung für das VSP.

Beweis. Offensichtlich ist (5.1) genau die Zielfunktion des VSP.

Wir zeigen, dass ein Umlaufplan  $\Psi$  genau dann gültig ist, wenn die Variablen x des ILP (5.2) erfüllen.

Aus x erhält man die Lösung  $\Psi$  für die Instanz des VSP wie folgt: Ein  $\delta^+$ - $\delta^-$ -Pfad mit Fluss gleich eins ist ein Pfad  $(\delta^+, \tau_1, ..., \tau_n, \delta^-)$ , für den gilt  $x_{\tau_i \tau_{i+1}} = 1 \ \forall i \in [n-1]$ . Durch (5.2) ist jede Fahrgastfahrt offensichtlich in genau einem  $\delta^+$ - $\delta^-$ -Pfad mit Fluss gleich eins enthalten. Alle Fahrgastfahrten auf so einem Pfad sind ein Umlauf und alle Umläufe der Umlaufplan  $\Psi$ . Damit ist jede Fahrgastfahrt in genau einem Umlauf enthalten. Sind zwei Fahrgastfahrten in einem Umlauf enthalten, so sind sie über eine Kante  $(\tau_i, \tau_j) \in E_{\mathcal{T}}$  verbunden. Für jede dieser Kanten gilt  $\tau_i \cap \tau_j = \emptyset$ . Damit überlappen sich keine zwei Fahrgastfahrten eines Umlaufs und jeder Umlauf ist per Definition 3.2 möglich. Also ist der Umlaufplan  $\Psi$  per Definition 3.3 gültig.

Ist ein Umlaufplan gültig, so setzen wir für jeden Umlauf bestehend aus den Fahrten  $\tau_1, ..., \tau_n$  die Variable  $x_{\delta^+\tau_1} = 1$ ,  $x_{\tau_i\tau_{i+1}} = 1$  für alle  $i \in [n-1]$  und  $x_{\tau_n\delta^-} = 1$ . Alle anderen Werte von x setzen wir gleich null. Dann ist (5.2) offensichtlich erfüllt.

Diesen Ansatz erweitern wir nun, um ein ILP für das Electric VSP aufzustellen.

### 5.2 ILP für das EVSP

Um zu prüfen, ob ein Umlauf auch von einem elektrischen Fahrzeug absolviert werden kann, benötigen wir weitere Variablen, die den Ladezustand eines Fahrzeugs angeben. Der Ladezustand hängt von der Wahl der Fahrgastfahrten ab. Bei dem eben beschriebenen ILP für das VSP ist der Fluss über jede Fahrgastfahrt gleich eins. Damit wissen wir für jede Fahrgastfahrt, ob, und wenn ja, welche Fahrgastfahrt davor von einem Fahrzeug ausgeführt wurde. Führen wir für jede Fahrgastfahrt eine Variable für den Ladezustand zu Beginn der Fahrt ein, können wir Bedingungen an den Ladezustand zu Beginn einer darauf folgenden Fahrgastfahrt stellen. So können wir sicherstellen, dass jedes elektrische Fahrzeug immer einen gültigen Ladezustand besitzt.

Annahme. Jede Bedingung eines linearen Programms ist eine lineare (Un-)Gleichung. Wollen wir die Bedingungen für den Ladezustand zu Beginn einer Fahrgastfahrt in Abhängigkeit des Ladezustands der Fahrt davor als lineare Gleichung angeben, stellen sich zwei Probleme, wenn wir das Laden der Batterie eines Fahrzeugs zwischen zwei Fahrgastfahrten betrachten:

Der Energiebetrag, der in einem Zeitraum  $\Delta t$  geladen wird, hängt durch Ladefunktion  $Q^C(q, \Delta t)$  vom Ladezustand zu Beginn des Ladevorgangs q ab. Ist die Funktion nicht stetig oder linear, so hängt der Ladezustand zu Beginn einer Fahrgastfahrt nicht linear vom Ladezustand der Fahrt davor ab, wenn zwischen den Fahrten geladen wird. Für die Formulierung eines ILP benötigen wir also eine stetige und lineare Ladefunktion. Deshalb nehmen wir die stetig auf  $[0, \beta] \times \mathbb{N} \to [0, \beta]$  definierte lineare Ladefunktion

$$Q^{C}(q, \Delta t) = \min(q + p_{C} \Delta t, \beta)$$

mit der konstanten Ladeleistung  $p_C$  an. Diese Annahme ist durchaus realistisch, denn die Ladeleistung kann im Bereich bis 80% der Batteriekapazität als konstant angenommen werden [2].

Wir nennen das Problem mit diesen Annahmen LINEAR EVSP (LEVSP). Bemerke, dass alle Resultate aus Kapitel 4 auch für das LEVSP gelten.

Wir nutzen die Variablen x, die Zielfunktion (5.1) und Bedingung (5.2) des ILP für das VSP und führen weitere Variablen und Bedingungen ein, die sicherstellen, dass jeder Umlauf eines elektrischen Fahrzeugs per Definition 3.4 möglich ist.

**Variablen.** Für jede Fahrgastfahrt  $\tau_i \in \mathcal{T}$  führen wir eine reelle Variable  $q_i \in [0, \beta]$  ein, die den Ladezustand eines Fahrzeugs zwischen 0 und Batteriekapazität  $\beta$  zu Beginn von  $\tau_i$  angibt. Für alle Fahrtenpaare  $(\tau_i, \tau_j) \in E_{\mathcal{T}}$  und Ladestationen  $c \in \mathcal{C}_{ij}$  sei  $y_{ij}^c \in \{0, 1\}$  eine binäre Variable, die genau dann gleich eins ist, wenn zwischen Fahrt  $\tau_i$  und  $\tau_j$  an Ladestation  $c \in \mathcal{C}_{ij}$  geladen wird.

**Bedingungen.** Für den Fall, dass ein Fahrzeug als erste Fahrt  $\tau_i$  ausführt, also  $x_{\delta^+\tau_i}=1$  ist, gilt für den Ladezustand vor der ersten Fahrt  $q_i \leq \beta - Q(\gamma_i, d_i)$ . Zur Erinnerung:  $\beta$  ist die Batteriekapazität,  $\gamma_i$  die nächste Ladestation von  $d_i$  oder das Depot und  $Q(\gamma_i, d_i)$ 

die benötigte Energie für die Fahrt von  $\gamma_i$  zum Abfahrtsort  $d_i$ . Falls  $x_{\delta^+\tau_i}=0$  ist, so ist  $q_i$  ohnehin höchstens gleich  $\beta$ . Also gilt

$$q_i \le \beta - x_{\delta^+ \tau_i} Q(\gamma_i, d_i) \qquad \forall (\delta^+, \tau_i) \in E_{\delta}.$$
 (5.3)

Seien  $\tau_i$  und  $\tau_j$  zwei Fahrgastfahrten, die hintereinander ausgeführt werden können und zwischen denen kein Ladevorgang möglich ist, also  $C_{ij} = \emptyset$  ist. Wird Fahrt  $\tau_i$  nach  $\tau_j$  ausgeführt, dann ist  $x_{\tau_i\tau_j} = 1$  und der Ladezustand  $q_j$  vor Fahrt  $\tau_j$  höchstens so groß wie der Ladezustand  $q_i$  vor Fahrt  $\tau_i$  abzüglich der für die Fahrt  $\tau_i$  benötigten Energie  $Q^P(\tau_i)$  und der für die Leerfahrt zwischen beiden Fahrgastfahrten benötigten Energie  $Q(a_i, d_j)$ , sprich  $q_j \leq q_i - x_{\tau_i\tau_j} (Q^P(\tau_i) - Q(a_i, d_j))$ . Damit die Ungleichung auch gilt, wenn Fahrt  $\tau_j$  nicht nach  $\tau_i$  ausgeführt wird, also wenn  $x_{\tau_i\tau_j} = 0$  ist, fügen wir in diesem Fall einen großen Summanden  $(1 - x_{\tau_i\tau_j})K$  mit einem beliebigen, ausreichend großem  $K \geq \beta$  hinzu:

$$q_j \le q_i - x_{\tau_i \tau_j} \left( Q^P(\tau_i) - Q(a_i, d_j) \right) + (1 - x_{\tau_i \tau_j}) K \qquad \forall (\tau_i, \tau_j) \in E_{\mathcal{T}} : \mathcal{C}_{ij} = \emptyset. \tag{5.4}$$

