Optimal Integration of Autonomous Vehicles in Car Sharing

Development of a Heuristic considering Multimodal Transport and Integration in an Optimal Framework

Martin Sperr

November 8, 2016

Contents

Chapter 1

Introduction

FiXme Note: Check customer and costumer
FiXme Note: Check linebreak for in-text formulas
FiXme Note: Lagrange Heuristic for customer-dependent splitting as outlook
FiXme Note: Remove KK16 in bibliography

In this thesis, a heuristic for the routing of autonomous vehicles is refined. It is based on two master's theses [Kai16] and [Kno16] which were created earlier on the same faculty. While [Kno16] develops heuristic solution methods for a simplified model, [Kai16] examines a decomposition approach for solving the original model exactly. In contrast to the previous theses, an extended model is regarded here, where the possibility of multimodal transport is considered. First, the previously developed heuristic is extended to the possibility of multimodal transport. Then, this heuristic is integrated in the already existing optimal approach.

In commercial car sharing, a customer rents a car for a limited period of time. In the classic version, the customer gets the car on a fixed location and returns it to this location after usage. In contrast to this, free-floating car sharing allows the customer to pick any available car and to put it down somewhere in the operation area. The customer usually books the car beforehand, typically via a smartphone application. He pays a certain amount per minute of car usage. This method is obviously more customer-friendly since the customer has no effort in getting to and from the renting location. But this means significantly more effort for the car sharing supplier. He has to provide a comprehensive offer of available cars, such that there is always a car where the customer needs it. Further he is responsible for refueling and servicing the cars, wherever they are. customers may park their car where it suits them and simultaneously only rent a car if it is within a small walking distance to their current position. Therefore, the distribution of the cars heavily depends on the customer behavior. This might lead to an imbalance of supply and demand.

A possible solution for this is the usage of autonomous vehicles. Although they are not available on present day, this topic is highly researched. Autonomous vehicles may be available within the next ten to twenty years (cf. [Hau15]). The obvious advantage of autonomous cars is that they do not need a driver. An autonomous car is able change to the position after satisfying a customer on its own. The car can drive to a refuel station or to a position where it is needed next. For the customer, this behavior is similar to calling a taxi. The car picks him up on his present location and takes him to his destination. The supplier profits since he does not need employees for refueling or relocating the cars.

Besides cars, it is often advantageous for the customer to use public transport. For suitable trips, the usage of public transport is often faster and particularly cheaper. It is further more efficient in a city with many cars. But with public transport, there also some bad facts arise. The next station may be too far away for walking or unfavorable changing times increase the total travel time. In these cases it is often a good idea to combine car sharing and public transport in one trip. The customer uses the car for driving to a station with a good connection and continues the trip by using public transport. The combination of different types of transport is called “multiple-leg”. While in the previous master’s theses the number of legs was restricted to one, the here developed solution methods can cope with multiple legs.

The introduction of autonomous cars is a huge change in the service provided to the customer. Therefore, it is hard to predict the changing of the customer behavior. This estimation is not part of this thesis, the focus lies rather in the potential of autonomous cars. Therefore, actual renting data from present day are used. In order to model realistic alternatives for the customer the data are slightly modified, for example by splitting existing trips into a car trip and a public transport trip. The goal is to find an optimal fleet size for the car sharing provider. For finding the optimal fleet size, the number of needed cars has to be traded off against the total driven distance. And these costs have to be considered while providing a good service to the customer. The determined fleet size can be compared to the actual fleet size in order to estimate the cost saving with autonomous cars.

The setting for the underlying theses was the following: Each customer has a set of alternative trips, where one trip each has to be fulfilled in the solution. Each trip is a single-leg trip, i. e. the trip can either be a car or a public transport trip. [Kai16] provides an approach to solve this problem exactly via column generation. [Kno16] provides a fast heuristic via a time-dependent splitting of the trips. This heuristic requires the further restriction that each customer has only one alternative, i. e. each trip has to be fulfilled. In contrast, this thesis provides solution methods without both of these restrictions. The heuristic developed in [Kno16] is extended in order to cope with multiple-leg multimodal transport. The goal of this heuristic is to determine a good solution fast. Afterwards, the extended heuristic is integrated in the optimal framework developed in [Kai16], in order to find an optimal solution for the problem.

Chapter 2

Problem Description and Classification

In this chapter, the problem is stated in detail and the notation is introduced. The problem is classified by relating it to known problems in literature and its complexity is determined. Finally, known approaches to similar problems are regarded. Most of the following considerations are already part of the underlying theses [KK16], except for the fact that multiple leg is allowed. For better clarity to the reader, all crucial results are repeated here.

2.1 Situation and Issue

We regard the situation of free-floating car sharing as it exists today in combination with autonomous vehicles. Free-floating car sharing means that a customer can rent an available car wherever and whenever there is one and use it as long as he needs it. After usage, he parks the car somewhere in the operation area. We assume the existence of autonomous vehicles. An autonomous vehicle behaves the same as if a human being drives it, but without that a human being is necessarily present. Instead of looking for a car, a customer books a car via a smartphone application and gets picked up by the car at the desired start location and at the desired start time. For the customer, this seems similar to a taxi service.

The car sharing issue is combined with public transport as it is known today. There is a fixed schedule, according to which the bus or train visits public transport stations in a row at certain time points. A possible route for a customer may look as follows: The customer is picked up at his start position by a car and is brought to a station where he gets on a train. After finishing the train trip, he is picked up again by another car and is brought to his destination. It is also possible to change trains during this public transport trip. This behavior is very advantageous for the customer. While an in-between train trip is cheaper than a pure car trip, the combination of car and train is faster than a pure public transport trip since does not have walking and changing time.

2.2 Problem Description and Notation

In order to realize the previously described problem, we introduce a formal notation for the problem. We have a set of customers with known travel requests. Each of those can be realized by a number of precomputed multimodal routes. Each route consists of a sequence of trips. A trip is either a car trip or a public transport trip and has a fixed start and end position, as well as a fixed start and end time. Fulfilling a route means that the customer takes all the trips of this route in a row, i. e. he starts at the start point of the first trip and is finished at the end point of the last trip. The transition between two subsequent trips is the changing from a car to a train or the other way round. Each customer has to be satisfied, i. e. he has to be able to fulfill one of his alternative routes.
Although we formally define public transport trips here, we regard only the car trips in detail. Public transport is maintained by an own schedule and is not of interest for this issue. The availability of suitable public transport trips has to be respected while creating the routes (Chapter 6). There, the car trips are created in such a way that they are suitable to the existing public transport trips. Here, we assume the existence of feasible routes and therefore only control the behavior of the cars.
For fulfilling the car trips, we have a set of vehicles. For each vehicle, we have a position where it starts and a time from when it is available. A vehicle can drive from its start point to a trip's start point, execute this trip, and then drive from the trip's end point to the next trip's start point. After fulfilling its last trip, the car stays at the end point of the last trip. The sequence, in which the vehicle executes the trips, is called the duty of the vehicle.
A further restriction to the problem is that the vehicles have a maximal range that they can drive without refueling. In the model, refueling points are included.
The objective is to create a schedule for the vehicles, such that for each customer exactly one route is fulfilled, i. e. every trip of this route is fulfilled. Further, each vehicle has a feasible fuel state all the time and visits a refuel station when necessary.

Customers, Trips and Vehicles

We are given a set of car trips $\mathcal{T}$. Each trip $t \in \mathcal{T}$ has a start and end location $p_t^{\text{start}}, p_t^{\text{end}}$ and a start and end time $z_t^{\text{start}}, z_t^{\text{end}}$. For the sake of completeness, we define the set of public transport trips $\mathcal{T}_{\text{public}}$ with the same properties for $t \in \mathcal{T}_{\text{public}}$.
The set of vehicles is denoted by $\mathcal{V}$. The start position and the start time of a vehicle $v \in \mathcal{V}$ is p_v and z_v.
We have the set of customers $\mathcal{C}$ and the set of multimodal routes $\mathcal{M}$. A route $m = (t_1, \dots, t_{k_m})$ is a finite sequence of trips with the following properties:

$$p_{t_i}^{\text{end}} = p_{t_{i+1}}^{\text{start}} \qquad z_{t_i}^{\text{end}} \leq z_{t_{i+1}}^{\text{start}} \qquad \text{for all } i \in [k-1].$$

We define the route start and end locations and times for $m \in \mathcal{M}$ as

$$p_m^{\text{start}} := p_{t_1}^{\text{start}} \qquad p_m^{\text{end}} := p_{t_k}^{\text{end}} \qquad z_m^{\text{start}} := z_{t_1}^{\text{start}} \qquad z_m^{\text{end}} := z_{t_k}^{\text{end}}.$$

The mapping $M : \mathcal{T} \to \mathcal{M}$ indicates to which route a trip belongs. Each customer $c \in \mathcal{C}$ has a finite set of alternative routes. The mapping $C : \mathcal{M} \to \mathcal{C}$ shows which route belongs to which customer.
We use the notation

$$\begin{aligned} C^{-1}(c) &:= C^{-1}(\{c\}) = \{m \in \mathcal{M} \mid C(m) = c\} && \text{for } c \in \mathcal{C} \\ M^{-1}(m) &:= M^{-1}(\{m\}) = \{t \in \mathcal{T} \mid M(t) = m\} && \text{for } m \in \mathcal{M} \\ (M \circ C)^{-1}(c) &:= M^{-1}\left(C^{-1}(c)\right) = \{t \in \mathcal{T} \mid C(M(t)) = c\} && \text{for } c \in \mathcal{C} \end{aligned}$$

for all routes of a customer, all trips of a route and all trips of a customer, respectively. For each route of the same customer $m \in C^{-1}(c)$, the start and end positions are the same, the start and end times may differ. We define the customer start and end times for $c \in \mathcal{C}$

$$z_c^{\text{start}} := \min_{m \in C^{-1}(c)} z_m^{\text{start}} \qquad z_c^{\text{end}} := \max_{m \in C^{-1}(c)} z_m^{\text{end}}.$$

Fuel and Refueling

We have to consider fuel restrictions. Fuel can be any form of energy the considered vehicle is powered with. For each vehicle, the fuel level is in the interval $[0, 1]$, where 1 means full capacity and 0 is empty. We call a trip without a customer, i. e. a trip between two trips, a deadhead trip. A car may visit a refuel station only during a deadhead trip. For simplicity of the model, each car is allowed to refuel at most once between two trips. On a refuel station, there are no capacity constraints, i. e. two or more vehicles may refuel at the same time on the same station.
For refueling, we have a set of refuel stations $\mathcal{R}$. A refuel station $r \in \mathcal{R}$ has a location p_r. We define $f_{s,t}^{\text{d}}$ for $s \in \mathcal{V} \cup \mathcal{T} \cup \mathcal{R}$, $t \in \mathcal{T} \cup \mathcal{R}$ as the amount, the fuel level decreases along a deadhead trip. f_t^{t} for $t \in \mathcal{T} \cup \mathcal{R}$ is the amount of fuel, a vehicle needs for a trip. For $r \in \mathcal{R}$ holds $f_r^{\text{t}} \leq 0$. f_v^0 for $v \in \mathcal{V}$ is the initial fuel state of a car.

Ordering of the Trips

We define the time, a vehicle needs to get from position p_1 to p_2 as t_{p_1,p_2}. We define

$$t_{s,t} = \begin{cases} t_{p_s^{\text{end}}, p_t^{\text{start}}} & \text{if } s, t \in \mathcal{T} \\ t_{p_s, p_t^{\text{start}}} & \text{if } s \in \mathcal{V} \cup \mathcal{R}, t \in \mathcal{T} \\ t_{p_s^{\text{end}}, p_t} & \text{if } s \in \mathcal{T}, t \in \mathcal{R} \\ t_{p_s, p_t} & \text{if } s \in \mathcal{V}, t \in \mathcal{R} \end{cases}$$

as the time a vehicle needs from one trip to another.
In order to decide whether a vehicle is able to fulfill two trips in a row, we define a partial ordering on the set of vehicles and trips. The set of public transport trips is left out in this definition.

Definition 1 (Order of trips)**.** The binary relation $\prec$ on $\mathcal{V} \mathbin{\dot{\cup}} \mathcal{T}$ is defined as follows:

$$\begin{aligned} s \prec t \quad &:\Leftrightarrow \quad \left(z_s^{\text{end}} + t_{s,t} \leq z_t^{\text{start}}\right) \wedge ((M \circ C)(s) \neq (M \circ C)(t) \vee M(s) = M(t)) \\ & \qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{for all } s \in \mathcal{V} \mathbin{\dot{\cup}} \mathcal{T}, t \in \mathcal{T} \\ s \not\prec t \quad & \qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{for all } s \in \mathcal{V} \mathbin{\dot{\cup}} \mathcal{T}, t \in \mathcal{V} \end{aligned}$$

The binary relation $\preceq$ on $\mathcal{V} \mathbin{\dot{\cup}} \mathcal{T}$ is defined as:

$$s \preceq t \qquad :\Leftrightarrow \qquad s = t \wedge s \prec t \qquad \text{for all } s, t \in \mathcal{V} \mathbin{\dot{\cup}} \mathcal{T}$$

The expression $s \prec t$ means, that one car is able to fulfill both trips, first s and then t. A car must not cover two trips of the same customer, except they belong to the same route. This results from the problem setting, that for each customer exactly one route is fulfilled.