Ist  $C_{ij} \neq \emptyset$  und lädt ein Fahrzeug zwischen zwei Fahrten  $\tau_i$  und  $\tau_j$  an einer Ladestation  $c \in C_{ij}$ , so ist  $y_{ij}^c = 1$ . Dann muss der Ladezustand  $q_i$  vor Fahrt  $\tau_i$  abzüglich der für die Fahrgastfahrt benötigten Energie  $Q^P(\tau_i)$  und der für die anschließende Fahrt zur Ladestation benötigten Energie  $Q(a_i,c)$  mindestens null sein. Falls  $y_{ij}^c = 0$  ist, so muss  $q_i$  abzüglich  $Q^P(\tau_i)$  ohnehin mindestens null sein. Also gilt, dass  $q_i - Q^P(\tau_i) - y_{ij}^c Q(a_i,c) \geq 0$  ist. Damit die Ungleichung erfüllt ist, wenn  $\tau_j$  nicht nach  $\tau_i$  ausgeführt wird, also  $x_{\tau_i\tau_j} = 0$  ist, fügen wir wieder den Summanden  $(1 - x_{\tau_i\tau_j})K$  hinzu:

$$q_i - Q^P(\tau_i) - y_{ij}^c Q(a_i, c) + (1 - x_{\tau_i \tau_j}) K \ge 0 \qquad \forall (\tau_i, \tau_j) \in E_{\mathcal{T}} \ \forall c \in \mathcal{C}_{ij}.$$
 (5.5)

Weiter ist nach einem Ladevorgang zwischen zwei Fahrten  $\tau_i$  und  $\tau_j$ , die hintereinander von einem Fahrzeug ausgeführt werden, der Ladezustand  $q_j$  vor Fahrt  $\tau_j$  höchstens so groß wie der Ladezustand  $q_i$  vor Fahrt  $\tau_i$  abzüglich der für die Fahrt  $\tau_i$  benötigten Energie  $Q^P(\tau_i)$  plus der Energiedifferenz  $Q^c_{ij}$  zwischen den Fahrten nach dem Ladevorgang an Ladestation c. Die Fälle, wenn  $x_{\tau_i\tau_j} = 0$  oder  $y^c_{ij} = 0$  ist, werden wie in (5.4) und (5.5) behandelt:

$$q_{j} \leq q_{i} - x_{\tau_{i}\tau_{j}}Q^{P}(\tau_{i}) + y_{ij}^{c}Q_{ij}^{c} + (1 - y_{ij}^{c})K + (1 - x_{\tau_{i}\tau_{j}})K \quad \forall (\tau_{i}, \tau_{j}) \in E_{\mathcal{T}} \ \forall c \in \mathcal{C}_{ij}.$$
(5.6)

Da die Batterie nur maximal bis zu ihrer Kapazität  $\beta$  geladen werden kann, ist nach einem Ladevorgang zwischen zwei Fahrten  $\tau_i$  und  $\tau_j$  der Ladezustand  $q_j$  bei Fahrtbeginn von  $\tau_j$  höchstens gleich  $\beta$  abzüglich der benötigten Energie für die Fahrt von der Ladestation zum Abfahrtsort  $d_j$ . Ist  $y_{ij}^c = 0$ , so ist  $q_j$  ohnehin kleiner oder gleich  $\beta$ . Also gilt, dass  $q_j \leq \beta - y_{ij}^c Q(c, d_j)$  ist. Damit auch diese Ungleichung erfüllt ist, wenn  $\tau_j$  nicht nach  $\tau_i$  ausgeführt wird, fügen wir wieder den Summanden  $(1 - x_{\tau_i \tau_j})K$  hinzu:

$$q_i \le \beta - y_{ij}^c Q(c, d_i) + (1 - x_{\tau_i \tau_i}) K \qquad \forall (\tau_i, \tau_i) \in E_{\mathcal{T}} \ \forall c \in \mathcal{C}_{ij}. \tag{5.7}$$

Damit sichergestellt ist, dass auch, wenn möglich, tatsächlich an einer Ladestation zwischen zwei Fahrten geladen wird und (5.6) nicht durch  $y_{ij}^c = 0$  für alle  $c \in \mathcal{C}_{ij}$  trivial erfüllt ist, muss gelten

$$\sum_{c \in \mathcal{C}_{ij}} y_{ij}^c = 1 \qquad \forall (\tau_i, \tau_j) \in E_{\mathcal{T}} : \mathcal{C}_{ij} \neq \emptyset.$$
 (5.8)

Fährt ein Fahrzeug nach Fahrt  $\tau_i$  zurück ins Depot, so ist  $x_{\tau_i\delta^-}=1$ . Dann muss nach  $\tau_i$  noch genug Energie verfügbar sein, um zur nächsten Ladestation vom Ankunftsort von  $\tau_i$  oder dem Depot  $\epsilon_i$  fahren zu können. Also muss gelten  $q_i - x_{\tau_i\delta^-}(Q^P(\tau_i) + Q(a_i, \epsilon_i)) \geq 0$ . Falls  $x_{\tau_i\delta^-}=0$  ist, so ist  $q_i$  ohnehin mindestens null. Also gilt

$$q_i - x_{\tau_i \delta^-} (Q^P(\tau_i) + Q(a_i, \epsilon_i)) \ge 0$$
  $\forall \tau_i \in \mathcal{T}.$ 

Da dies sowieso für den Ladezustand vor jeder Fahrt gelten muss, können wir die Gleichung durch folgende einfachere Gleichung ersetzten:

$$q_i - Q^P(\tau_i) - Q(a_i, \epsilon_i) \ge 0$$
  $\forall \tau_i \in \mathcal{T}.$  (5.9)

Nun haben wir Folgendes gezeigt: Existiert eine gültige Lösung für eine Instanz des LEVSP, sodass immer zwischen zwei Fahrten geladen wird, wenn das möglich ist, dann sind (5.2)–(5.9) erfüllt. Enthält der Umlauf einer gültigen Lösung zwischen zwei Fahrten  $\tau_i$  und  $\tau_j$  keinen Ladevorgang und ist ein Ladevorgang  $\zeta_{ij}^c$  zwischen den Fahrten an Ladestation c möglich, so können wir ohne Beschränkung der Allgemeinheit  $\zeta_{ij}^c$  zum Umlauf hinzufügen. Denn der Umlauf ist offensichtlich weiter möglich.

**Lemma 5.2.** Existiert eine gültige Lösung für eine Instanz des LEVSP, dann existiert für die Variablen x, y und q eine gültige Belegung, sodass (5.2) – (5.9) erfüllt sind.

Nun zeigen wir, dass auch die Rückrichtung gilt und damit das konstruierte ILP korrekt ist.

**Lemma 5.3.** Es existiert eine gültige Lösung für eine Instanz des LEVSP, wenn (5.2)–(5.9) erfüllt sind.

Beweis. Nach Lemma 5.1 sind alle Fahrgastfahrten  $\tau_1, ..., \tau_n$  auf einem  $\delta^+$ - $\delta^-$ -Pfad mit Fluss gleich eins ein Umlauf und alle Umläufe eine gültige Lösung für die Instanz des VSP, wenn (5.2) erfüllt ist. Damit sind per Definition 3.4 und 3.5 alle Umläufe eine gültige Lösung für die Instanz des EVSP, wenn für jeden Umlauf der Ladezustand immer mindestens null und höchstes gleich der Batteriekapazität  $\beta$  ist.

Die kleinstmöglichen Ladezustände sind vor den Ladevorgängen bei Ankunft an einer Ladestation sowie der Ladezustand am Ende des Umlaufs bei Ankunft am Depot. Für einen Umlauf mit den Fahrten  $\tau_1, ... \tau_n$  zeigen wir per Induktion über i von n bis 1, dass der Ladezustand  $q_i$  vor Fahrt  $\tau_i$  so groß ist, dass der Ladezustand vor jedem Ladevorgang zwischen  $\tau_i$  und  $\tau_n$  und der Ladezustand bei Ankunft am Depot mindestens null ist.

Wegen (5.9) ist  $q_n$  so groß, dass der Ladezustand bei Ankunft an  $\epsilon_n$  mindestens null ist. Falls  $\epsilon_n$  nicht das Depot ist, so folgt aus der Definition von  $\epsilon_n$ , dass die benötigte Energie von  $\epsilon_n$  zum Depot höchstens  $\beta$  ist. Nach einem vollständigen Ladevorgang ist also der Ladezustand bei Ankunft am Depot mindestens null. Dies ist unser Induktionsanfang. Nehmen wir an, dass der Ladezustand  $q_i$  vor Fahrt  $\tau_i$  so groß ist, dass der Ladezustand vor jedem Ladevorgang zwischen  $\tau_i$  und  $\tau_n$  mindestens null ist, dann ist  $q_{i-1}$  so groß, dass der Ladezustand vor jedem Ladevorgang zwischen  $\tau_{i-1}$  und  $\tau_n$  mindestens null ist:

Wenn  $C_{i-1,i} \neq \emptyset$  ist, dann laden wir wegen (5.8) an einer Ladestation c. Dann gilt wegen (5.5), dass der Ladezustand zu Beginn des Ladevorgangs an c mindestens null ist. Wegen (5.6) gilt, dass  $q_{i-1} \geq q_i + Q^P(\tau_{i-1}) - Q^c_{i-1,i}$  ist. Wir haben angenommen, dass  $q_i$  so groß ist, dass der Ladezustand vor jedem Ladevorgang zwischen  $\tau_i$  und  $\tau_n$  mindestens null ist Daraus können wir jetzt folgern, dass  $q_{i-1}$  so groß ist, dass dies auch für  $\tau_{i-1}$  bis  $\tau_n$  gilt. Wenn  $C_{i-1,i} = \emptyset$  ist, dann laden wir zwischen  $\tau_{i-1}$  und  $\tau_i$  nicht. Es gilt wegen (5.4), dass  $q_{i-1} \geq q_i + Q^P(\tau_{i-1}) + Q(a_{i-1}, d_i)$  ist. Daraus können wir die gleiche Schlussfolgerung ziehen wie in dem Fall, wenn  $C_{i-1,i} \neq \emptyset$  ist.

Also wissen wir, dass  $q_1$  so groß ist, dass der Ladezustand zwischen  $\tau_1$  und  $\tau_n$  und zuletzt am Depot immer mindestens null ist.

Es bleibt zu zeigen, dass der Ladezustand immer höchstens gleich  $\beta$  ist. Die größten möglichen Ladezustände sind nach den Ladevorgängen und nach Abfahrt am Depot. Wir zeigen per Induktion über i von 1 bis n, dass der Ladezustand nach jedem Ladevorgang zwischen  $\tau_1$  und  $\tau_i$  sowie der Ladezustand bei Abfahrt am Depot höchstens gleich  $\beta$  ist.

Wegen (5.3) ist der Ladezustand bei Abfahrt an  $\gamma_1$  gleich  $\beta$ . Ist  $\gamma_1$  nicht das Depot, so folgt aus der Definition von  $\gamma_1$ , dass der Ladezustand bei Abfahrt am Depot höchstens gleich  $\beta$  sein muss. Dies ist unser Induktionsanfang. Nehmen wir an, dass der Ladezustand nach jedem Ladevorgang zwischen  $\tau_1$  und  $\tau_i$  höchstens gleich  $\beta$  ist, dann ist der Ladezustand nach jedem Ladevorgang zwischen  $\tau_1$  und  $\tau_{i+1}$  höchstens gleich  $\beta$ : Ist  $C_{i,i+1} \neq \emptyset$ , so wird wegen (5.8) an einer Ladestation c geladen. Wegen (5.7) ist der Ladezustand nach dem Ladevorgang höchstens gleich  $\beta$ . Mit unserer Annahme können wir folgern, dass der Ladezustand nach jedem Ladevorgang zwischen  $\tau_1$  und  $\tau_{i+1}$  höchstens gleich  $\beta$  ist. Also ist der Ladezustand während eines Umlaufs immer höchstens gleich  $\beta$ . Das schließt unseren Beweis ab.

Aus Lemma 5.2 und 5.3 und der Zielfunktion (5.1) des LEVSP folgt

**Theorem 5.4.** Gleichungen (5.1)–(5.9) sind eine ILP-Formulierung für das LEVSP.

**Datenreduktion.** Wir stellen eine Reduktionsregel für das EVSP auf mit dem Ziel, die Anzahl der zu betrachtenden Ladestationen zwischen zwei Fahrgastfahrten  $\tau_i$  und  $\tau_j$  zu verringern.

Sei  $q_{ij}^c$  der Ladezustand bei Fahrtbeginn von  $\tau_j$ , nachdem nach Fahrt  $\tau_i$  so lange wie möglich an Ladestation c geladen wurde. Für die Fahrt vom Ankunftsort  $a_i$  der Fahrt  $\tau_i$  zu c wird Energie  $Q(a_i,c)$  benötigt. Existiert eine Ladestation c', sodass  $Q(a_i,c') < Q(a_i,c)$  und  $q_{ij}^{c'} > q_{ij}^c$  ist, so wissen wir, dass es immer besser ist an c' zu laden.

Reduktionsregel 5.1. Sei  $\tau_i, \tau_j \in \mathcal{T}$  und  $c \in \mathcal{C}_{ij}$  eine Ladestation. Existiert eine Ladestation  $c' \in \mathcal{C}_{ij}$  mit  $Q(a_i, c') < Q(a_i, c)$  und  $q_{ij}^{c'} > q_{ij}^c$ , dann lösche c aus der Menge  $\mathcal{C}_{ij}$ .

**Lemma 5.5.** Reduktionsregel 5.1 ist korrekt und kann in  $\mathcal{O}(|\mathcal{T}|^2|\mathcal{C}|^2)$  Zeit auf alle Fahrgastfahrten  $\mathcal{T}$  und Ladestationen  $\mathcal{C}$  angewandt werden.

Beweis. Enthält eine Lösung zwischen den Fahrten  $\tau_i$  und  $\tau_j$  einen Ladevorgang an Ladestation c, so kann man diesen durch einen Ladevorgang an Ladestation c' ersetzten. Denn der Ladevorgang an c' ist möglich, da der Ladezustand bei Ankunft an c' größer ist als bei Ankunft an c. Außerdem sind die Fahrgastfahrt  $\tau_j$  und alle folgenden Fahrgastfahrten möglich, da  $q_{ij}^{c'} > q_{ij}^c$  ist.

Um die Regel anwenden zu können, müssen wir für zwei Ladestationen  $c,c' \in \mathcal{C}_{ij}$  bestimmen, ob  $q_{ij}^{c'} > q_{ij}^c$  ist. Der Ladezustand nach der Fahrgastfahrt  $\tau_i$  bei Ankunft an Ladestation c ist der Ladezustand  $q_i$  vor  $\tau_i$  abzüglich der für die Fahrgastfahrt benötigten Energie  $Q^P(\tau_i)$  und der für die anschließende Fahrt vom Ankunftsort  $a_i$  zu c benötigten Energie  $Q(a_i,c)$ . Ist  $Q(a_i,c) > Q(a_i,c')$ , so ist der Ladezustand bei Ankunft an Ladestation c kleiner als an c'. Nach dem Ladevorgang ist der Ladezustand gleich dem Ladezustand bei Ankunft an der Ladestation plus der Ladeleistung  $p_C$  mal dem Ladezeitraum  $t_{ij}^c$  und höchstens gleich der Batteriekapazität  $\beta$ . Ist die Ladedauer  $t_{ij}^c \leq t_{ij}^{c'}$ , ist also  $T(a_i,c)+T(c,d_j)\geq T(a_i,c')+T(c',d_j)$ , so ist der Ladezustand nach dem Ladevorgang an c kleiner als der Ladezustand nach dem Ladevorgang an c kleiner als der Ladezustand nach dem Ladevorgang abzüglich der benötigten Energie für die Fahrt von c zum Abfahrtsort  $d_j$ . Ist schließlich  $Q(c',d_j)< Q(c,d_j)$ , so ist  $q_{ij}^{c'}>q_{ij}^c$ . Also können wir alle Ladestationen c aus  $\mathcal{C}_{ij}$  entfernen, für die gilt: Es existiert eine Ladestation  $c'\in\mathcal{C}_{ij}$ , sodass  $Q(a_i,c')< Q(a_i,c)$ ,  $T(a_i,c)+T(c,d_j)\geq T(a_i,c')+T(c',d_j)$  und  $Q(c',d_j)< Q(c,d_j)$  ist.