Remark 1. Note, that $\preceq$ is not a partial order on $\mathcal{V} \mathbin{\dot{\cup}} \mathcal{T}$ since the transitivity is missing. Let $t_1, t_2, t_3 \in \mathcal{T}$ with

$$z_{t_1}^{\text{end}} + t_{t_1,t_2} \leq z_{t_2}^{\text{start}} \qquad\qquad z_{t_2}^{\text{end}} + t_{t_2,t_3} \leq z_{t_3}^{\text{start}}$$

$$(M \circ C)(t_1) = (M \circ C)(t_3) \qquad (M \circ C)(t_1) \neq (M \circ C)(t_2) \qquad M(t_1) \neq M(t_3)$$

Then,

$$t_1 \preceq t_2 \preceq t_3 \qquad\qquad t_1 \not\preceq t_3$$

Problem Description

We define the considered problem as follows: Find a schedule of trips for every vehicle including refueling stops and a sequence of trips for every customer. Therefore, the car trips are fulfilled by the scheduled car and the public transport trips by public transport according to its timetable. For this, we have the following conditions:

- Each car is able to serve its scheduled trips, considering time and location.
- The fuel state of each car is always in a feasible range.
- Each customer is able to complete his route, considering time and location.
- For each customer, exactly one route is chosen.

The goal is to find a cost-minimal feasible schedule considering all these constraints.

Costs

We introduce the following types of costs:

- Vehicles costs c^{v}: unit costs for each used vehicle
- Deadhead costs $c^{\mathrm{d}}_{s,t}$ for $s \in \mathcal{V} \cup \mathcal{T} \cup \mathcal{R}, t \in \mathcal{T} \cup \mathcal{R}$: costs, if a vehicle drives to a trip or a refuel station without a customer using it
- Trip costs c^{t}_t for $t \in \mathcal{T}$: costs for fulfilling a trip

For public transport, we define either trip costs for each public transport trip or fixed costs for each customer using public transport. Finally, we define costs to consider the customer preferences.

- Trip costs c^{t}_t for $t \in \mathcal{T}_{\text{public}}$: costs for using public transport
- Route-dependent costs $\bar{c}^{\mathrm{r}}_m$ for $m \in \mathcal{M}$: costs for customer preferences and unit costs for using public transport

Since the trip costs for public transport are connected with the choice of the route, we easily add these costs to the trip costs.

$$c^{\mathrm{r}}_m := \bar{c}^{\mathrm{r}}_m + \sum_{t \in m \cap \mathcal{T}_{\text{public}}} c^{\mathrm{t}}_t \qquad \text{for } m \in \mathcal{M}$$

The route costs additionally include customer preferences. Each customer has a set of alternative routes. He does not choose the route by himself, but this is decided by the problem whatever fits best into the system. The user preference costs work as penalty terms for an inconvenient route choice. This means, a route that is disadvantageous for the customer is penalized. Then, either the less favorable route is chosen (if it fits better into the system) and this is penalized. Or the more favorable route is chosen although it is not so good for the system. With this, a realistic customer behavior is modeled.
The customer preferences are e. g. the total travel time, the number of changes or the costs for the customer. Typically, a pure car trip is faster but more expensive. Further, a late departure time or an early arrival time can be criteria for this cost function.

Additional Assumptions

In the following, we summarize all the assumptions we made on the input data.
All costs are non-negative.

$$c^{\mathrm{v}} \geq 0 \qquad c^{\mathrm{d}}_{s,t} \geq 0 \qquad c^{\mathrm{t}}_t \geq 0 \qquad \bar{c}^{\mathrm{r}}_m \geq 0 \qquad \text{for all } s, t \in \mathcal{T}, m \in \mathcal{M}. \tag{2.1}$$

The fuel consumption is non-negative, except for refueling.

$$f_t^{\mathrm{t}} \geq 0 \qquad f_r^{\mathrm{t}} \leq 0 \qquad \text{for all } t \in \mathcal{T}, r \in \mathcal{R} \tag{2.2}$$

$$f_{s_1,s_2}^{\mathrm{d}} \geq 0 \qquad \text{for all } s_1 \in \mathcal{V} \cup \mathcal{T} \cup \mathcal{R}, s_2 \in \mathcal{T} \cup \mathcal{R} \tag{2.3}$$

There are no zero-time rentals.

$$z_t^{\mathrm{start}} < z_t^{\mathrm{end}} \qquad \text{for all } t \in \mathcal{T} \tag{2.4}$$

We assume the Triangle Inequalities for $s \in \mathcal{V} \cup \mathcal{T} \cup \mathcal{R}$ and $r, t \in \mathcal{T} \cup \mathcal{R}$:

$$t_{s,t} \leq t_{s,r} + t_{r,t} \qquad c_{s,t}^{\mathrm{d}} \leq c_{s,r}^{\mathrm{d}} + c_{r,t}^{\mathrm{d}} \qquad f_{s,t}^{\mathrm{d}} \leq f_{s,r}^{\mathrm{d}} + f_{r,t}^{\mathrm{d}} \tag{2.5}$$

From (2.1) and (2.5) we get:

$$c_{s,t}^{\mathrm{d}} \leq c_{s,r}^{\mathrm{d}} + c_r^{\mathrm{t}} + c_{r,t}^{\mathrm{d}} \qquad \text{for all } s, r, t \in \mathcal{T} \tag{2.6}$$

2.3 Classification

We want to classify our problem in relation to other known problems in literature and state the difficulty of these problems.

Vehicle Scheduling Problems

According to the structure of the problem stated in Section 2.2, we regard the field of vehicles scheduling problems (VSP). [BK09] define the (VSP) as follows: "Given a set of timetabled trips with fixed travel (departure and arrival) times and start and end locations as well as traveling times between all pairs of end stations, the objective is to find an assignment of trips to vehicles such that each trip is covered exactly once, each vehicle performs a feasible sequence of trips and the overall costs are minimized." The complexity of some different variants of the (VSP) is regarded by [LK81].
A similar problem formulation is the dial-a-ride problem (DARP). [CL07] discuss the differences of the (DARP) to other vehicle routing problems and write: "What makes the DARP different from most such routing problems is the human perspective. When transporting passengers, reducing user inconvenience must be balanced against minimizing operating costs." The basic formulations of (VSP) and (DARP) are the same, therefore we use the formulation (VSP).

Depot Variants

In [BK09], there are two main variants for the (VSP) with respect to the fact where vehicles start and return. In the single depot case (SD-VSP), there is one depot from

where all vehicles start. After usage, all vehicles return to this depot. The multiple depot case (MD-VSP) means that there is more than one depot and from each depot starts a certain number of vehicles. After usage, each vehicle returns to the depot from where it has started.
In order to make our problem more realistic, we have more than one depot. We have more than one vehicle and each vehicle starts from its start position. The vehicles do not have a certain point, where there have to return after usage, i. e. they can stay, wherever the last trip of their duty ends. [DP95] claim that "if the vehicle[s] are allowed to return to a depot different from its origin depot, [...] the problem can be solved as a single depot instance." We see, that our problem is in the single depot case (SD-VSP) concerning the depot variant.
In [DF54] is proven that (SD-VSP) can be solved in polynomial time, i. e. (SD-VSP) is in $\mathcal{P}$. In contrast, the multiple depot case (MD-VSP) is $\mathcal{NP}$-hard as shown in [BCG87].

Additional Constraints

In the basic (SD-VSP), all existing trips have to be fulfilled. For more generality, there were additional cover constraints introduced in the underlying master theses. There are "costumers with sets of alternative trips out of which exactly one trip shall be fulfilled, respectively." ([KK16, p.10]) In our problem, even more general cover constraints are required. We have costumers with sets of alternative routes, consisting of trips; for each costumer, exactly one route has to be fulfilled, i. e. each of its trips is fulfilled. We call these constraints "multi-leg" cover constraints and write the problem (VSP-MC). This is a generalization to the previous cover constraints, as can be seen easily by rewriting them: There are costumers with sets of alternative routes, where each route consists of exactly one trip. According to this reformulation, we call the primary constraints "single-leg" cover constraints and write the problem (VSP-SC). We see in Section 2.4 that (VSP-SC) is already $\mathcal{NP}$-hard.
We further have to respect the fuel constraints. In literature, there are named general resources like time, mileage or fuel, summarized in the general term "route constraints" (cf. [BK09, p. 16], [Raf83]). The respective problems with route constraints are called (SD-VSP-RC) and (MD-VSP-RC). [FP95] describe the (VSP) with time constraints, [Raf83] presents the (VSP) with path constraints, what is a more general formulation. In these models, a vehicle returns to the depot after the respective resource is exhausted, while the vehicle has the possibility to refuel in our model. The problem with not refilling the resource is a special case of the problem with the possibility to refill the resource. We see in Section 2.4 that (SD-VSP-RC) is already $\mathcal{NP}$-hard.
In summary, we have a problem with two types of constraints where each individually makes the problem $\mathcal{NP}$-hard. The first one is the multi-leg cover constraint and the

second one is the fuel constraint. The only difference to [KK16] is that the single-leg cover constraint is replaced by the multi-leg cover constraint. Their solution methods are extended to the requirements in this thesis. To the knowledge of the author, a combination of these constraints, as well as these constraints on their own, are not treated in literature so far.

2.4 Complexity

We regard the complexity of our problem. As we have seen in Section 2.3, the problem can be modeled as a single depot vehicle scheduling problem (SD-VSP) with resource constraints, the possibility of refueling and multi-leg cover constraints. The (SD-VSP) itself can be solved in polynomial time, which is proven by [DF54]. It can be formulated as minimum-cost flow problem. If we extend the basic formulation with one of the additional constraints, it gets $\mathcal{NP}$-hard.

Theorem 1 (VSP with resource constraints)**.** *The vehicle scheduling problem with resource constraints and the special objective of minimizing the number of used vehicles is $\mathcal{NP}$-hard.*

This theorem is proven in [KK16, p. 11] by a polynomial reduction of the bin packing problem. This theorem holds for the case of resource constraints without the possibility of refilling the resources. Since this is a special case of resource constraints with refueling, this problem is also $\mathcal{NP}$-hard.

Theorem 2 (VSP with cover constraints)**.** *The vehicle scheduling problem with cover constraints and the special objective of minimizing the number of used vehicles is $\mathcal{NP}$-hard.*

This theorem is proven in [KK16, p. 12] by a polynomial reduction of the set cover problem. This theorem treats the case of single-leg cover constraints.

Theorem 3 (VSP with multi-leg cover constraints)**.** *The vehicle scheduling problem with multi-leg cover constraints and the special objective of minimizing the number of used vehicles is $\mathcal{NP}$-hard.*

Proof. We prove this statement by a polynomial reduction of the (VSP-SC). Consider a (VSP-SC) with vehicles $\hat{\mathcal{V}}$, costumers $\hat{\mathcal{C}}$, trips $\hat{\mathcal{T}}$ and the function $\hat{C} : \hat{\mathcal{T}} \to \hat{\mathcal{C}}$ that maps trips to their respective costumers. The problem is defined in detail in [KK16, pp. 5-8]. We create the corresponding (VSP-MC) as follows: $\mathcal{V} := \hat{\mathcal{V}}$, $\mathcal{C} := \hat{\mathcal{C}}$, $\mathcal{T} := \hat{\mathcal{T}}$ stay the same. We define the multimodal routes as one-element sequence for each trip

$$\mathcal{M} := \left\{ (t) \mid t \in \hat{\mathcal{T}} \right\}$$

and the mappings

$$M : \mathcal{T} \to \mathcal{M},\ M(t) = (t), \qquad C : \mathcal{M} \to \mathcal{C},\ C(m) = \hat{C}(t) \qquad \text{for } t \in m \text{ unique.}$$

The solution of (VSP-SC) can easily be mapped to a solution of the corresponding (VSP-MC) and vice versa. If a trip is chosen in (VSP-SC), the respective route is chosen in (VSP-MC). Then, every trip of this route is fulfilled and the costumer is satisfied. The other way round, if a route is chosen in (VSP-MC), the trip contained in this route is chosen in (VSP-SC) and the costumer is satisfied. The feasibility of the vehicle duties is not affected by this procedure.
This is a polynomial reduction, (VSP-SC) is is $\mathcal{NP}$-hard by Theorem 2 and hence (VSP-MC) is $\mathcal{NP}$-hard.

□

With Theorem 1 and Theorem 3 we can see, that our problem gets $\mathcal{NP}$-hard only with cover constraints or with resource constraints and considering the number of vehicles. The problem gets even harder, as we do not only consider the number of vehicles but include the operational cost and penalty terms for costumer preferences. Further, we want to have the possibility to refuel during the process, i. e. we have negative resource cost. Finally, we want to apply both of these constraints simultaneously.
We will model our problem as a mixed-integer linear program in a size polynomial in the input size. Therefore, it is possible to verify a solution in polynomial time. This means, our problem is $\mathcal{NP}$-complete.

2.5 Approaches in Literature

In the previous sections we have seen, that the problem we are dealing with is very hard. We cannot expect to find a algorithm that solves the problem running in polynomial time. Therefore we focus on the field of meta heuristics. Using them it is possible to find a good solution for realistic instances in reasonable time. The field of the vehicle scheduling problems is highly researched and there are many solution approaches available in literature. [BK09] provide an extensive overview for both problems (SD-VSP) and (MD-VSP). We have classified our problem as (SD-VSP) with resource constraints and multi-leg cover constrains. This is only a small extension to the problem treated in the underlying theses, which contained single-leg cover constraints. [Kai16] and [Kno16] already discussed the concerning approaches in literature. Since their problem is the basis of our problem and the theses have been published only a short time before this, we can expect their literature approaches to be almost complete. Thus we refer to [Kai16, pp. 13-15] and [Kno16, pp. 13-15] for more details. To the knowledge of the author there is no work of cover constraints in context with vehicle scheduling available in literature.