Das ist in  $\mathcal{O}(|\mathcal{C}_{ij}|^2)$  Zeit möglich. Die Regel muss für jedes Paar von Fahrgastfahrten angewandt werden. Das benötigt  $\mathcal{O}(|\mathcal{T}|^2|\mathcal{C}|^2)$  Zeit.

# 6 Greedy-Algorithmus

In diesem Kapitel verlassen wir das Streben nach Optimalität, sondern versuchen schnell eine gute Lösung für das EVSP zu finden. Wir präsentieren einen Greedy-Algorithmus, der einfach und schnell ist, für dessen Lösungen es aber keine Qualitätsgarantie gibt. Eine Eigenschaft dieses Algorithmus ist, dass er im Gegensatz zum angegebenen ILP in Abschnitt 5.2 auch nichtlineares Laden der Batterie berücksichtigen kann.

Zuerst beschreiben wir die Idee für den Algorithmus und unsere Annahmen. Dann beschreiben wir den Algorithmus, zeigen die Laufzeit und dass er korrekt ist. Abschließend präsentieren wir eine EVSP-Instanz, für die der Greedy-Algorithmus keine gute Lösung ausgibt.

**Idee.** Unser Greedy-Ansatz ist inspiriert von dem Algorithmus für INTERVAL PARTITIONING von Gupta, Lee und Leung [4]. Dabei sollen Jobs, die durch eine Start-und Endzeit definiert sind, sprich Intervalle, auf eine kleinstmögliche Menge von Maschinen "partitioniert" werden, sodass sich alle einer Maschine zugeordneten Intervalle nicht überlappen.

Interval Partitioning

**Eingabe:** Jobs J. Jeder Job  $(s_i, f_i) \in J$  hat eine Startzeit  $s_i \in \mathbb{N}$  und Endzeit  $f_i \in \mathbb{N}$ 

**Frage:** Gibt es k paarweise disjunkte Teilmengen  $M_1, ..., M_k$ , genannt Maschinen, die vereinigt J sind, sodass für alle  $(s_i, f_i), (s_j, f_j) \in M_m$   $[s_i, f_i] \cap [s_j, f_j] = \emptyset$  für alle  $m \in [k]$ ?

Jobs sind bei uns die Fahrgastfahrten und Maschinen die Fahrzeuge. Man kann also das VSP auch als Interval Partitioning Problem mit einer Distanz zwischen jedem Intervall-Paar sehen [7]. Ist der Abfahrts-und Ankunftsort jeder Fahrgastfahrt derselbe oder wird für jede Fahrt zwischen den Fahrgastfahrten keine Zeit benötigt, so ist das VSP äquivalent zu Interval Partitioning. Das löst der Greedy-Algorithmus [4] in  $\mathcal{O}(n \log n)$  Zeit optimal wie folgt: "ordne die Jobs den Maschinen sortiert nach Startzeit zu, nutze dabei eine bereits genutzte Maschine, wann immer möglich." [7].

**Annahmen.** Wir nehmen an, dass die Batteriekapazität der Fahrzeuge groß genug ist, um vom Depot zu jeder Ladestation und von jeder Ladestation zum Depot fahren zu können, sprich, für alle  $c \in \mathcal{C}$  gilt, dass  $Q(\delta, c) \leq \beta$  und  $Q(c, \delta) \leq \beta$  ist. Diese Annahmen sind für den öffentlichen Nahverkehr einer Stadt realistisch.

#### Algorithmus 1 Pseudocode des Greedy-Algorithmus für das EVSP

```
1: sortiere Fahrgastfahrten \mathcal{T} aufsteigend nach Abfahrtszeit
 2: Umlaufplan \Omega \leftarrow \emptyset
 3: for Fahrgastfahrt \tau \in \mathcal{T} do
 4:
            \Omega_{\text{poss}} \leftarrow \emptyset
            for \omega \in \Omega do
 5:
                 if letzte Fahrgastfahrt \tau_n von \omega überlappt sich zeitlich nicht mit \tau then
 6:
 7:
                        berechne bestmöglichen Ladevorgang \zeta_{\tau_n\tau} und Ladezustand q_{\tau}^{\omega}
                       if \omega \cup \{\zeta_{\tau_n\tau}, \tau, \zeta_{\tau\delta}\} energetisch möglich then
 8:
 9:
                             \Omega_{\text{poss}} \leftarrow \Omega_{\text{poss}} \cup \omega
10:
                       end if
                  end if
11:
            end for
12:
            if \Omega_{\text{poss}} \neq \emptyset then
13:
                 \omega \leftarrow \operatorname{argmax}_{\omega \in \Omega_{\operatorname{poss}}} q_{\tau}^{\omega}\omega \leftarrow \omega \cup \{\zeta_{\tau_n \tau}, \tau\}
14:
15:
16:
                  if Umlauf \{\zeta_{\delta\tau}, \tau, \zeta_{\tau\delta}\} energetisch möglich then
17:
                       füge neuen Umlauf \omega' = \{\zeta_{\delta\tau}, \tau\} zu \Omega hinzu
18:
19:
                  else
                        return "Instanz nicht lösbar"
20:
21:
                  end if
            end if
22:
23: end for
24: füge jedem Umlauf \omega \in \Omega Ladevorgang \zeta_{\tau_n \delta} nach der letzten Fahrgastfahrt \tau_n hinzu
```

Algorithmus. Als Erstes sortieren wir die Fahrgastfahrten aufsteigend nach der Abfahrtszeit, um sie dann Umläufen zuzuordnen. Dabei ordnen wir eine Fahrgastfahrt  $\tau$  immer einem existierenden Umlauf zu, wenn das möglich ist. Das ist der Fall, wenn sich $\tau$  nicht mit der letzten Fahrgastfahrt  $\tau_n$  eines Umlaufs  $\omega$  zeitlich überlappt und der Umlauf mit  $\tau$  energetisch möglich ist. Dazu prüfen wir für jede Ladestation, ob zwischen  $\tau_n$  und  $\tau$  ein Ladevorgang dort möglich ist. Gibt es mögliche Ladevorgänge, dann sei  $\zeta_{\tau_n\tau}$  der beste mögliche Ladevorgang, nach dem der Ladezustand  $q_{\tau}^{\omega}$  zu Beginn von  $\tau$  am größten ist. Besitzt das Fahrzeug, das  $\omega$  ausführt, nach  $\tau$  noch genug Energie, um zu einer Ladestation oder dem Depot fahren zu können, so ist der Umlauf  $\omega \cup \{\zeta_{\tau_n\tau}, \tau, \zeta_{\tau\delta}\}$  energetisch möglich. Falls es eine Ladestation  $\epsilon$  gibt, zu der vom Ankunftsort von  $\tau$  weniger Energie gebraucht wird als zum Depot, dann sei  $\zeta_{\tau\delta}$  ein vollständiger Ladevorgang nach  $\tau$  an  $\epsilon$ .

Gibt es mehrere Umläufe, zu denen wir  $\tau$  hinzufügen können, wählen wir "gierig" den Umlauf  $\omega$  mit dem Fahrzeug, dessen Ladezustand zu Beginn von  $\tau$  am größten ist, sodass das Fahrzeug möglichst viele weitere Fahrgastfahrten absolvieren kann und fügen  $\zeta_{\tau_n\tau}$  und  $\tau$  zu  $\omega$  hinzu.

Gibt es keinen Umlauf, der mit  $\tau$  kompatibel ist, prüfen wir, ob ein Umlauf  $\{\zeta_{\delta\tau}, \tau, \zeta_{\tau\delta}\}$ 

energetisch möglich ist. Falls es eine Ladestation  $\gamma$  gibt, von der zum Abfahrtort von  $\tau$  weniger Energie gebraucht wird als vom Depot, dann sei  $\zeta_{\tau\delta}$  ein vollständiger Ladevorgang vor  $\tau$  an  $\gamma$ . Falls dieser Umlauf nicht energetisch möglich ist, so ist die Instanz des Problems nicht lösbar. Sonst fügen wir einen neuen Umlauf  $\omega' = \{\zeta_{\delta\tau}, \tau\}$  hinzu.

Nachdem jede Fahrgastfahrt einem Umlauf zugewiesen worden sind, fügen wir jedem Umlauf Ladevorgang  $\zeta_{\tau_n\delta}$  nach der letzten Fahrgastfahrt  $\tau_n$  hinzu. Der Pseudocode der beschriebenen Heuristik ist Algorithmus 1.

Wir zeigen nun, dass unser Algorithmus korrekt ist.