Electric Vehicle Scheduling Problem

The most related problems in literature are the Alternative-Fuel Vehicle Scheduling Problem (AF-VSP), treated by [Adl14] and the Electric Vehicle Scheduling Problem (E-VSP), treated by [WLR+16]. Both denote a (MD-VSP) with resource constraints and refuel stations. While [Adl14] restrict the problem to full charging in conjunction with a fixed charging time, [WLR+16] allow a partial charging and treats the charging time as a linear function. Thus, the differences of (E-VSP) to our problem is that cover constraints are not considered there.
Besides a mathematical problem formulation, [WLR+16] provide an "Adaptive Large Neighborhood Search" heuristic (ALNS) in order to tackle the problem. We present a rough outline of this algorithm. First, an initial solution is created by a greed heuristic. From this solution, a certain number of trips is deleted (destroy method). After this, several new feasible duties are created by inserting respective trips with lowest cost increasing at certain positions (repair method). Finally, the new duties are selected with minimal total cost, such that the duties cover each trip and the depot constraints are fulfilled. There are several destroy and repair methods presented and there is a deterministic and a randomized version of the heuristic.
We do not examine the possibility of extending (ALNS) in order to cope with cover constraints. We adapt only the solution methods developed by [Kai16] and [Kno16].

Approaches to Model Refuel Points

There is no straight-forward procedure to model the refuel points. It is not possible to use them individually in a task graph since several vehicles may use the same refuel point.
In order to model refuel points for the (VRP), [EMH12] and [nFORT14] include several copies of the refuel points, each for one potential visit between two nodes. This guarantees that a refuel point can be visited more than once. The number of used copies is not specified there. While [EMH12] considers only complete refueling, in [nFORT14] partial refueling is allowed.
In literature concerning the (VSP), [WLR+16] create two copies for each refuel point. One copy is set between the depot and the trip, the other is set after the trip. With this, they guarantee that a vehicle is able to refuel before and after each trip. There are constraints that prevent a car from visiting the same refuel station twice. With this formulation $2|\mathcal{T}||\mathcal{R}|$ refuel point nodes are needed. In other (VSP) models, a preprocessing of the graph removes arcs between trips that cannot be fulfilled after another due to time. Using this refuel point modeling, the complete graph has to be considered in order to model a partial refueling and a feasible timing of the routes is ensured by explicit constraints.

In the underlying theses ([Kai16], [Kno16]), a copy of each refuel point between each pair of feasible trips is created. With this method it is possible to determine the respective charging time individually in advance and modeling partial refueling is easy. In the task graph, time infeasible arcs are removed in advance and are not treated by additional constraints. In the worst case there are needed $\frac{1}{2}|\mathcal{R}||\mathcal{T}|^2$ refuel point nodes. In order to reduce the size of the task graph, they provide a preprocessing on the refuel points nodes, based on Pareto optimality. Using this for realistic instances, most of the copies can be eliminated without cutting the optimal solution.

Chapter 3

Mathematical Models

We introduce the mathematical model, with which we want to solve the previously described problem. First we define the underlying task graph and afterwards we develop an arc flow formulation on this graph. The main idea is that we model a flow of the vehicles to the trips with additional requirements in order to fulfill the cover constraints and the fuel constraints. As mentioned before, we totally neglect public transport trips. They have only an indirect effect on the model by contribution to the trip creation and the route cost.

3.1 Task Graph

We introduce the task graph, on which the model is based. This is a directed graph corresponding to the relation $\prec$ which we defined in Section 2.2. The graph is basically the same as used in [KK16] with the only difference that the customer and route considerations are adapted here.

Definition 2 (Task Graph)**.** Let $d^{\mathrm{s}}, d^{\mathrm{e}}$ be special vertices describing the source and sink of the vehicle flow. We define the task graph as $\hat{G} = (\hat{V}, \hat{A})$, where

FiXme Note: G task graph, Ghat extended task graph

FiXme Note: widehat

$$\hat{V} := \{d^{\mathrm{s}}, d^{\mathrm{e}}\} \mathbin{\dot\cup} \mathcal{V} \mathbin{\dot\cup} \mathcal{T}$$

is the vertex set consisting of the source, the sink, the vehicle set $\mathcal{V}$ and the trip set $\mathcal{T}$. The arc set is

$$\hat{A} := (\{d^{\mathrm{s}}\} \times \mathcal{V}) \mathbin{\dot\cup} \left\{(s,t) \in (\mathcal{V} \mathbin{\dot\cup} \mathcal{T})^2 \,|\, s \prec t\right\} \mathbin{\dot\cup} ((\mathcal{V} \mathbin{\dot\cup} \mathcal{T}) \times \{d^{\mathrm{e}}\})\,.$$

A vertex $s \in \mathcal{V}$ represents the initial state of a vehicle s where it becomes available for the first time. Each d^{s}-d^{e}-path in $\hat{G}$ is the duty of one vehicle, i.e. this vehicle fulfills the trips in the order given by the path. Hence, two trips are connected only if it is possible that one car fulfills both trips, i.e. the relation $\prec$ holds.

Lemma 1. *$\hat{G}$ is a directed acyclic graph.*

Proof. Assume there is a cycle in $\hat{G}$. The source d^{s} and sink d^{e} have only ingoing, respectively outgoing arcs and are therefore not part of the cycle. For $v \in \mathcal{V}$, all ingoing arcs come from d^{s}, hence $v \in \mathcal{V}$ are not part of the cycle, too. This means, a cycle consists only of trips.
Consider an arbitrary cycle of trips $t_1, \ldots, t_k \in \mathcal{T}$, $k \geq 2$. These trips form a cycle, i. e. $t_1 \prec \cdots \prec t_k$ and $t_k \prec t_1$. With Definition 1 and the assumptions (2.4) and (2.5) holds:

$$z_i^{\text{start}} < z_i^{\text{end}} \leq z_i^{\text{end}} + t_{t_i,t_{i+1}} \leq z_{i+1}^{\text{start}} < z_{i+1}^{\text{end}} \qquad \text{for all } i \in [k-1]$$

$$\Rightarrow \qquad z_1^{\text{start}} < z_k^{\text{end}} \qquad \Rightarrow \qquad z_k^{\text{end}} + t_{t_1,t_k} > z_1^{\text{start}} \qquad \Rightarrow \qquad t_k \not\prec t_1$$

This is a contradiction to the existence of a cycle.

□

In order to consider refueling and refuel stations, we introduce an extended task graph.

Definition 3 (Extended Task Graph)**.** For every $s, t \in \mathcal{V} \cup \mathcal{T}$ with $s \prec t$ we create a copy of $\left\{r \in \mathcal{R} | z_s^{\text{end}} + t_{s,r} + t_{r,t} \leq z_t^{\text{start}}\right\}$ denoted by $\mathcal{R}_{s,t}$. This means, various copied sets are pairwise disjoint. The expression $r \in \mathcal{R}_{s,t}$ means that a vehicle is able to finish trip s, then drive to refuel station r and then start trip t in time.
We define the extended task graph $G = (V, A)$ with vertex set

$$V := \hat{V} \cup \bigcup_{\substack{s,t \in \mathcal{V} \cup \mathcal{T} \\ s \prec t}} \mathcal{R}_{s,t}$$

and arc set

$$A := \hat{A} \cup \{(s,r) | s,t \in \mathcal{V} \cup \mathcal{T}, s \prec t, r \in \mathcal{R}_{s,t}\} \cup \{(r,t) | s,t \in \mathcal{V} \cup \mathcal{T}, s \prec t, r \in \mathcal{R}_{s,t}\}\,.$$

It is possible that there is a copy of each refuel station for each feasible pair of trips. This leads to an enormous size of the task graph and thus a bad solution behavior is expected. [KK16, pp. 24-30] describe a method to reduce the size of $\mathcal{R}_{s,t}$ without cutting the optimal solution. This method only considers Pareto optimal refuel station w. r. t. a suitable function. From now on, we will use $G = (V, A)$ with restricted $\mathcal{R}_{s,t}$.

Lemma 2. *G is a directed acyclic graph.*

Proof. Assume there is a cycle in G. In comparison to $\hat{G}$, only arcs (s, r) and (r, t) for $r \in \mathcal{R}_{s,t}, s \prec t$ were added. Assume there is a cycle containing $r \in \mathcal{R}_{s,t}$. r has only one ingoing arc (s, r) and one outgoing arc (r, t) and only if the arc (s, t) exists. There is no cycle on the vertices $\{s, r, t\}$. Every other cycle containing r is also a cycle using the arc (s, t). This is a contradiction to the fact that $\hat{G}$ is cycle-free as proven in Lemma 1.

□

In the extended task graph G, a d^{s}-d^{e}-path further represents the duty of a vehicle. The additional arcs $(s,r),(r,t)$ for $r \in \mathcal{R}_{s,t}$ describe a possible detour between the trips s and t in order to refuel at refuel station r.
We introduce the following frequently used notation:

$$\mathrm{N}_G^-(t) := \{s \in V \mid (s,t) \in A\} \qquad\qquad \mathrm{N}_G^+(t) := \{t \in V \mid (s,t) \in A\}$$

$\mathrm{N}_G^-(t)$ is the set of in-neighbors of $t \in V$, $\mathrm{N}_G^+(s)$ is the set of out-neighbors of $s \in V$.

3.2 Arc Flow Formulation

In the following, we model the problem via a flow of the vehicles on the extended task graph. The trips and multimodal routes are given in advance. The fact, whether two trips can be fulfilled subsequently in one duty, is already given by the underlying task graph. We additionally have to model the cover constraints and the fuel constraints. Since the duties of various vehicles are disjoint w.r.t. $\mathcal{T}$, we are able to use one common set of variables for the flow of one vehicle. From the flow, we can easily extract the individual d^{s}-d^{e}-paths in order to identify the duties of the respective vehicles.

Basic Model

We model the arc flow as a mixed-integer linear program. The formulation is basically built on the (MILP) formulation as described in [KK16, p. 34]. We use the following decision variables:

- $x_{s,t} \in \{0,1\}$ for $(s,t) \in A$: indicates, whether trip $t \in \mathcal{T}$ is fulfilled after $s \in \mathcal{V} \cup \mathcal{T}$
- $z_{s,r,t} \in \{0,1\}$ for $t \in \mathcal{T}, s \in \mathrm{N}_G^-(t), r \in \mathcal{R}_{s,t}$: indicates, whether refuel station $r \in \mathcal{R}$ is visited between s and t
- $e_s \in [0,1]$ for $s \in V \setminus \{d^{\mathrm{s}}, d^{\mathrm{e}}\}$: states the fuel of the respective vehicle after fulfilling trip $s \in \mathcal{T}$

If $s \in \mathcal{V}$, then $x_{s,t}$ determines whether trip t is the first trip fulfilled by s and e_s is the initial fuel state f_s^0 of vehicle s.
Additionally to (MILP), we introduce decision variables in order to ensure the cover constraints:

- $u_m \in \{0,1\}$ for $m \in \mathcal{M}$: indicates whether multimodal route m is fulfilled

The basic constraints are developed in [KK16, pp. 21-34] and not explained in detail here. The basic constraints are the flow conservation constraint, the constraint for considering every car and the constraints ensuring feasible fuel states all the time.

Customer and Route Constraints

In (MILP), each customer has a set of alternative trips and from this set, exactly one trip has to be fulfilled. This is modeled as follows:

$$\sum_{t \in C^{-1}(c)} \sum_{s \in \mathrm{N}_G^-(t)} x_{s,t} = 1 \qquad \text{for all } c \in \mathcal{C} \tag{3}$$

In contrast to (MILP), here each customer has a set of alternative routes consisting of trips and from this set, exactly one route has to be fulfilled. Therefore, we replace (3) by the following formulation:

$$\sum_{m \in C^{-1}(c)} u_m = 1 \qquad \text{for all } c \in \mathcal{C} \tag{3.1}$$

$$\sum_{s \in \mathrm{N}_G^-(t)} x_{s,t} = u_m \qquad \text{for all } m \in \mathcal{M}, t \in m \tag{3.2}$$

The constraint (3.1) says, that for every customer exactly one route is fulfilled. The constraint (3.2) says, if a route is fulfilled then every trip of this route must be fulfilled.

Objective Function

The objective function in (MILP) is given by

$$\sum_{s \in \mathcal{V}} \sum_{t \in \mathrm{N}_G^+(s) \setminus \{d^{\mathrm{e}}\}} x_{s,t} c^{\mathrm{v}} + \sum_{t \in \mathcal{T}} \sum_{s \in \mathrm{N}_G^-(t)} \left[x_{s,t} \left(c^{\mathrm{d}}_{s,t} + c^{\mathrm{t}}_t \right) + \sum_{r \in \mathcal{R}_{s,t}} z_{s,r,t} \left(c^{\mathrm{d}}_{s,r} + c^{\mathrm{d}}_{r,t} - c^{\mathrm{d}}_{s,t} \right) \right]$$

considering the vehicle costs c^{v}, the trip costs c^{t}_t for $t \in \mathcal{T}$ and the deadhead costs c^{d}. What is missing, are the route-dependent costs c^{r}. Thus, we add the term

$$\sum_{m \in \mathcal{M}} u_m c^{\mathrm{r}}_m$$

to the objective function.

LP Formulation

Putting all this together, we get the following formulation, called (MMILP):

$$
\begin{aligned}
\min \quad & \sum_{s\in\mathcal{V}} \sum_{t\in \mathrm{N}_G^+(s)\setminus\{d^{\mathrm{e}}\}} x_{s,t} c^{\mathrm{v}} + \sum_{m\in\mathcal{M}} u_m c_m^{\mathrm{r}} \\
& + \sum_{t\in\mathcal{T}} \sum_{s\in \mathrm{N}_G^-(t)} \left[x_{s,t} \left(c_{s,t}^{\mathrm{d}} + c_t^{\mathrm{t}} \right) + \sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t} \left(c_{s,r}^{\mathrm{d}} + c_{r,t}^{\mathrm{d}} - c_{s,t}^{\mathrm{d}} \right) \right] && \text{(MMILP)} \\
\text{s.t.} \quad & \sum_{s\in \mathrm{N}_G^-(t)} x_{s,t} = \sum_{s\in \mathrm{N}_G^+(t)} x_{t,s} && \text{for all } t \in V\setminus\{d^{\mathrm{s}}, d^{\mathrm{e}}\} && (3.3) \\
& \sum_{s\in \mathrm{N}_G^-(t)} x_{s,t} = 1 && \text{for all } t \in \mathcal{V} && (3.4) \\
& \sum_{m\in C^{-1}(c)} u_m = 1 && \text{for all } c \in \mathcal{C} && (3.1) \\
& \sum_{s\in \mathrm{N}_G^-(t)} x_{s,t} = u_m && \text{for all } m \in \mathcal{M}, t \in m && (3.2) \\
& \sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t} \le x_{s,t} && \text{for all } t \in \mathcal{T}, s \in \mathrm{N}_G^-(t) && (3.5) \\
& e_s \le f_s^0 && \text{for all } s \in \mathcal{V} && (3.6) \\
& 0 \le e_s - \sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t} f_{s,r}^{\mathrm{d}} && \text{for all } t \in \mathcal{T}, s \in \mathrm{N}_G^-(t) && (3.7) \\
& e_t \le 1 - f_t^{\mathrm{t}} - \sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t} f_{r,t}^{\mathrm{d}} && \text{for all } t \in \mathcal{T}, s \in \mathrm{N}_G^-(t) && (3.8) \\
& e_t \le e_s - x_{s,t} \left(f_{s,t}^{\mathrm{d}} + f_t^{\mathrm{t}} \right) - \sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t} \left(f_{s,r}^{\mathrm{d}} + f_r^{\mathrm{t}} + f_{r,t}^{\mathrm{d}} - f_{s,t}^{\mathrm{d}} \right) + (1 - x_{s,t}) \\
& && \text{for all } t \in \mathcal{T}, s \in \mathrm{N}_G^-(t) && (3.9) \\
& x_{s,t} \in \{0,1\} && \text{for all } (s,t) \in A && (3.10) \\
& z_{s,r,t} \in \{0,1\} && \text{for all } t \in \mathcal{T}, s \in \mathrm{N}_G^-(t), r \in \mathcal{R}_{s,t} && (3.11) \\
& e_s \in [0,1] && \text{for all } s \in V\setminus\{d^{\mathrm{s}}, d^{\mathrm{e}}\} && (3.12) \\
& u_m \in \{0,1\} && \text{for all } m \in \mathcal{M} && (3.13)
\end{aligned}
$$

Chapter 4

Successive Heuristics

In this chapter, successive heuristics are introduced in order to solve our problem. As seen in Section 2.4, the problem is $\mathcal{NP}$-hard even if we apply one of the restrictions, the cover constraints or the fuel constraints, individually. Our goal is to develop a heuristic that can cope with both multi-leg cover constraints and fuel constraints. We build our heuristic on a heuristic for a simpler version of the problem, developed in the underlying theses. [Kno16] present heuristical solution methods for the problem only with fuel constraints. The problem settings assumes that there is a set of trips where each of these trips shall be fulfilled. They already claim, that solving a complete instance of 24 hours to optimality is not possible with their respective computing capacity. Therefore it is a plausible assumption that an optimal solution for our problem cannot be expected in reasonable time.

Their solution methods are based on the idea of splitting the complete instance according to several time intervals. For each interval, only the trips starting in the respective interval are considered. From this formulation emerge several separate partial instances that are still loosely connected to each other. Each of these partial instances is solved separately and then the partial solutions are connected to a complete feasible solution. Two different approaches are presented in order to solve the problem: The constraints connecting the partial instances are relaxed by using Lagrange Relaxation. With suitable computation of Lagrange multipliers, the partial instances are solved in parallel. In the other one, the partial instances are solved successively, where the respective connecting constraints are fixed beginning from the end.

An adaption of the cover constraints to the heuristic using Lagrange Relaxation seems not practicable. This heuristics heavily exploits the loosely connection of the partial instances. The cover constraints strongly influence the complete instance by selecting the fulfilled trips, the multi-leg cover constraints even require an additional set of variables, belonging to none of the partial instances. Therefore, an additional relaxing of these cover constraints is not a promising approach. Instead, we focus on the second approach of Successive Heuristics.

The crucial difficulty for this procedure is to ensure the customer satisfaction. In particular, if trips of a customer are wide apart in terms of time, these trips will lie in different splittings. This makes it hard to keep control over the trip selection in

separately solved partial instances.
We first define the splitting of the instance and the arising adaptions of task graph and model. Then, we describe the heuristic in general. Finally, we introduce different splitting methods, one according to the customers and one according to time.

4.1 Successive Heuristics

4.1.1 Splitting the Problem

In order to create the partial instances, we define splittings of $\mathcal{V}$ and $\mathcal{T}$. In contrast to [Kno16], we define the splittings in a general way.

Definition 4 (Splitting)**.** Let $n \in \mathbb{N}$ and let

$$\mathcal{T} = \bigcup_{i=1}^{n} \mathcal{T}_i \qquad\qquad \mathcal{V} = \bigcup_{i=1}^{n} \mathcal{V}_i$$

FiXme Note: Remove splitting of vehicles

be partitions of the set of trips, respectively vehicles. Then we call $\{\mathcal{T}_i \mid i \in [n]\}$ and $\{\mathcal{V}_i \mid i \in [n]\}$ splitting of $\mathcal{T}$ and $\mathcal{V}$ and $\mathcal{T}_i$ and $\mathcal{V}_i$ partial trip respectively vehicle set.

Adaption of the Task Graph

We transform our task graph such that it contains the splittings as defined in Definition 4. For this, we introduce so called split points connecting the partial sets. Arcs that connected two partial sets before, take a detour over the respective split point in the transformed graph.

Definition 5 (Transformed Task Graph)**.** Let $\{\mathcal{T}_1, \dots \mathcal{T}_n\}$ be a splitting of $\mathcal{T}$ according to Definition 4. Then we define:

1. Split Point: Let $s \in \mathcal{T}_i$ for $i \in [n]\backslash\{1\}$. For $j \in [i-1]$, we define the split point $\mathrm{SP}_j(s)$ with $p^{\text{start}}_{\mathrm{SP}_j(s)} = p^{\text{end}}_{\mathrm{SP}_j(s)} =: p^{\text{start}}_s, z^{\text{start}}_{\mathrm{SP}_j(s)} = z^{\text{end}}_{\mathrm{SP}_j(s)} =: z^{\text{start}}_s$ and $f^{\text{t}}_{\mathrm{SP}_j(s)} =: 0$.
2. For $i \in [n]\backslash\{1\}$ and $j \in [i-1]$, we define $\mathcal{P}_{j,i} := \{\mathrm{SP}_j(s) \mid s \in \mathcal{T}_i\}$.
3. Partial Split Point Set: For $j \in [n-1]$, we define the partial split point set $\mathcal{P}_j := \bigcup_{i=j+1}^{n} \mathcal{P}_{j,i}$.
4. Split Point Set: We define the split point set $\mathcal{P} := \bigcup_{j=1}^{n-1} \mathcal{P}_j$.

Let $G = (V, A)$ be the task graph, $\{\mathcal{V}_1, \dots, \mathcal{V}_n\}$ a splitting of $\mathcal{V}$.

5. Transformed Task Graph: We define the transformed task graph $\overline{G} = \left(\overline{V}, \overline{A}\right)$ with vertex set

$$\overline{V} := V \cup \mathcal{P} = V \cup \{\mathrm{SP}_i(s) \mid i \in [n-1], j \in [n+1] \backslash [i], s \in \mathcal{T}_j\}$$

and arc set

$$\begin{aligned}\overline{A} := & (d^{\mathrm{s}} \times \mathcal{V}) \cup \bigcup_{i=1}^{n} \{(s,t) \in (\mathcal{V}_i \cup \mathcal{T}_i) \times (\mathcal{T}_i \cup \mathcal{P}_i) \mid s \prec t\} \\ & \cup \bigcup_{i=1}^{n} \left\{ (s,t) \in \left(\left(\bigcup_{j=1}^{i-1} \mathcal{P}_{j,i} \right) \times \mathcal{T}_i \right) \mid s = \mathrm{SP}_i(t) \right\} \cup ((\mathcal{V} \cup \mathcal{T}) \times \{d^{\mathrm{e}}\})\end{aligned}$$

Adaption of the Model

In order to adapt (MMILP) to the transformed task graph, we make the following considerations:
For all split points we define the costs and fuel states as

$$c_s^{\mathrm{t}} := 0 \qquad c_{s,t}^{\mathrm{d}} := 0 \qquad f_s^{\mathrm{t}} := 0 \qquad f_{s,t}^{\mathrm{d}} := 0 \qquad \text{for } s \in \mathcal{P}, t \in \mathrm{N}_{\overline{G}}^{+}(s)$$

since $p_s^{\mathrm{end}} = p_t^{\mathrm{start}}$ and $z_s^{\mathrm{end}} = z_t^{\mathrm{start}}$. Furthermore, refueling is not possible between s and t.
In the transformed task graph, the arcs between two trips of different splittings are replaced by the detour over the splitting point. Therefore, the trip costs of trip directly after a split point are not considered in the objective function any more. In order to compensate this, we add the following term to the objective function:

$$\sum_{s \in \mathcal{P}} \sum_{t \in \mathrm{N}_{\overline{G}}^{+}(s)} x_{s,t} c_t^{\mathrm{t}}$$

We want to ensure the flow conservation also in the new nodes $\mathcal{P}$, thus we add the inequality

$$\sum_{t \in \mathrm{N}_{\overline{G}}^{-}(s)} x_{t,s} = \sum_{t \in \mathrm{N}_{\overline{G}}^{+}(s)} x_{s,t} \qquad \text{for all } s \in \mathcal{P} \tag{4.1}$$

The equations (3.3) and (4.1) are contracted to

$$\sum_{t \in \mathrm{N}_{\overline{G}}^{-}(s)} x_{t,s} = \sum_{t \in \mathrm{N}_{\overline{G}}^{+}(s)} x_{s,t} \qquad \text{for all } s \in \overline{V} \backslash \{d^{\mathrm{s}}, d^{\mathrm{e}}\} \tag{4.2}$$

$$
\begin{aligned}
\min \quad & \sum_{s\in\mathcal{V}} \sum_{t\in \mathrm{N}^+_{\overline{G}}(s)\setminus\{d^{\mathrm{e}}\}} x_{s,t}c^{\mathrm{v}} + \sum_{s\in\mathcal{P}} \sum_{t\in \mathrm{N}^+_{\overline{G}}(s)} x_{s,t}c^{\mathrm{t}}_t + \sum_{m\in\mathcal{M}} u_m c^{\mathrm{r}}_m \\
& + \sum_{t\in\mathcal{T}\cup\mathcal{P}} \sum_{s\in \mathrm{N}^-_{\overline{G}}(t)\setminus\mathcal{P}} \left[x_{s,t}\left(c^{\mathrm{d}}_{s,t} + c^{\mathrm{t}}_t\right) + \sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t}\left(c^{\mathrm{d}}_{s,r} + c^{\mathrm{d}}_{r,t} - c^{\mathrm{d}}_{s,t}\right)\right] && \text{(SMILP)} \\
\text{s.t.} \quad & \sum_{t\in \mathrm{N}^-_{\overline{G}}(s)} x_{t,s} = \sum_{t\in \mathrm{N}^+_{\overline{G}}(s)} x_{s,t} && \text{for all } s\in\overline{V}\setminus\{d^{\mathrm{s}},d^{\mathrm{e}}\} && (4.2) \\
& \sum_{s\in \mathrm{N}^-_{\overline{G}}(t)} x_{s,t} = 1 && \text{for all } t\in\mathcal{V} && (3.4) \\
& \sum_{m\in C^{-1}(c)} u_m = 1 && \text{for all } c\in\mathcal{C} && (3.1) \\
& \sum_{s\in \mathrm{N}^-_{\overline{G}}(t)} x_{s,t} = u_m && \text{for all } m\in\mathcal{M}, t\in m && (3.2) \\
& \sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t} \le x_{s,t} && \text{for all } t\in\mathcal{T}\cup\mathcal{P}, s\in \mathrm{N}^-_{\overline{G}}(t)\setminus\mathcal{P} && (4.3) \\
& e_s \le f^0_s && \text{for all } s\in\mathcal{V} && (3.6) \\
& 0 \le e_s - \sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t} f^{\mathrm{d}}_{s,r} && \text{for all } t\in\mathcal{T}\cup\mathcal{P}, s\in \mathrm{N}^-_{\overline{G}}(t)\setminus\mathcal{P} && (4.4) \\
& e_t \le 1 - f^{\mathrm{t}}_t - \sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t} f^{\mathrm{d}}_{r,t} && \text{for all } t\in\mathcal{T}\cup\mathcal{P}, s\in \mathrm{N}^-_{\overline{G}}(t)\setminus\mathcal{P} && (4.5) \\
& e_t \le e_s - x_{s,t}\left(f^{\mathrm{d}}_{s,t} + f^{\mathrm{t}}_t\right) - \sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t}\left(f^{\mathrm{d}}_{s,r} + f^{\mathrm{t}}_r + f^{\mathrm{d}}_{r,t} - f^{\mathrm{d}}_{s,t}\right) + (1 - x_{s,t}) \\
& && \text{for all } t\in\mathcal{T}\cup\mathcal{P}, s\in \mathrm{N}^-_{\overline{G}}(t)\setminus\mathcal{P} && (4.6) \\
& e_t \le e_s - x_{s,t} f^{\mathrm{t}}_t + (1 - x_{s,t}) && \text{for all } s\in\mathcal{P}, t\in \mathrm{N}^+_{\overline{G}}(s) && (4.10) \\
& x_{s,t} \in \{0,1\} && \text{for all } (s,t)\in\overline{A} && (4.7) \\
& z_{s,r,t} \in \{0,1\} && \text{for all } t\in\mathcal{T}\cup\mathcal{P}, s\in \mathrm{N}^-_{\overline{G}}(t)\setminus\mathcal{P}, r\in\mathcal{R}_{s,t} && (4.8) \\
& e_s \in [0,1] && \text{for all } s\in\overline{V}\setminus\{d^{\mathrm{s}},d^{\mathrm{e}}\} && (4.9) \\
& u_m \in \{0,1\} && \text{for all } m\in\mathcal{M} && (3.13)
\end{aligned}
$$