#### Lemma 6.1. Liefert Algorithmus 1 eine Lösung für das EVSP, so ist sie gültig.

Beweis. Wird gemäß Algorithmus 1 eine Fahrgastfahrt  $\tau$  und ein Ladevorgang  $\zeta_{\delta\tau}$  einem neuen Umlauf  $\omega'$  hinzugefügt (Zeile 18), so ist  $\omega' \cup \{\zeta_{\tau\delta}\}$  energetisch möglich.

Wird  $\tau$  einem bestehenden Umlauf  $\omega \in \Omega$  hinzugefügt, so überlappt sich  $\tau$  nicht mit der letzten Fahrgastfahrt  $\tau_n$  von  $\omega$ . Da die Fahrgastfahrten aufsteigend nach der Abfahrtszeit sortiert den Umläufen hinzugefügt werden, überlappt sich  $\tau$  auch mit keiner Fahrgastfahrt aus  $\omega$ . Wird  $\zeta_{\tau_n\tau}$  und  $\tau$  Umlauf  $\omega$  hinzugefügt, so ist  $\omega \cup \{\zeta_{\tau_n\tau}, \tau, \zeta_{\tau\delta}\}$  energetisch möglich.

Nachdem alle Fahrgastfahrten einem Umlauf zugeordnet worden sind, wird jedem Umlauf der Ladevorgang  $\zeta_{\tau_n\delta}$  nach der letzten Fahrgastfahrt hinzugefügt. Dann ist jeder Umlauf energetisch möglich und es überlappen sich keine zwei Fahrgastfahrten eines Umlaufs. Damit ist jeder Umlauf per Definition 3.4 möglich und jede Fahrgastfahrt in genau einem Umlauf enthalten. Also ist der Umlaufplan  $\Omega$  per Definition 3.5 eine gültige Lösung.

**Lemma 6.2.** Existiert eine gültige Lösung für das EVSP, dann liefert Algorithmus 1 eine gültige Lösung.

Beweis. Existiert eine gültige Lösung, so ist jede Fahrgastfahrt  $\tau$  in genau einem möglichen Umlauf  $\omega$  enthalten. Dann ist ein Umlauf  $\omega' = \{\zeta_{\delta\tau}, \tau, \zeta_{\tau\delta}\}$ , der  $\tau$  enthält, auch möglich. Denn der Ladezustand zu Beginn von  $\tau$  nach dem bestmöglichen Ladevorgang  $\zeta_{\delta\tau}$  ist mindestens gleich dem Ladezustand vor  $\tau$  in  $\omega$ , da jede Ladestation vom Depot aus erreichbar ist. Demnach fügt Algorithmus 1 jede Fahrgastfahrt immer einem möglichen Umlauf hinzu und gibt damit eine gültige Lösung.

Schließlich wollen wir die Laufzeit des Algorithmus bestimmen. Diese ist offensichtlich abhängig von der Anzahl der Fahrgastfahrten  $|\mathcal{T}|$  und der Anzahl der Ladestationen  $|\mathcal{C}|$ .

**Lemma 6.3.** Algorithmus 1 liefert eine Lösung für das EVSP in  $\mathcal{O}(|\mathcal{T}|^2|\mathcal{C}|)$  Zeit.

Beweis. Das Sortieren der Fahrten in Zeile 1 geschieht in  $\mathcal{O}(|\mathcal{T}|\log|\mathcal{T}|)$  Zeit. Für jede Fahrgastfahrt wird  $\mathcal{O}(|\mathcal{T}||\mathcal{C}|)$  Zeit für die Schritte in den Zeilen 4-22 benötigt: Speichern wir die Ankunftszeit und den Ankunftsort der letzten Fahrgastfahrt  $\tau_n$  jedes Umlaufs, kann für einen Umlauf in konstanter Zeit überprüft werden, ob sich eine Fahrgastfahrt  $\tau$  mit  $\tau_n$  zeitlich überlappt. Speichern wir weiter für jeden Umlauf  $\omega$  den Ladezustand des Fahrzeugs, das  $\omega$  ausführt, am Ende von  $\tau_n$ , können wir in  $\mathcal{O}(|\mathcal{C}|)$  Zeit Ladevorgänge  $\zeta_{\tau_n\tau}$ 

Abbildung 6.1: Fahrgastfahrten aus Theorem 6.5 für n=2. Jedes Intervall ist eine Fahrgastfahrt. An den Enden des Intervalls steht der Abfahrts-und Ankunftsort.

und  $\zeta_{\tau\delta}$  finden und Ladezustand  $q_{\tau}^{\omega}$  berechnen. Dann lässt sich in konstanter Zeit prüfen, ob der Umlauf mit  $\zeta_{\tau_n\tau}$ ,  $\tau$  und  $\zeta_{\tau\delta}$  energetisch möglich ist. Es gibt höchstens so viele Umläufe wie Fahrgastfahrten. Also ist die gesamte Laufzeit  $\mathcal{O}(|\mathcal{T}|^2|\mathcal{C}|)$ .

Aus Lemma 6.1 – 6.3 folgt unmittelbar

**Theorem 6.4.** Die Heuristik in Algorithmus 1 liefert eine gültige Lösung für das EVSP in  $\mathcal{O}(|\mathcal{T}|^2|\mathcal{C}|)$  Zeit genau dann, wenn eine gültige Lösung für das EVSP existiert.

Zuletzt zeigen wir, dass die Lösung unserer Heuristik beliebig weit vom Optimum entfernt sein kann.

**Theorem 6.5.** Sei  $n \in \mathbb{N}$ , dann existiert eine Instanz des EVSP mit 4n Fahrgastfahrten, für die der Greedy-Algorithmus eine Lösung  $\Omega$  der Größe  $|\Omega| = 2n$  ausgibt und eine Lösung  $\Omega_{opt}$  der Größe  $|\Omega_{opt}| = 2$  existiert.

Beweis. Wir konstruieren eine Instanz mit zwei Orten A und B und 4n Fahrgastfahrten. Für jedes  $k \in [n]$  ist  $\tau_{4k-3} = (A, 8k, B, 8k + 2)$ ,  $\tau_{4k-2} = (A, 8k + 3, B, 8k + 5)$ ,  $\tau_{4k-1} = (B, 8k + 4, A, 8k + 6)$  und  $\tau_{4k} = (B, 8k + 7, A, 8k + 9)$ . Die Fahrten sind für n = 2 in Abbildung 6.1 dargestellt. Für alle Fahrgastfahrten wird Energie gleich 1 benötigt. Für eine Fahrt zwischen ein und demselben Ort wird keine Energie benötigt, allerdings wird für die Leerfahrt zwischen A und B und zwischen B und A Energie n - 1 benötigt. Das Depot  $\delta$  ist am Ort A, die Batteriekapazität  $\beta$  ist gleich 2n. Es gibt keine Möglichkeit zu laden; die Menge der Ladestationen  $\mathcal{C} = \emptyset$ .

Eine optimale Lösung  $\Omega_{\rm opt}$  besteht aus zwei Umläufen  $\omega_1$  und  $\omega_2$ . Umlauf  $\omega_1$  besteht aus den Fahrten mit ungeradem Index,  $\omega_2$  aus den Fahrten mit geradem Index. Die Umläufe sind möglich, da sich keine Fahrgastfahrten zeitlich überlappen, keine Energie für Leerfahrten benötigt wird und alle Fahrgastfahrten 2n, also gleich  $\beta$ , Energie kosten. Da sich zwei Fahrgastfahrten zeitlich überlappen, werden mindestens zwei Umläufe benötigt. Damit ist die Lösung optimal.

Da Algorithmus 1 die Fahrgastfahrten nach Abfahrtszeit sortiert zu Umläufen hinzufügt, liefert er die Lösung  $\Omega = \{\{\tau_{2i-1}, \tau_{2i}\} \mid i \in [2n]\}$  mit  $2n = n|\Omega_{\text{opt}}|$  Umläufen.  $\square$ 

Freilich ist die konstruierte Instanz künstlich, da in der Realität die benötigte Energie für eine Fahrt zwischen zwei Orten A und B, nicht um ein Vielfaches größer ist als die Energie, die für eine Fahrgastfahrt zwischen A und B benötigt wird.

# 7 Experimente

Wir bewerten die Leistung unserer Algorithmen auf Instanzen des öffentlichen Personennahverkehrs in Berlin. Wir lösen die Instanzen mit dem ILP mit und ohne Datenvorverarbeitung sowie mit dem Greedy-Algorithmus.

**Datensätze.** Wir nutzen einen Datensatz extrahiert aus dem öffentlichen Personennahverkehr in Berlin mit 39 Linien und 4748 Fahrgastfahrten. Kleinere Datensätze wurden durch das Löschen von Fahrten in regelmäßigen Abständen erzeugt. Tabelle 7.1 zeigt alle Datensätze.

In Anlehnung an Jefferies und Göhlich [6], welche einen Überblick über den Stand der Technik elektrischer Busse mit Batterie geben, nehmen wir eine Batteriekapazität  $\beta$  von 200kWh und eine Ladeleistung  $p_C$  gleich 300kW an. Die Energie Q(x,y), die ein Fahrzeug für eine Fahrt zwischen zwei Endhaltestellen x und y verbraucht, berechnen wir aus der Distanz D(x,y) zwischen den Orten und einem durchschnittlichem Verbrauch  $\overline{q}$  von 1.5kWh/km, sprich  $Q(x,y) = \overline{q} \cdot D(x,y)$ . Die Fahrzeit T(x,y) zwischen zwei Endhaltestellen x und y berechnen wir auch aus der Distanz D(x,y) und einer Durchschnittsgeschwindigkeit  $\overline{v}$  von 20km/h, sprich  $T(x,y) = D(x,y)/\overline{v}$ . Wir nehmen an, dass für eine Fahrgastfahrt  $\tau_i$  genauso viel Energie benötigt wird, wie für eine Leerfahrt zwischen dem Abfahrtsort  $d_i$  und Ankunftsort  $a_i$  von  $\tau_i$ , sprich  $Q^P(\tau_i) = Q(d_i, a_i)$ .

Die Distanz zwischen dem Abfahrts-und Ankunftsort jeder Fahrgastfahrt ist gegeben. Alle übrigen Distanzen zwischen den Endhaltestellen und dem Depot wurden mit Hilfe

Tabelle 7.1: Datensätze.

Datensatz	# Fahrgastfahrten	# Ladestationen	# Endhaltestellen
1	95	10	52
2	238	10	69
3	475	10	78
4	792	10	77
5	1,187	10	84
6	1,583	10	84
7	2,035	10	84
8	2,374	10	90
9	2,775	10	86
10	3,165	10	90
11	3,836	10	91
12	4,748	10	93

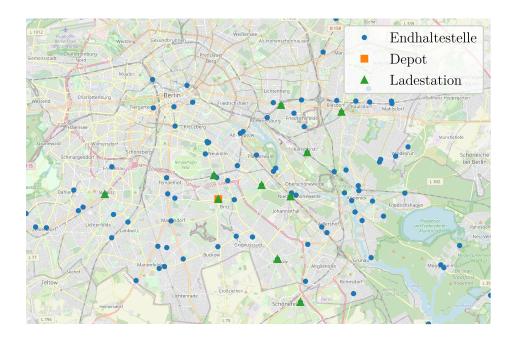


Abbildung 7.1: Verteilung der Endhaltestellen, des Depots und der Ladestationen. Die Karte ist von OpenStreetMap<sup>1</sup>.

von OpenStreetMap¹ berechnet. Es wurden neun Ladestationen an den Endhaltestellen mit den meisten Abfahrten und Ankünften aller Fahrgastfahrten und eine Ladestation am Depot platziert. In Abbildung 7.1 sind alle Endhaltestellen, Ladestationen und das Depot des Datensatzes zu sehen.

**Ergebnisse.** Die Experimente wurden auf einer Maschine mit einer 4-Kern Intel Xeon W-2125 CPU getaktet mit 4GHz und 256 GB DDR4-RAM auf Linux Kernel 4.15 ausgeführt. Das ILP wurde mit Gurobi Version 8.1.1 gelöst. Die Rechenzeit wurde auf 30h begrenzt. Der Greedy-Algorithmus wurde in Python geschrieben und mit Python 2.7 ausgeführt.

Tabelle 7.2 zeigt die Ergebnisse unserer Experimente. Obwohl die Entscheidungsvariante des EVSP wie in Kapitel 4 gezeigt  $\mathcal{NP}$ -vollständig ist, fand Gurobi optimale Lösungen. Nach Anwendung der Reduktionsregel konnten Instanzen mit bis zu 3836 Fahrgastfahrten optimal gelöst werden. Ohne Datenreduktion besaß die größte optimal lösbare Instanz 2374 Fahrten. Mit der Reduktionsregel konnten also etwa 60% größere Instanzen gelöst werden. Dabei war die Laufzeit bis zu 18.5 Mal schneller. In Tabelle 7.3 sind die Laufzeiten zum Aufstellen der Variablen und Bedingungen sowie zum Lösen des ILP mit und ohne Datenreduktion angegeben. Durch die Reduktionsregel konnte die Zeit zum Aufstellen der Bedingungen des ILP bereits so verkürzt werden, dass sich der Zeitaufwand für die Reduktion der Anzahl an Variablen des ILP lohnte. Die Größe der aufgestellten Programme mit und ohne Reduktionsregel ist in Tabelle 7.4 angegeben.

<sup>&</sup>lt;sup>1</sup>https://www.openstreetmap.org

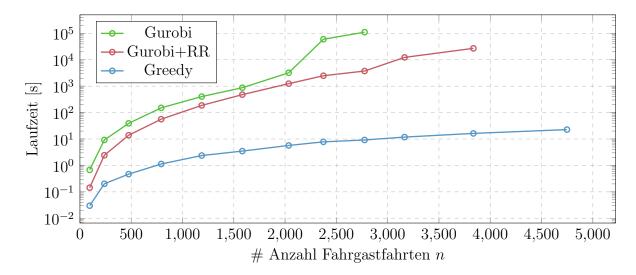


Abbildung 7.2: Laufzeiten der Lösungsansätze in Abhängigkeit der Anzahl der Fahrgastfahrten.

Die Anzahl der Variablen nach Anwendung der Regel ist im Durchschnitt 2.4 Mal kleiner, die Anzahl der Bedingungen 2.6 Mal. Instanzen mit mehr als 2775 Fahrten konnten ohne Datenvorverarbeitung und Instanzen mit mehr als 3836 Fahrten auch danach nicht von Gurobi gelöst werden, da zu wenig Arbeitsspeicher verfügbar war.

Der Greedy-Algorithmus liefert in wenigen Sekunden eine Lösung. Das ist eine andere

Tabelle 7.2: Die Ergebnisse der Experimente. Wir präsentieren für jeden Datensatz (DS) die Größe der Lösung (Sol), also die minimale Anzahl der benötigten Fahrzeuge, die Laufzeit t und die Lücke zwischen der von Gurobi gefundenen und der berechneten optimalen Lösung (Gap), beziehungsweise die Güte der Lösung unserer drei Algorithmen.

Gurobi			Gurobi+RR			Greedy			
DS	Sol	t [h:m:s]	$\mathrm{Gap}~[\%]$	Sol	t [h:m:s]	$\mathrm{Gap}~[\%]$	Sol	t[s]	Güte [%]
1	8	4.0	0	8	2.8	0	8	0	0
2	19	30.0	0	19	20.1	0	20	0.2	5.3
3	33	2:04	0	33	1:21	0	33	0.5	0
4	48	6:26	0	48	4:09	0	51	1.1	6.3
5	70	15:45	0	70	10:21	0	72	2.4	2.9
6	90	30:45	0	90	20:45	0	92	3.5	2.2
7	109	1:20:20	0	109	42:14	0	114	5.7	4.6
8	123	17:14:14	0	123	1:10:45	0	127	7.8	3.3
9	139	31:34:17	0.7	138	1:42:36	0	145	9.2	5.1
10	-	-	-	156	4:22:18	0	160	11.8	2.6
11	-	-	-	179	8:55:35	0	185	16.3	3.4
12	-	-	-	-	-	-	215	22.7	-

#### 7 Experimente

Größenordnung im Vergleich zur Zeit, die zum Lösen des ILP benötigt wird. Während die Laufzeit zum Lösen des ILP exponentiell ist, liegt die durchschnittliche Laufzeit des Greedy-Algorithmus weit unter der Worst-Case-Laufzeit  $\mathcal{O}(n^2m)$  für n Fahrgastfahrten und m Ladestationen. In Abbildung 7.2 sind die Laufzeiten der Algorithmen dargestellt. Für unsere Instanzen lieferte der Greedy-Algorithmus teils optimale Lösungen, im Durchschnitt waren sie 3.2% und maximal 6.3% größer ist als die optimale Lösung. Nur mit der Heuristik konnte eine Lösung für die größte Probleminstanz mit knapp 5000 Fahrten gefunden werden.