The fuel constraints are adapted in the following way: (3.5), (3.7), (3.8) and (3.9) hold also on the arcs leading to $\mathcal{P}$ and are therefore replaced by (4.3), (4.4), (4.5) and (4.6). Further the arcs leading from a split points to its respective trips have to be considered. Since refueling is not possible there, we have only to adapt (3.9). Since $f^{\mathrm{d}}_{s,t} = 0$ and refueling is not possible between s and t, the constraint reads as follows:

$$e_t \leq e_s - x_{s,t} f_t^{\mathrm{t}} + (1 - x_{s,t}) \qquad \text{for all } s \in \mathcal{P}, t \in \mathrm{N}_{\overline{G}}^{+}(s) \tag{4.10}$$

The customer constraints (3.1) are not affected by transforming the graph. The decision whether a trip $t \in \mathcal{T}$ is fulfilled is still given by $\sum_{s \in \mathrm{N}_{\overline{G}}^{-}(t)} x_{s,t}$, no matter if the ingoing arc is a split point or not. Thus, the route constraints (3.2) do not change either.

Putting all together, we have the formulation (SMILP).

4.1.2 General Setting

In this section, we describe the general setting of the Successive Heuristics. First we describe how the partial task graph is created, given a splitting of $\mathcal{V}$ and $\mathcal{T}$. It is based on $\overline{G}$ and contains start and end points, which are created in the partial instances solved before this. Then we treat the order in which the partial instances are solved. The first partial instance is a special instance since there the vehicles come into play. Therefore, this instance is solved last. We explain how start and end points are are created out of a partial solution. Finally, we describe the feasible connection of the partial instances to an overall solution.

Order of Solving the Partial Instances

Consider a splitting $\{\mathcal{T}_1, \dots, \mathcal{T}_n\}$ and $\{\mathcal{V}_1, \dots, \mathcal{V}_n\}$ for $\mathcal{T}$ and $\mathcal{V}$ respectively. Let $\sigma \in S_n$ be a permutation of $[n]$ with $\sigma(n) = 1$. σ indicates in which order the partial instances are solved. This means, partial instance $\sigma(i) \in [n]$ is solved at the i-th position, the first partial instance is solved at last. The actual specification of σ follows in the description of the respective heuristic.

Determination of Start and End Points

The sets of start and end points $\hat{\mathcal{V}}_i, \hat{\mathcal{P}}_i$ are initially empty for all $i \in [n]$. Assume, we have solved the partial instance $\sigma(i)$ just now. Based on the received partial solution, we update the start point set of the next partial instance after $\sigma(i)$ and we update the end point set of the next partial instance before $\sigma(i)$ which is not yet solved. This means, we update $\hat{\mathcal{V}}_{\sigma(j)}$ and $\hat{\mathcal{P}}_{\sigma(k)}$ for $j = \arg\min_{j>i} \{\sigma(j) \mid \sigma(j) > \sigma(i)\}$ and $k = \arg\min_{k>i} \{\sigma(k) \mid \sigma(k) < \sigma(i)\}$.

For each duty of the partial solution that does either visit no node out of $\hat{\mathcal{V}}_{\sigma(i)}$ or no node out of $\hat{\mathcal{P}}_{\sigma(i)}$, we create a start point and/or an end point. If a duty starts with a trip or an end point s, we create an end point out of it. The end point t has the following properties

$$p_t^{\text{start}} = p_t^{\text{end}} := p_s^{\text{start}} \qquad z_t^{\text{start}} = z_t^{\text{end}} := z_s^{\text{start}} \qquad f_t^0 := e_s + f_s^{\text{t}}$$

where e_s is the respective value of decision variable e in the partial solution. We add t to the end point set $\hat{\mathcal{P}}_{\sigma(k)}$. If a duty ends with a trip or a start point s, we create a start point out of it. The start point t has the following properties

$$p_t^{\text{start}} = p_t^{\text{end}} := p_s^{\text{end}} \qquad z_t^{\text{start}} = z_t^{\text{end}} := z_s^{\text{end}} \qquad f_t^0 := e_s$$

where e_s is the respective value of decision variable e in the partial solution. We add t to the start point set $\hat{\mathcal{V}}_{\sigma(j)}$.
For each start or end point s, we call the respective trip t from where it is created, the trip representing t. If a start point t is created from a start point s, the trip representing t is the trip representing s. Note, that each start point $s \in \bigcup_{i=1}^{n} \hat{\mathcal{P}}_i$ has a trip $t \in \mathcal{T}$ representing it. This works analogously for end points.

Remark 2. Since $\sigma(n) = 1$, the set $\{k > i \mid \sigma(k) < \sigma(i)\}$ is never empty for $i \in [n-1]$. Therefore, it is always possible to create an end point. If there is no later partial instance left, which is not yet solved, i. e. $\{j > i \mid \sigma(j) > \sigma(i)\} = \emptyset$, then we create no start points out of $\sigma(i)$.
If a duty consists of exactly one trip, then for this trip we create both a start and an end point.

The partial instance I_1 has a special role. In this instance, we have no start points, but we include the vehicle set $\mathcal{V}$. Therefore, we set $\hat{\mathcal{V}}_1 := \mathcal{V}$. Since I_1 is solved last, here is decided which vehicles are actually used.

Feasible Connection of Partial Solutions

In order to generate an overall solution which is feasible for (MMILP), we connect the partial solutions. Let $\{S_1, \dots \S_n\}$ be the partial solutions, solved as described before with the start and end points created as before. The connection works as follows:
For each duty of S_1, we check whether it ends with an end point $t \in \bigcup_{i=1}^{n} \hat{\mathcal{P}}_i$. We call this duty start duty.

- If it does, we delete the end point and append the duty of S_i for $i \in [n]\backslash\{1\}$, that starts with the trip representing t, to the start duty. We then restart this procedure with the new end of the start duty.
- If it does not, it ends with a trip or start point $t \in \left(\mathcal{T} \,\dot\cup \bigcup_{i=1}^{n} \hat{\mathcal{V}}_i\right)$. We check whether there is a duty in S_i for $i \in [n]\backslash\{1\}$, that starts with a start point represented by t. If such a duty exists, we append it without the start point to the start duty. We then restart this procedure with the new end of the start duty. If no such duty exists, this start duty is finished.

Algorithm 1: Successive Heuristic (general setting)

Input: splitting $\mathcal{T} = \{\mathcal{T}_1, \dots, \mathcal{T}_n\}$, $\mathcal{V} = \{\mathcal{V}_1, \dots, \mathcal{V}_n\}$, $\sigma \in S_n$ with $\sigma(n) = 1$
Output: overall solution S

foreach $i \in [n]$ **do** $\hat{\mathcal{V}}_{\sigma(i)} \leftarrow \emptyset$, $\hat{\mathcal{P}}_{\sigma(i)} \leftarrow \emptyset$;
foreach $i \in [n-1]$ **do**
  solve $I_{\sigma(i)}$, receive partial solution $S_{\sigma(i)}$ with duty set $D_{\sigma(i)}$;
  $j = \arg\min_{j>i} \{\sigma(j) \mid \sigma(j) > \sigma(i)\}, k = \arg\min_{k>i} \{\sigma(k) \mid \sigma(k) < \sigma(i)\}$;
  foreach $D_{\sigma(i)} \ni d = (s_1, \dots, s_l)$ **do**
    if $s_1 \in \mathcal{T}_{\sigma(i)} \cup \hat{\mathcal{P}}_{\sigma(i)}$ **then**
      create end point t;
      $p_t^{\text{start}} \leftarrow p_{s_1}^{\text{start}}, p_t^{\text{end}} \leftarrow p_{s_1}^{\text{start}}, z_t^{\text{start}} \leftarrow z_{s_1}^{\text{start}}, z_t^{\text{end}} \leftarrow z_{s_1}^{\text{start}}, f_t^0 \leftarrow e_{s_1} + f_{s_1}^{\text{t}}$;
      $\hat{\mathcal{P}}_{\sigma(k)} \leftarrow \hat{\mathcal{P}}_{\sigma(k)} \cup \{t\}$;
    end
    if $s_l \in \hat{\mathcal{V}}_{\sigma(i)} \cup \mathcal{T}_{\sigma(i)}$ **then**
      create start point t;
      $p_t^{\text{start}} \leftarrow p_{s_l}^{\text{end}}, p_t^{\text{end}} \leftarrow p_{s_l}^{\text{end}}, z_t^{\text{start}} \leftarrow z_{s_l}^{\text{end}}, z_t^{\text{end}} \leftarrow z_{s_l}^{\text{end}}, f_t^0 := e_{s_l}$;
      $\hat{\mathcal{V}}_{\sigma(j)} \leftarrow \hat{\mathcal{V}}_{\sigma(j)} \cup \{t\}$;
    end
  end
end
$\hat{\mathcal{V}}_1 \leftarrow \mathcal{V}$;
solve I_1, receive partial solution S_1 with duty set D_1;
foreach $D_1 \ni d = (s_1, \dots, s_l)$ **do**
  repeat
    if $s_l \in \bigcup_{i=1}^n \hat{\mathcal{P}}_i$ **then**
      determine duty $d' = (t_1, \dots, t_{l'})$ with t_1 representing s_l;
      $d \leftarrow (s_1, \dots, s_{l-1}, t_1, \dots, t_{l'})$;
    end
    else if $\exists d' \in \bigcup_{i=1}^n D_i$ *s. t.* s_l *represents* $t_1 \in \bigcup_{i=1}^n \hat{\mathcal{V}}_i$ **then**
      determine duty $d' = (t_1, \dots, t_{l'})$;
      $d \leftarrow (s_1, \dots, s_l, t_2, \dots, t_{l'})$;
    end
    else Duty finished;
  until *Duty finished*;
end
return S *with duties* $d \in D_1$

4.1.3 Identifying the Subproblems

Given a splitting of $\mathcal{T}$ and $\mathcal{V}$, we describe how the subproblems of (SMILP) are created. For each partial trip and vehicle set $\mathcal{T}_i$, $\mathcal{V}_i$, we solve a partial instance I_i. We call the solution of a partial instance I_i partial solution S_i.

Partial Instances

First we define the task graph with which we can solve the partial instances. The transformed task graph $\overline{G}$ covers the complete instance, but contains the partial sets from the splittings of $\mathcal{V}$ and $\mathcal{T}$. The partial task graph $\overline{G}_i$ only contains the respective partial trip and vehicle set. It additionally contains a start point set $\hat{\mathcal{V}}_i$ and an end point set $\hat{\mathcal{P}}_i$. How these sets are defined is explained later.