Vergleich. Die in den Arbeiten [10, 12, 13] entwickelten Lösungsansätze wurden auch auf Instanzen des öffentlichen Personennahverkehrs getestet. Diese basieren jedoch auf anderen Modellen und wurden auf Maschinen mit weniger Arbeitsspeicher ausgeführt. Reuer, Kliewer und Wolbeck [10] lösten mit einer Heuristik eine Instanz mit zirka 10000 Fahrgastfahrten. Die Güte der Lösung ist jedoch nicht klar. Van Kooten Niekerk, van den Akker und Hoogeveen [12] lösten eine Instanz mit etwa 500 Fahrgastfahrten optimal. Wen u. a. [13] konnten eine Instanz mit 30 Fahrten optimal lösen, Instanzen mit 500 Fahrgastfahrten lösten sie mit einer Metaheuristik. Für das Reproduzieren der Algorithmen dieser Arbeiten war im Rahmen dieser Arbeit nicht genug Zeit vorhanden.

Tabelle 7.3: Die Laufzeiten zum Aufstellen und Lösen der ILPs. Wir zeigen die Zeit  $t_{\text{Vars}}$  um die Variablen des ILP zu berechnen, die Zeit  $t_{\text{Constrs}}$  um die Bedingungen aufzustellen, die Zeit  $t_{\text{Solve}}$  um das ILP zu lösen mit und ohne Anwendung der Reduktionsregel (RR) und die Zeit  $t_{\text{RR}}$  für die Anwendung der Regel für jeden Datensatz (DS).

Gurobi				Gurobi+RR			
DS	$t_{\mathrm{Vars}}$ [s]	$t_{\rm Constrs}$ [s]	$t_{\rm Solve}$ [s]	$t_{\rm RR}~[{ m s}]$	$t_{ m Vars}$ [s]	$t_{\rm Constrs}$ [s]	$t_{\rm Solve}$ [s]
1	$4.51 \cdot 10^{-1}$	$2.86 \cdot 10^{0}$	$6.78 \cdot 10^{-1}$	$1.16 \cdot 10^{0}$	$1.61 \cdot 10^{0}$	$1.04 \cdot 10^{0}$	$1.44 \cdot 10^{-1}$
2	$2.63 \cdot 10^{0}$	$1.81 \cdot 10^1$	$9.21 \cdot 10^{0}$	$7.64 \cdot 10^{0}$	$1.03 \cdot 10^{1}$	$7.37 \cdot 10^{0}$	$2.43 \cdot 10^{0}$
3	$1.01 \cdot 10^1$	$7.47\cdot 10^1$	$3.92 \cdot 10^{1}$	$2.92\cdot 10^1$	$3.93 \cdot 10^{1}$	$2.81\cdot 10^1$	$1.39 \cdot 10^{1}$
4	$2.79 \cdot 10^{1}$	$2.08 \cdot 10^{2}$	$1.51 \cdot 10^2$	$8.29 \cdot 10^{1}$	$1.11 \cdot 10^{2}$	$8.3 \cdot 10^{1}$	$5.62\cdot 10^1$
5	$6.36 \cdot 10^{1}$	$4.78 \cdot 10^{2}$	$4.04 \cdot 10^{2}$	$1.89 \cdot 10^{2}$	$2.53 \cdot 10^{2}$	$1.82 \cdot 10^{2}$	$1.87 \cdot 10^2$
6	$1.12 \cdot 10^{2}$	$8.6 \cdot 10^{2}$	$8.74 \cdot 10^{2}$	$3.3\cdot 10^2$	$4.42\cdot 10^2$	$3.26 \cdot 10^{2}$	$4.78 \cdot 10^{2}$
7	$1.86 \cdot 10^{2}$	$1.43 \cdot 10^{3}$	$3.21 \cdot 10^{3}$	$5.62\cdot 10^2$	$7.48 \cdot 10^{2}$	$5.43 \cdot 10^{2}$	$1.24 \cdot 10^{3}$
8	$2.56 \cdot 10^{2}$	$2.03 \cdot 10^{3}$	$5.98 \cdot 10^4$	$7.54\cdot 10^2$	$1.01 \cdot 10^{3}$	$7.52\cdot 10^2$	$2.48 \cdot 10^{3}$
9	$4.05 \cdot 10^{2}$	$2.86 \cdot 10^{3}$	$1.1 \cdot 10^{5}$	$9.64 \cdot 10^2$	$1.37 \cdot 10^3$	$1.05 \cdot 10^3$	$3.74 \cdot 10^{3}$
10	-	-	-	-	$1.86 \cdot 10^{3}$	$1.6 \cdot 10^{3}$	$1.23 \cdot 10^4$
11	-	-	-	-	$2.78 \cdot 10^3$	$2.36 \cdot 10^3$	$2.7 \cdot 10^4$

Tabelle 7.4: Die Größe der aufgestellten ILPs. Wir zeigen die Anzahl der Variablen und Anzahl der Bedingungen mit und ohne Anwendung der Reduktionsregel (RR) für jeden Datensatz (DS).

	, ,	` '			
	G	urobi	Gurobi+RR		
DS	# Variablen	# Bedingungen	# Variablen	# Bedingungen	
1	$4.26\cdot 10^4$	$1.19\cdot 10^5$	$1.75 \cdot 10^4$	$4.41\cdot 10^4$	
2	$2.64 \cdot 10^{5}$	$7.41 \cdot 10^5$	$1.15\cdot 10^5$	$2.94 \cdot 10^{5}$	
3	$1.04 \cdot 10^{6}$	$2.92 \cdot 10^{6}$	$4.44 \cdot 10^{5}$	$1.13 \cdot 10^{6}$	
4	$2.89 \cdot 10^{6}$	$8.12 \cdot 10^{6}$	$1.24 \cdot 10^{6}$	$3.17 \cdot 10^{6}$	
5	$6.53 \cdot 10^{6}$	$1.83 \cdot 10^{7}$	$2.78 \cdot 10^{6}$	$7.09 \cdot 10^{6}$	
6	$1.15 \cdot 10^{7}$	$3.24 \cdot 10^{7}$	$4.93 \cdot 10^{6}$	$1.26 \cdot 10^{7}$	
7	$1.91 \cdot 10^{7}$	$5.35 \cdot 10^{7}$	$8.15 \cdot 10^{6}$	$2.08 \cdot 10^{7}$	
8	$2.61 \cdot 10^{7}$	$7.33 \cdot 10^{7}$	$1.11 \cdot 10^{7}$	$2.84 \cdot 10^{7}$	
9	$3.56 \cdot 10^{7}$	$1 \cdot 10^{8}$	$1.51 \cdot 10^7$	$3.83 \cdot 10^{7}$	
10	-	-	$1.97 \cdot 10^7$	$5.01 \cdot 10^7$	
11	-	-	$2.88\cdot 10^7$	$7.34 \cdot 10^{7}$	

### 8 Fazit

Wir führten ein Modell für die Umlaufplanung von batteriebetriebenen Bussen ein, fanden neue Erkenntnisse zur Komplexität des Problems und präsentierten zwei neue Lösungsansätze mit dem Ziel, die minimale Anzahl benötigter Busse zu finden. Eine Neuheit des Modells ist, dass das teilweise (nicht vollständige) Laden der Batterie an beliebigen Ladestationen ermöglicht worden ist. Das aufgestellte Integer Linear Program des  $\mathcal{NP}$ -schweren LEVSP konnte in Kombination mit unserer Datenreduktionsregel für Instanzen mit knapp 4000 Fahrgastfahrten optimal gelöst werden. Damit ist es der erste Lösungsansatz, der die wichtigsten Punkte beim Laden batteriebetriebener Busse berücksichtigt und eine optimale Lösung für praktisch relevante Instanzen findet. Der eingeführte Greedy-Algorithmus berechnet für Fahrplandaten in Berlin in Sekunden teils optimale Lösungen und ist höchstens 6% schlechter als das Optimum.

Mit den in dieser Arbeit entwickelten Lösungsmethoden wäre es nun interessant, den Bedarf elektrischer Busse in Abhängigkeit von Variablen, die aus praktischer Sicht relevant sind, wie die Batteriekapazität, die Anzahl und Verteilung der Ladestationen und die Ladeleistung zu untersuchen.

Der Greedy-Algorithmus kann mit einer nicht linearen Ladefunktion rechnen und ließe sich einfach so erweitern, dass ein Bus an mehreren Ladestationen hintereinander laden kann. Sollte das wider Erwarten Auswirkungen auf den Fahrzeugbedarf haben, wäre es interessant, den exakten Lösungsansatz so zu erweitern, dass diese Faktoren berücksichtigt werden. Es scheint vielversprechend zu versuchen, die schnelle Lösung des Greedy-Algorithmus mit Metaheuristiken, wie z.B. lokaler Suche, zu verbessern. Außerdem könnte es sich lohnen zu untersuchen, welchen Einfluss kleine Änderungen am Fahrplan auf den Fahrzeugbedarf haben.

Unser Modell bildet zwar die wichtigsten Punkte bei der Umlaufplanung von batteriebetriebenen Bussen ab, kann aber um einige Facetten erweitert werden. In der Praxis können an einer Ladestation nur begrenzt viele Bussen gleichzeitig laden. Weiter nehmen wir eine homogene Fahrzeugflotte an. Eine direkte Erweiterung des Modells in dieser Arbeit sind mehrere Fahrzeugtypen und Depots wie beim klassischen Problem der Umlaufplanung.

### **Glossar**

### Wir nennen

```
\mathcal{X}
                  Orte;
\delta
                  Depot, \delta \in \mathcal{X};
                 Fahrgastfahrt, die an Abfahrtsort d_i zur Zeit s_i beginnt und am Ankunftsort
\tau_i
                 a_i zur Zeit e_i endet, \tau_i \in \mathcal{X}^2 \times \mathbb{N}^2;
                 Fahrgastfahrten, \mathcal{T} \subset \mathcal{X}^2 \times \mathbb{N}^2;
\mathcal{T}
                 die für eine Fahrt zwischen Ort x und y benötigte Zeit, T: \mathcal{X}^2 \to \mathbb{N};
T(x,y)
\psi
                  den Umlauf eines Fahrzeugs, \psi \subseteq \mathcal{T};
Ψ
                  einen Umlaufplan, \Psi \subseteq \mathcal{P}(\mathcal{T});
β
                  Batteriekapazität eines Fahrzeugs, \beta \in \mathbb{N};
\mathcal{C}
                  Ladestationen, C \subseteq \mathcal{X};
                 die für eine Fahrgastfahrt \tau benötigte Energie, Q^P: \mathcal{T} \to \mathbb{N};
Q^P(\tau)
                 die für eine Fahrt zwischen Ort x und y benötigte Energie, Q: \mathcal{X}^2 \to \mathbb{N};
Q(x,y)
Q^C(q, \triangle t) Ladefunktion, Q^C: [0, \beta] \times \mathbb{N} \to [0, \beta];
                 Ladevorgang vor Fahrgastfahrt \tau_i an Ladestation c, \zeta_{\delta i}^c \in \delta \times \mathcal{T} \times \mathcal{C};
\zeta_{\delta i}^c
                 Ladevorgang zwischen Fahrgastfahrt \tau_i und \tau_j an Ladestation c, \zeta_{ij}^c \in \mathcal{T}^2 \times \mathcal{C};
\zeta_{ij}^c
\zeta^c_{i\delta}
                 Ladevorgang nach Fahrgastfahrt \tau_i an Ladestation c, \zeta_{i\delta}^c \in \mathcal{T} \times \delta \times \mathcal{C};
Z
                  Ladevorgänge;
                  den Umlauf eines elektrischen Fahrzeugs, \omega \subseteq \mathcal{T} \cup Z;
\omega
                  einen Umlaufplan elektrischer Fahrzeuge, \Omega \subseteq \mathcal{P}(\mathcal{T} \cup Z);
\Omega
                  die Ladestation oder das Depot, von dem am wenigsten Energie zu d_i benötigt
\gamma_i
                  wird, \gamma_i \in \mathcal{C} \cup \{\delta\};
                 die Ladestation oder das Depot, zu der von a_i am wenigsten Energie benötigt
\epsilon_i
                  wird, \epsilon_i \in \mathcal{C} \cup \{\delta\}.
```

### Literatur

- [1] Jonathan D. Adler. "Routing and Scheduling of Electric and Alternative-Fuel Vehicles". Diss. Arizona State University, Mai 2014 (siehe S. 5, 6).
- [2] Buchmann. Basic to advanced battery information from battery university. zuletzt geprüft am 02.12.2020. 2014. URL: https://batteryuniversity.com (siehe S. 18).
- [3] Stefan Bunte und Natalia Kliewer. "An overview on vehicle scheduling models". In: Public Transport 1.4 (2009), S. 299–317. DOI: 10.1007/s12469-010-0018-5 (siehe S. 5).
- [4] Gupta, Lee und Leung. "An Optimal Solution for the Channel-Assignment Problem". In: *IEEE Transactions on Computers* C-28.11 (1979), S. 807–810 (siehe S. 23).
- [5] Klaus Jansen, Stefan Kratsch, Dániel Marx und Ildikó Schlotter. "Bin packing with fixed number of bins revisited". In: *Journal of Computer and System Sciences* 79.1 (2013), S. 39 –49. DOI: 10.1016/j.jcss.2012.04.004 (siehe S. 14).
- [6] Dominic Jefferies und Dietmar Göhlich. "A Comprehensive TCO Evaluation Method for Electric Bus Systems Based on Discrete-Event Simulation Including Bus Scheduling and Charging Infrastructure Optimisation". In: World Electric Vehicle Journal 11 (Aug. 2020), S. 56. DOI: 10.3390/wevj11030056 (siehe S. 5, 27).
- [7] Antoon W.J. Kolen, Jan Karel Lenstra, Christos H. Papadimitriou und Frits C.R. Spieksma. "Interval scheduling: A survey". In: *Naval Research Logistics (NRL)* 54.5 (2007), S. 530–543. DOI: 10.1002/nav.20231 (siehe S. 23).
- [8] Stefan Köller. Zwölf Großstädte wollen ab 2025 nur noch E-Busse kaufen. zuletzt geprüft am 08.10.2020. 2017. URL: https://www.electrive.net/2017/10/23/zwoelf-grossstaedte-wollen-ab-2025-nur-noch-e-busse-kaufen/ (siehe S. 5).
- [9] T. Paul und H. Yamada. "Operation and charging scheduling of electric buses in a city bus route network". In: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC). 2014, S. 2780–2786. DOI: 10.1109/ITSC.2014. 6958135 (siehe S. 5).
- [10] Josephine Reuer, Natalia Kliewer und Lena Wolbeck. "The Electric Vehicle Scheduling Problem A study on time-space network based and heuristic solution approaches". In: Juli 2015 (siehe S. 5, 6, 30).

#### Literatur

- [11] Ons Sassi und Ammar Oulamara. "Electric vehicle scheduling and optimal charging problem: complexity, exact and heuristic approaches". In: *International Journal of Production Research* 55.2 (2017), S. 519–535. DOI: 10.1080/00207543.2016. 1192695 (siehe S. 5, 14).
- [12] M. E. Van Kooten Niekerk, J. M. van den Akker und J. A. Hoogeveen. "Scheduling electric vehicles". In: *Public Transport* 9.1 (2017), S. 155–176. DOI: 10.1007/s12469-017-0164-0 (siehe S. 5, 6, 30).
- [13] M. Wen, E. Linde, S. Ropke, P. Mirchandani und A. Larsen. "An adaptive large neighborhood search heuristic for the Electric Vehicle Scheduling Problem". In: Computers and Operations Research 76 (2016), S. 73 –83. DOI: 10.1016/j.cor. 2016.06.013 (siehe S. 5, 30).
- [14] ZeEUS Project. ZeEUS eBus Report #2: An updated overview of electric buses in Europe. zuletzt geprüft am 30.11.2020. 2017. URL: http://zeeus.eu/uploads/publications/documents/zeeus-report2017-2018-final.pdf (siehe S. 5).
- [15] Guochuan Zhang, Xiaoqiang Cai und C.K. Wong. "Linear time-approximation algorithms for bin packing". In: *Operations Research Letters* 26.5 (2000), S. 217 –222. DOI: 10.1016/S0167-6377(99)00077-2 (siehe S. 15).