Definition 6 (Partial Transformed Task Graph)**.** Let $i \in [n]$. For a set of start points $\hat{\mathcal{V}}_i$, a set of end points $\hat{\mathcal{P}}_i$ and the partial trip set $\mathcal{T}_i$, the partial transformed task graph is the directed graph $\overline{G}_i = \left(\overline{V}_i, \overline{A}_i\right)$ with vertex set

$$\overline{V}_i := \{d^{\mathrm{s}}, d^{\mathrm{e}}\} \cupdot \hat{\mathcal{V}}_i \cupdot \mathcal{T}_i \cupdot \hat{\mathcal{P}}_i$$

and arc set

$$\begin{aligned}\overline{A}_i := &\left(\{d^{\mathrm{s}}\} \times \left(\hat{\mathcal{V}}_i \cupdot \mathcal{T}_i \cupdot \hat{\mathcal{P}}_i\right)\right) \cupdot \left\{(s,t) \in \left(\hat{\mathcal{V}}_i \cupdot \mathcal{T}\right) \times \left(\mathcal{T}_i \cupdot \hat{\mathcal{P}}_i\right) \mid s \prec t\right\}\\ &\cupdot \left(\left(\hat{\mathcal{V}}_i \cupdot \mathcal{T}_i \cupdot \hat{\mathcal{P}}_i\right) \times \{d^{\mathrm{e}}\}\right)\end{aligned}$$

Solving of the Partial Instances

In order to solve each partial instance, we create a formulation which is based on the partial transformed task graph $\overline{G}_i$. The flow constraints and the fuel constraints are basically the same as in (SMILP), restricted to $\overline{G}_i$. This formulation is only a basic structure for the partial instances. Some details are depending on the actual choice of the splitting and left out here. These details are inserted later.
In the partial instance, there is no vehicle set anymore. Instead we ensure that each start and each end point is visited. Therefore we replace (3.4) by

$$\sum_{s \in \mathrm{N}^{-}_{\overline{G}_i}(t)} x_{s,t} = 1 \qquad \text{for all } t \in \hat{\mathcal{V}}_i \cup \hat{\mathcal{P}}_i \tag{4.11}$$

For start and end points, we are given initial fuel levels. The initial fuel level for a start point indicates the still available fuel from the end of the previous partial duty. The initial fuel level for an end point indicates the required fuel for the start of the next partial duty. Therefore, these fuel levels work as boundaries for the duties considered in this partial instance. In order to guarantee this, we introduce the constraints

$$e_s \leq f_s^0 \qquad \text{for all } s \in \hat{\mathcal{V}}_i \tag{4.12}$$
$$f_s^0 \leq e_s \qquad \text{for all } s \in \hat{\mathcal{P}}_i \tag{4.13}$$

Since there are no vehicles in the partial instance, the constraint (3.6) is dropped.
We introduce two additional constraints. If a duty starts or ends with a trip, then the fuel at the start or at the end of this duty has to be in certain boundaries $f^{\min}$ or $f^{\max}$, respectively. How these boundaries are actually defined, is part of the heuristic. The constraints are the following:

$$e_s + f_s^{\mathrm{t}} \leq f_s^{\max} + (1 - x_{d^{\mathrm{s}},s}) \cdot \left(1 + f_s^{\mathrm{t}}\right) \qquad \text{for all } s \in \mathcal{T}_i \tag{4.14}$$
$$f_s^{\min} \leq e_s + (1 - x_{s,d^{\mathrm{e}}}) \qquad \text{for all } s \in \mathcal{T}_i \tag{4.15}$$

As mentioned before, it requires some additional work to include the cover constraints into the partial instances. The fulfilling of the cover constraints is also part of the respective heuristics and therefore (3.1) and (3.2) are left out in this formulation.

Cost Function

The trip costs are handled as before. If a trip is fulfilled, it is fulfilled in exactly one partial instance and the trip cost is included there. The deadhead costs between two trips $s, t \in \mathcal{T}_i$ are the same as before. If start and end points are included, the behavior is as follows: Each start and end point represents a trip of an already solved partial instance. The deadhead cost after this trip (if it is a start point) or before this trip (if it is an end point) has not occurred up to now. The deadhead cost including a start or an end point is therefore the same as the deadhead cost with the trip representing it.
The fixed vehicle costs require a different treatment. If a duty begins at a start point or finishes at an end point, the vehicles cost of this duty arises already in the partial instance where the start point is created. Therefore, the arcs $\{d^{\mathrm{s}}\} \times \hat{\mathcal{V}}_i$ have no vehicle cost, the arcs $\{d^{\mathrm{s}}\} \times \left(\mathcal{T}_i \cup \hat{\mathcal{P}}_i\right)$ have vehicles cost c^{v}. Since the vehicle cost for an end point are already paid in the other partial instance, the arcs $\hat{\mathcal{P}}_i \times \{d^{\mathrm{e}}\}$ have negative vehicle cost. This means, a duty beginning at a start point and finishing at an end point has negative vehicle cost, since the vehicle cost has already occurred twice, in the partial instance where start, respectively end point was created.
Since the cover constraints are left out in this formulation, also the route cost is not treated here. This is also defined with the specification of the splitting. The formulation of the partial instance is called (SMILP_i) for $i \in [n]\backslash\{1\}$.

$$\min \quad \left(\sum_{s \in \mathcal{T}_i \cup \hat{\mathcal{P}}_i} x_{d^{\mathrm{s}},s} - \sum_{s \in \hat{\mathcal{P}}_i} x_{s,d^{\mathrm{e}}} \right) c^{\mathrm{v}}$$

$$\sum_{t \in \mathcal{T}_i \cup \hat{\mathcal{P}}_i} \sum_{s \in \mathrm{N}^-_{\overline{G}_i}(t) \setminus \{d^{\mathrm{s}}\}} \left[x_{s,t} \left(c^{\mathrm{d}}_{s,t} + c^{\mathrm{t}}_t \right) + \sum_{r \in \mathcal{R}_{s,t}} z_{s,r,t} \left(c^{\mathrm{d}}_{s,r} + c^{\mathrm{d}}_{r,t} - c^{\mathrm{d}}_{s,t} \right) \right] \quad (\mathrm{SMILP}_i)$$

$$\text{s.t.} \quad \sum_{t \in \mathrm{N}^-_{\overline{G}_i}(s)} x_{t,s} = \sum_{t \in \mathrm{N}^+_{\overline{G}_i}(s)} x_{s,t} \qquad \text{for all } s \in \overline{V}_i \setminus \{d^{\mathrm{s}}, d^{\mathrm{e}}\} \tag{4.16}$$

$$\sum_{s \in \mathrm{N}^-_{\overline{G}_i}(t)} x_{s,t} = 1 \qquad \text{for all } t \in \hat{\mathcal{V}}_i \cup \hat{\mathcal{P}}_i \tag{4.11}$$

$$\sum_{r \in \mathcal{R}_{s,t}} z_{s,r,t} \le x_{s,t} \qquad \text{for all } t \in \mathcal{T}_i \cup \hat{\mathcal{P}}_i, s \in \mathrm{N}^-_{\overline{G}_i}(t) \setminus \{d^{\mathrm{s}}\} \tag{4.17}$$

$$e_s \le f^0_s \qquad \text{for all } s \in \hat{\mathcal{V}}_i \tag{4.12}$$

$$f^0_s \le e_s \qquad \text{for all } s \in \hat{\mathcal{P}}_i \tag{4.13}$$

$$0 \le e_s - \sum_{r \in \mathcal{R}_{s,t}} z_{s,r,t} f^{\mathrm{d}}_{s,r} \qquad \text{for all } t \in \mathcal{T}_i \cup \hat{\mathcal{P}}_i, s \in \mathrm{N}^-_{\overline{G}_i}(t) \setminus \{d^{\mathrm{s}}\} \tag{4.18}$$

$$e_t \le 1 - f^{\mathrm{t}}_t - \sum_{r \in \mathcal{R}_{s,t}} z_{s,r,t} f^{\mathrm{d}}_{r,t} \qquad \text{for all } t \in \mathcal{T}_i \cup \hat{\mathcal{P}}_i, s \in \mathrm{N}^-_{\overline{G}_i}(t) \setminus \{d^{\mathrm{s}}\} \tag{4.19}$$

$$e_t \le e_s - x_{s,t} \left(f^{\mathrm{d}}_{s,t} + f^{\mathrm{t}}_t \right) - \sum_{r \in \mathcal{R}_{s,t}} z_{s,r,t} \left(f^{\mathrm{d}}_{s,r} + f^{\mathrm{t}}_r + f^{\mathrm{d}}_{r,t} - f^{\mathrm{d}}_{s,t} \right) + (1 - x_{s,t})$$
$$\text{for all } t \in \mathcal{T}_i \cup \hat{\mathcal{P}}_i, s \in \mathrm{N}^-_{\overline{G}_i}(t) \tag{4.20}$$

$$e_s + f^{\mathrm{t}}_s \le f^{\max}_s + (1 - x_{d^{\mathrm{s}},s}) \cdot \left(1 + f^{\mathrm{t}}_s \right) \qquad \text{for all } s \in \mathcal{T}_i \tag{4.14}$$

$$f^{\min}_s \le e_s + (1 - x_{s,d^{\mathrm{e}}}) \qquad \text{for all } s \in \mathcal{T}_i \tag{4.15}$$

$$x_{s,t} \in \{0,1\} \qquad \text{for all } (s,t) \in \overline{A}_i \tag{4.21}$$

$$z_{s,r,t} \in \{0,1\} \qquad \text{for all } t \in \mathcal{T}_i \cup \hat{\mathcal{P}}_i, s \in \mathrm{N}^-_{\overline{G}_i}(t) \setminus \{d^{\mathrm{s}}\}, r \in \mathcal{R}_{s,t} \tag{4.22}$$

$$e_s \in [0,1] \qquad \text{for all } s \in \overline{V}_i \setminus \{d^{\mathrm{s}}, d^{\mathrm{e}}\} \tag{4.23}$$

Solving Partial Instance I_1

As mentioned before, the partial instance I_1 plays a special role since here the vehicles are introduced. We have $\hat{\mathcal{V}}_1 = \mathcal{V}$. This requires some changes in the formulation. All duties have to start with a vehicle node, therefore all nodes out of $\{d^{\mathrm{s}}\} \times \left(\mathcal{T}_1 \cup \hat{\mathcal{V}}_1 \right)$ are deleted. This graph is denoted as $\overline{G}_1 = \left(\overline{V}_1, \overline{A}_1\right)$. The objective function is modified, too. Each vehicles causes vehicle costs c^{v}. Thus, the term

$$\left(\sum_{s \in \mathcal{T}_i \cup \hat{\mathcal{P}}_i} x_{d^{\mathrm{s}},s} - \sum_{s \in \hat{\mathcal{P}}_i} x_{s,d^{\mathrm{e}}}\right) c^{\mathrm{v}}$$

in the objective function is replaced by

$$\left(\sum_{s \in \hat{\mathcal{V}}_1} \sum_{t \in \mathrm{N}^{+}_{\overline{G}_1}(s) \setminus \{d^{\mathrm{e}}\}} x_{s,t} - \sum_{s \in \hat{\mathcal{P}}_1} x_{s,d^{\mathrm{e}}}\right) c^{\mathrm{v}}$$

We call this formulation (SMILP$_1$).

4.2 Customer-dependent Splitting

In this section, we introduce customer-dependent splitting. In contrast to the splitting performed by [Kno16], the trips are not split according to their start times but according to their customers' start times. This means, that all trips of a route and all routes of a customer are in the same splitting. This has the advantage, that the cover constraints can be applied easily in the respective subproblems. The problem is that this formulation is potentially not equivalent to the original problem, i. e. duties that are feasible in (MMILP) can be cut in this formulation. We show restrictions, in which the application of this splitting is sensible, though.

Splitting

The customer-dependent splitting is defined as follows:

Definition 7 (customer-dependent Splitting)**.** Given points in time c_i for $i \in [n-1]$ with $c_i < c_{i+1}$ for $i \in [n-2]$. We first define a splitting of the customers $\mathcal{C} = \bigcup_{i=1}^{n} \mathcal{C}_i$ as

$$\mathcal{C}_i := \begin{cases} \{c \in \mathcal{C} \mid z_c^{\text{start}} \leq c_1\} & \text{for } i = 1 \\ \{c \in \mathcal{C} \mid c_{i-1} < z_c^{\text{start}} \leq c_i\} & \text{for } i \in [n-1] \setminus \{1\} \\ \{c \in \mathcal{C} \mid c_{n-1} < z_c^{\text{start}}\} & \text{for } i = n. \end{cases}$$

Based on the customer splitting, we define the splittings of $\mathcal{T}$ and $\mathcal{V}$ as

$$\mathcal{T}_i := \{t \in \mathcal{T} \mid (M \circ C)(t) \in \mathcal{C}_i\} \qquad \text{for } i \in [n]$$

and

$$\mathcal{V}_i := \begin{cases} \{v \in \mathcal{V} \mid z_v \leq c_1\} & \text{for } i = 1 \\ \{v \in \mathcal{V} \mid c_{i-1} < z_v \leq c_i\} & \text{for } i \in [n-1] \backslash \{1\} \\ \{v \in \mathcal{V} \mid c_{n-1} < z_v\} & \text{for } i = n. \end{cases}$$

We denote the formulation (SMILP) with a splitting according to Definition 7 as (CMILP).

Solving of the Partial Instances

The formulation is built on the basic structure (SMILP_i). The cover constraints are not considered so far. We therefore introduce the decision variable $u_m \in \{0,1\}$ for $m \in C^{-1}(\mathcal{C}_i)$. Since for a customer $c \in \mathcal{C}_i$ all trips are in the splitting $\mathcal{T}_i$, only the cover constraints concerning these customers are included in partial instance I_i. We therefore add the following constraints:

$$\sum_{m \in C^{-1}(c)} u_m = 1 \qquad \text{for all } c \in \mathcal{C}_i \tag{4.24}$$

$$\sum_{s \in \mathrm{N}^-_{\overline{G}_i}(t)} x_{s,t} = u_m \qquad \text{for all } m \in C^{-1}(\mathcal{C}_i), t \in m \tag{4.25}$$

$$u_m \in \{0,1\} \qquad \text{for all } m \in C^{-1}(\mathcal{C}_i) \tag{4.26}$$

The route cost is not considered in (SMILP_i) so far. Again, we have to consider only the route costs belonging to $c \in \mathcal{C}_i$. We therefore add the following term to the objective function

$$\sum_{m \in C^{-1}(\mathcal{C}_i)} u_m c^{\mathrm{r}}_m$$

We call this formulation (CMILP_i) for $i \in [n]$.

Model Equivalence

This heuristic formulation is not equivalent to the original formulation (MMILP). This is shown by the following example.

Example 1. Let t_1, t_2, t_3 with $t_1 \prec t_2 \prec t_3$ be trips with the properties shown in Table 4.1
In this case, customer C_1 uses public transport between 8:15 and 9:00. The duty (t_1, t_2, t_3) is a feasible result of the (MMILP).

Trip	Start	End	Route	customer
t_1	8:00	8:15	m_1	C_1
t_2	8:30	8:45	m_2	C_2
t_3	9:00	9:15	m_1	C_1

Table 4.1: Trips

If there is a split point at 8:15 then the splittings are $\mathcal{T}_1 = \{t_1, t_3\}, \mathcal{T}_2 = \{t_2\}$. Hence, there is one split point $\mathrm{SP}_1(t_2)$ with $z^{\text{start}}_{\mathrm{SP}_1(t_2)} = 8{:}30$. The partial solution of instance 1 is (t_1, t_3) and $t_3 \not\prec \mathrm{SP}_1(t_2)$. Thus, the partial solutions cannot be feasibly connected to the solution (t_1, t_2, t_3).

With this example we have seen, that the formulations (CMILP) and (MMILP) are not equivalent. It is even possible, that an optimal solution of (MMILP) is not feasible in (CMILP).
Although the formulations are not equivalent, we can give an estimation on the objective value when we make some restrictions.

Definition 8. For $n \geq 3$, consider a customer set $\mathcal{C}$ and split points c_i for $i \in [n-1]$ with $c_i < c_{i+1}$ for all $i \in [n-2]$. We define the following values:

- customer Extension for $c \in \mathcal{C}$: $L_\mathrm{C}(c) := \max\limits_{t \in (M \circ C)^{-1}(c)} z^{\text{start}}_t - \min\limits_{t \in (M \circ C)^{-1}(c)} z^{\text{start}}_t$
- customer Extension: $L_\mathrm{C} := \max\limits_{c \in \mathcal{C}} L_\mathrm{C}(c)$
- Splitting Length: $L_\mathrm{S} := \min\limits_{i \in [n-1]} c_{i+1} - c_i$

Theorem 4. *For $n \geq 3$, consider the problem with customer set $\mathcal{C}$ and split points c_i for $i \in [n-1]$ with $c_i < c_{i+1}$ for all $i \in [n-2]$. Let*

$$L_\mathrm{C} \leq L_\mathrm{S} \tag{4.27}$$

Let $d := (t_1, \ldots, t_k)$ be the duty of a vehicle of a feasible solution of the (MMILP). Then, there are duties $d_1 \uplus d_2 = d$, where d_1, d_2 are part of a feasible solution of (CMILP)*. Moreover, there holds*

$$\operatorname{cost}(d_1) + \operatorname{cost}(d_2) \leq 2 \cdot \operatorname{cost}(d). \tag{4.28}$$

Proof. We consider the vehicle duty $d = (t_1, \ldots t_k)$. We write $s \prec t$ according to Definition 1, i.e. (s, t) is feasible in (MMILP). We write $s \to t$ iff (s, t) is feasible in (CMILP).

Consider $s \prec t$ with $s \not\to t$ and customers $C_s := (M \circ C)(s)$ and $C_t := (M \circ C)(t)$. Then s is in a later splitting than t. There are split points c_{l-1}, c_l, c_{l+1} for $l \in [n]$ with

$$z_s^{\text{start}} < z_t^{\text{start}} \qquad z_{C_t}^{\text{start}} \leq c_l < z_{C_s}^{\text{start}} \qquad c_l + L_{\text{S}} \leq c_{l+1} \qquad z_{C_s}^{\text{start}} \leq z_s^{\text{start}} \leq z_{C_s}^{\text{start}} + L_{\text{C}}$$

Since (4.27), holds

$$\begin{aligned} z_{C_s}^{\text{start}} \leq z_s^{\text{start}} < z_t^{\text{start}} \leq z_{C_t}^{\text{start}} + L_{\text{C}} \leq c_l + L_{\text{C}} \leq c_l + L_{\text{S}} \leq c_{l+1} \\ z_{C_t}^{\text{start}} \geq z_t^{\text{start}} - L_{\text{C}} > z_s^{\text{start}} - L_{\text{C}} \geq z_{C_s}^{\text{start}} - L_{\text{C}} > c_l - L_{\text{C}} \geq c_l - L_{\text{S}} \geq c_{l-1} \end{aligned}$$

and therefore $t \in \mathcal{T}_l, s \in \mathcal{T}_{l+1}$. Here, we use $c_0 := -\infty, c_{n+1} := +\infty$.

Feasibility

For arbitrary $i \in [k-2]$ holds: $t_i \prec t_{i+1} \prec t_{i+2}$, therefore also $t_i \prec t_{i+2}$. We prove that t_{i+2} can be appended after t_i or t_{i+1}. We differentiate between the following cases:

1. $t_{i+1} \to t_{i+2}$: Clear.
2. $t_{i+1} \not\to t_{i+2}$: Then holds $t_{i+2} \in \mathcal{T}_l$ and $t_{i+1} \in \mathcal{T}_{l+1}$ for some $l \in [k]$. From $t_i \prec t_{i+2}$ follows $t_i \in \bigcup_{j=1}^{l+1} \mathcal{T}_j$. Therefore $t_i \to t_{i+1}$ or $t_i \to t_{i+2}$.
   - $t_i \to t_{i+2}$: Clear.
   - $t_i \not\to t_{i+2}$: Then holds $t_{i+2} \in \mathcal{T}_l$ and $t_i, t_{i+1} \in \mathcal{T}_{l+1}$ and therefore $t_i \to t_{i+1}$. For $i' \geq i$ holds $t_{i'} \in \bigcup_{j=l}^{n} \mathcal{T}_j$ and therefore $t_{i+1} \to t_{i'}$ or $t_{i+2} \to t_{i'}$. Thus, every later trip can be appended after on of these duties.

We have seen that two duties d_1, d_2 can fulfill the trips of duty d, such that d_1 and d_2 are feasible in (CMILP). Each trip can be appended to d_1 or to d_2.

Costs

The costs of duty d are

$$\text{cost}(d) = c^{\text{v}} + c^{\text{d}}_{v,t_1} + c^{\text{t}}_{t_1} + \sum_{i=2}^{k} \left(c^{\text{d}}_{t_{i-1},t_i} + c^{\text{t}}_{t_i} \right).$$

Each duty d_1, d_2 has cost $c^{\text{d}}_{t,t'} + c^{\text{t}}_{t'} + c^{\text{d}}_{t',t''}$ if trip t' is covered and cost $c^{\text{d}}_{t,t''}$ if not. According to (2.6), the costs for not covering the trip do not exceed the costs for covering. Therefore we have

$$\text{cost}(d_1) + \text{cost}(d_2) \leq 2 \cdot \text{cost}(d).$$

□

Corollary 1. *Consider the problem with $L_C \leq L_S$. Let S_1 be a feasible solution of* (MMILP)*. Then there exists a solution S_2 feasible also in* (CMILP)*.*

$$\text{val}(S_2) \leq 2 \cdot \text{val}(S_1)$$

4.3 Time-dependent Splitting

The developed formulation (CMILP) based on a customer-dependent splitting is not equivalent to the original formulation (MMILP). The goal now is to develop a splitting that is equivalent and create a heuristic based on this splitting. Therefore, it is necessary that trips of the same customer may be in different splittings. This leads to the following problem: When the partial instances are solved successively, we need a possibility to still guarantee the customer satisfaction for the entire problem. This has to be applied already in the first partial instance where a certain customer is concerned, although we do not have any knowledge about the trips of the same customer in the later solved partial instances.

4.3.1 Basic Idea

We define time-dependent splitting similar to [Kno16]. Based on this splitting, we adapt the model and describe the necessary cost estimation.

Splitting

We split the sets $\mathcal{T}$ and $\mathcal{V}$ according to their start times.

Definition 9 (Time-dependent Splitting)**.** Given points in time c_i for $i \in [n-1]$ with $c_i < c_{i+1}$ for $i \in [n-2]$. We define the splitting of $\mathcal{T}$ and $\mathcal{V}$ as follows:

$$\mathcal{T}_i := \begin{cases} \{t \in \mathcal{T} \mid z_t^{\text{start}} \leq c_1\} & \text{for } i = 1 \\ \{t \in \mathcal{T} \mid c_{i-1} < z_t^{\text{start}} \leq c_i\} & \text{for } i \in [n-1] \backslash \{1\} \\ \{t \in \mathcal{T} \mid c_{n-1} < z_t^{\text{start}}\} & \text{for } i = n \end{cases}$$

and

$$\mathcal{V}_i := \begin{cases} \{v \in \mathcal{V} \mid z_v \leq c_1\} & \text{for } i = 1 \\ \{v \in \mathcal{V} \mid c_{i-1} < z_v \leq c_i\} & \text{for } i \in [n-1] \backslash \{1\} \\ \{v \in \mathcal{V} \mid c_{n-1} < z_v\} & \text{for } i = n \end{cases}$$

We denote the formulation (SMILP) with a splitting according to Definition 9 as (TMILP).

Solving of the Partial Instances

Since the trips of the same customer may be in different splittings, we cannot easily guarantee the customer satisfaction only in just one partial instance. We have to put great effort in this issue. Let $\sigma \in S_n$ with $\sigma(n) = 1$ be the order in which the partial instances are solved. We first define the earliest solved partial instance, in which a trip of a customer arises, as follows:

$$\gamma : \mathcal{C} \to [n] \qquad \gamma(c) := \sigma\left(\min\left\{i \in [n] \mid \left((M \circ C)^{-1}(c) \cap \mathcal{T}_{\sigma(i)}\right) \neq \emptyset\right\}\right)$$

Depending on γ and $\{\mathcal{T}_1, \ldots \mathcal{T}_n\}$ we define a partition $\mathcal{C} = \{\mathcal{C}_1, \ldots \mathcal{C}_n\}$ as

$$\mathcal{C}_i := \{c \in \mathcal{C} \mid \gamma(c) = i\} \qquad \text{for } i \in [n]$$

Consider customer $c \in \mathcal{C}$. In partial instance $I_{\gamma(c)}$, a multimodal route $m \in C^{-1}(c)$ is chosen and this choice is definite. This means, in all subsequently solved partial instances, all trips $t \in m$ are fixed to be chosen in advance and all trips $t \in ((M \circ C)^{-1}(c) \backslash m)$ are fixed to be neglected.
In partial instance $\gamma(c)$ we have at least one trip of c. But there are also trips of c that are in other splittings. There are even multimodal routes with no trip in this splitting at all. These routes must not be neglected. Therefore, we need a method to choose the routes where all routes $m \in C^{-1}(c)$ are considered. Therefore, we try to estimate the costs of the routes in advance. The solving of the partial instances is again based on (SMILP$_i$).
For the cover constraints, we again introduce the decision variable $u_m \in \{0, 1\}$ for $m \in C^{-1}(\mathcal{C}_i)$. Notice that the definition of $\mathcal{C}_i$ is different to the definition in Section 4.2. In the customer constraints, only the customers in this splitting are considered. The route constraints are restricted to the trips that are actually in this splitting. The cover constraints read as follows:

$$\sum_{m \in C^{-1}(c)} u_m = 1 \qquad \text{for all } c \in \mathcal{C}_i \tag{4.29}$$

$$\sum_{s \in \mathrm{N}^{-}_{\overline{G}_i}(t)} x_{s,t} = u_m \qquad \text{for all } m \in C^{-1}(\mathcal{C}_i), t \in m \cap \mathcal{T}_i \tag{4.30}$$

$$u_m \in \{0, 1\} \qquad \text{for all } m \in C^{-1}(\mathcal{C}_i) \tag{4.31}$$

For the constraint (4.29) it is irrelevant, if the considered route has a trip in this splitting.
After solving the partial instance, all determined u_m are fixed for the later solved partial instances. The fixed route decisions from the previous partial instances have an impact on this instance, too.

Let $\bar{u}_m \in [0,1]$ be the fixed route choices from the previous instances. Define

$$\overline{\mathcal{C}}_i := \{c \in \mathcal{C} \mid \gamma(c) < \sigma(i)\}$$

as the customers that are already treated. Then, we introduce the constraint

$$\sum_{s \in \mathrm{N}^-_{\overline{G}_i}(t)} x_{s,t} = \bar{u}_m \qquad \text{for all } m \in C^{-1}\left(\overline{\mathcal{C}}_i\right), t \in m \cap \mathcal{T}_i \tag{4.32}$$

which ensures that the previous route choices are considered.

Cost Estimation

In order to choose a route in a partial instance, we have to estimate the costs for all routes of the same customer in all subsequently solved partial instances in advance. The entire cost for the problem consists of vehicle costs c^{v}, trip costs c^{t}, deadhead costs c^{d} and route costs c^{r}. While we can determine the trip costs and route costs easily for a route, the vehicle costs and trip costs strongly depend on the environment of the route and cannot be determined. We therefore focus on the trip and route costs and define the estimated route cost as follows:

$$C_1(m) := c^{\mathrm{r}}_m + \sum_{t \in m} c^{\mathrm{t}}_t \qquad \text{for } m \in \mathcal{M}$$

We use these costs in order to define the modified route costs

$$\hat{c}^{\mathrm{r}}_m := c^{\mathrm{r}}_m + \sum_{t \in m \setminus \mathcal{T}_i} c^{\mathrm{t}}_t \qquad \text{for } m \in \mathcal{M}$$

and add the following term to the objective function:

$$\sum_{m \in C^{-1}(\mathcal{C}_i)} u_m \hat{c}^{\mathrm{r}}_m$$

Remark 3. The trips in the considered splitting $t \in (m \cap \mathcal{T}_i)$ are not considered in $\hat{c}^{\mathrm{r}}_m$ since they are already part of the objective function. The other trips $t \in (m \setminus \mathcal{T}_i)$ are added to $\hat{c}^{\mathrm{r}}_m$, such that they have an impact on the choice of the routes.

Consider a trip t that is decided before this partial instance, i.e. $t \in (M \circ C)\left(\overline{\mathcal{C}}_i\right)$. Its trip costs c^{t}_t arise twice in the objective functions. Once in the partial instance $\gamma((M \circ C)(t))$ as part of $\hat{c}^{\mathrm{r}}_{M(t)}$ and once in partial instance I_i as c^{t}_t. But since in partial instance I_i the trip has fulfilled anyway, this cost is only an additional constant that does not influence the solution.

We denote the formulation (SMILP$_i$) with the constraints (4.29), (4.30), (4.32) and (4.31) and the new objective function including modified route costs as (TMILP$_i$) for $i \in [n]$.

4.3.2 Iterative Approach

We use the previously developed heuristic for an iterative approach. First, we compute an initial solution while we choose the routes according to cost function C_1. Based on this solution, we determine the actual costs of the routes. With this, we can estimate the contribution of a route to the objective function. We compare the estimated route cost to the actual route cost. If the actual route cost is considerably higher than the estimated route cost, it is likely that this route choice was bad. We identify the customer with the worst route choice and solve a subproblem, where we fix all route choices except for the considered customer. Regarding one customer after another, we can iteratively improve the solution.

Initial Solution

We determine a solution with the heuristic developed in Section 4.3.1 with a splitting according to Definition 9. Based on this solution $S = (\bar{x}, \bar{z}, \bar{e}, \bar{u})$, we determine

$$C_1(c) := C_1(m) \qquad \text{for } c \in \mathcal{C}, m \in C^{-1}(c) \text{ with } \bar{u}_m = 1$$

Finding Bad Route Choice

Given a solution of the problem, the subproblem is to find a customer with a bad route choice. This means, for this customer there is another route, such that the total cost is lower choosing this route. We can exchange these routes and compute a new solution considering the new route.
An initial idea is to compute the cost, one route in the solution contributes to the entire solution. Then, we can compare this to the cost, with which we estimated the route costs before. If the actual cost are considerably higher than the estimated cost, this customer is a candidate for exchanging routes. Since we cannot determine the contributing cost exactly, we try to estimate this cost.
Let $S = (\bar{x}, \bar{z}, \bar{e}, \bar{u})$ be a solution of (MMILP). In order to determine the contributing cost for route $m \in \mathcal{M}$, we define the following auxiliary costs for every trip $t \in \mathcal{T}$ of the solution:
Vehicle costs $c_t^{\mathrm{v}}(S)$: Let $v \in \mathcal{V}$ be the vehicle covering t and k_v the number of trips covered by v:

$$c_t^{\mathrm{v}}(S) := \frac{c^{\mathrm{v}}}{k_v}$$

Refueling costs $c_t^{\mathrm{refuel}}(S)$: Let $r \in \mathcal{R}$ be the next refuel station used after t and T_r all trips covered since the last station, let $\bar{z}_{s,r,s'} = 1$:

$$c_t^{\text{refuel}}(S) := \frac{f_t^{\text{t}}}{\sum_{t' \in T_r} f_{t'}^{\text{t}}} \left(c_{s,r}^{\text{d}} + c_{r,s'}^{\text{d}} - c_{s,s'}^{\text{d}} \right)$$

If the vehicle is not refueled after t, then $c_t^{\text{refuel}}(S) := 0$.
Deadhead costs $c_t^{\text{d}}(S)$: Let $s \in \mathcal{V} \cup \mathcal{T}, s' \in \mathcal{T}$ be the trips covered directly before and after t by vehicle v, i.e. $\bar{x}_{s,t} = \bar{x}_{t,s'} = 1$:

$$c_t^{\text{d}}(S) := \frac{1}{2} \left(c_{s,t}^{\text{d}} + c_{t,s'}^{\text{d}} \right)$$

If t is the last trip of the duty, i.e. $\bar{x}_{s,t} = \bar{x}_{t,d^{\text{e}}} = 1$, then $c_t^{\text{d}}(S) := \frac{1}{2} c_{s,t}^{\text{d}}$.
With these auxiliary costs we can define new route costs which describe the contribution of a multimodal route to the entire solution better:

Definition 10 (Improved Cost Estimation)**.** Let $S = (\bar{x}, \bar{z}, \bar{e}, \bar{u})$ be a solution of the (MMILP). With the auxiliary costs described before, we define the improved cost estimation for all multimodal routes $m \in \{m \in \mathcal{M} \mid \bar{u}_m = 1\}$:

$$C_2(S, m) := C_1(m) + \sum_{t \in m} \left(c_t^{\text{v}}(S) + c_t^{\text{refuel}}(S) + c_t^{\text{d}}(S) \right)$$

We further define

$$C_2(S, c) := C_2(S, m) \qquad \text{for } c \in \mathcal{C}, m \in C^{-1}(c) \text{ with } \bar{u}_m = 1$$

Now we can evaluate our previous estimation for the route contribution. If $C_2(S, c)$ is significantly higher than $C_1(S, c)$ then the probability is high that we made a bad route choice for customer $c \in \mathcal{C}$.
We therefore determine

$$c^* := \arg\max_{c \in \mathcal{C}} \frac{C_2(S, c)}{C_1(S, c)}$$

The probability is high that we made a bad route choice for customer c^*. Thus, we look at the route choice for c^* again.

Remark 4. For simplicity of notation, we assume that S is a solution of (MMILP). This is possible since the formulations (TMILP) and (MMILP) are equivalent.

Subproblem

Let $S = (\bar{x}, \bar{z}, \bar{e}, \bar{u})$ be a solution of (MMILP) and $c \in \mathcal{C}$ a candidate for a bad route choice. We define the following subproblem (HSP$_c$): Assume the schedule according

to S for the entire time without $[z_c^{\text{start}}, z_c^{\text{end}}]$ and all route choices for customers except c as fix. Determine an optimal schedule within these restrictions.
We define the splittings $\mathcal{T} = \{\mathcal{T}_1^c, \mathcal{T}_2^c, \mathcal{T}_3^c\}$ and $\mathcal{V} = \{\mathcal{V}_1^c, \mathcal{V}_2^c, \mathcal{V}_3^c\}$ by

$$\mathcal{T}_i^c := \begin{cases} \{t \in \mathcal{T}^c \mid z_t^{\text{start}} < z_c^{\text{start}}\} & \text{if } i = 1 \\ \left\{t \in \mathcal{T}^c \mid z_c^{\text{start}} \leq z_t^{\text{start}} \leq z_c^{\text{end}}\right\} & \text{if } i = 2 \\ \left\{t \in \mathcal{T}^c \mid z_c^{\text{end}} < z_t^{\text{start}}\right\} & \text{if } i = 3 \end{cases}$$

and

$$\mathcal{V}_i^c := \begin{cases} \{v \in \mathcal{V} \mid z_v < z_c^{\text{start}}\} & \text{if } i = 1 \\ \left\{v \in \mathcal{V} \mid z_c^{\text{start}} \leq z_v \leq z_c^{\text{end}}\right\} & \text{if } i = 2 \\ \left\{v \in \mathcal{V} \mid z_v < z_c^{\text{end}}\right\} & \text{if } i = 3 \end{cases}$$

We then define the start point set $\hat{\mathcal{V}}_2$ and the end point set $\hat{\mathcal{P}}_2$

$$\hat{\mathcal{V}}_2 := \{s \in \mathcal{T}_1^c \mid \bar{x}_{s,t} = 1 \text{ for } t \in (\mathcal{T}_2^c \cup \mathcal{T}_3^c \cup \{d^{\text{e}}\})\} \cup \mathcal{V}_1^c \cup \mathcal{V}_2^c$$
$$\hat{\mathcal{P}}_2 := \{t \in \mathcal{T}_3^c \mid \bar{x}_{s,t} = 1 \text{ for } s \in (\{d^{\text{s}}\} \cup \mathcal{T}_1^c \cup \mathcal{T}_2^c)\}$$

With these definitions, we can adapt the formulation (TMILP$_i$) for $i = 2$ to (HSP$_c$). The only modified constraints are the cover constraints (4.29), (4.30), (4.31) and (4.32). They are replaced by

$$\sum_{m \in C^{-1}(c)} u_m = 1 \tag{4.33}$$

$$\sum_{s \in \mathrm{N}^-_{\overline{G}_2}(t)} x_{s,t} = u_m \qquad \text{for all } m \in C^{-1}(c), t \in m \tag{4.34}$$

$$\sum_{s \in \mathrm{N}^-_{\overline{G}_2}(t)} x_{s,t} = \bar{u}_{M(t)} \qquad \text{for all } t \in \mathcal{T}_2^c \backslash (M \circ C)^{-1}(c) \tag{4.35}$$

$$u_m \in \{0, 1\} \qquad \text{for all } m \in M^{-1}(c) \tag{4.36}$$

We decide only the routes of customer c. Thus, we use the objective function of (SMILP$_i$) and add the following term:

$$\sum_{m \in C^{-1}(c)} u_m c_m^{\text{r}}$$

Creating an Improved Solution

With solving (HSP$_c$), we receive a new partial solution denoted as $\hat{S}_2^c$. Let S be the original entire solution. First, we transform S into three partial solutions $\{S_1^c, S_2^c, S_3^c\}$ according to the splitting $\mathcal{T} = \{\mathcal{T}_1^c, \mathcal{T}_2^c, \mathcal{T}_3^c\}$ and $\mathcal{V} = \{\mathcal{V}_1^c, \mathcal{V}_2^c, \mathcal{V}_3^c\}$. Then, we feasibly connect the partial solutions $\left\{S_1^c, \hat{S}_2^c, S_3^c\right\}$ to a new solution $\hat{S}$ according to the procedure described in Section 4.1.2.

e Note: This is ccording to the procedure in ...

The original partial solution S_2^c is a feasible solution of (HSP$_c$). Therefore, with this method we cannot get a worse entire solution than before.
After completing this step, we can apply this procedure to the customer with the second-highest ratio of $\frac{C_2(S,c)}{C_1(c)}$.

Remark 5. The customer extension L_{C} is not bounded explicitly like (4.27). But also here a small customer extension is beneficial due to the size of the (HSP$_c$).

4.3.3 Restricted Approach

We regard the special case in which each customer has trips in at most two subsequent splittings. This can be ensured if the customer extension is bounded by the splitting length. We try to exploit this special structure. For each customer, we basically distinguish between two cases: There are more trips of this customer in the splitting whose partial instance is solved first (Case 1) or there are more trips in the splitting whose partial instance is solved later (Case 2). In Case 1, the cost estimation for the routes is easy since most of the structure is already contained in the first processed partial instance. In Case 2, there is not much structure in the first processed partial instance, so the cost prediction will be imprecise. In order to prevent an imprecise cost estimation as in Case 2, we inspect the possibility of reversing a previous route choice in the later solved partial instance, if we find a better alternative there. In this section, we inspect the potential of cost saving for a belated trip deletion and develop a more flexible formulation in order to receive a better solution.

Lemma 3. *For $n \geq 3$, consider the problem with customer set $\mathcal{C}$ and split points c_i for $i \in [n-1]$ with $c_i < c_{i+1}$ for all $i \in [n-2]$. Let*

$$L_{\mathrm{C}} \leq L_{\mathrm{S}} \tag{4.27}$$

For every customer $c \in \mathcal{C}$, there is $i \in [n-1]$ such that

$$t \in (\mathcal{T}_i \cup \mathcal{T}_{i+1}) \qquad \text{for all } t \in (M \circ C)^{-1}(c) \tag{4.37}$$

this means, each customer is represented in at most two subsequent splittings.

Proof. For simplicity of notion, we state $c_0 := -\infty$ and $c_n := +\infty$. Consider customer $c \in \mathcal{C}$ and $i \in [n]$ such that $c_{i-1} \leq z_c^{\text{start}} < c_i$. For $i = n$ all trips of c are in splitting n. For $i < n$ we have

$$z_c^{\text{start}} \leq z_t^{\text{start}} \leq z_c^{\text{start}} + L_{\mathrm{C}} < c_i + L_{\mathrm{C}} \leq c_i + L_{\mathrm{S}} \leq c_{i+1} \quad \text{for all } t \in (M \circ C)^{-1}(c)$$

Thus we have

$$t \in (\mathcal{T}_i \,\dot{\cup}\, \mathcal{T}_{i+1}) \qquad\qquad \text{for all } t \in (M \circ C)^{-1}(c)$$

□

In the following considerations, we neglect the customer whose trip are in one splitting. These cover constraints are already ensured in the partial instance.
Consider partial instance $i \in [n]$, the customer set

$$\mathcal{C}_i^{\mathrm{R}} := \left\{ c \in \mathcal{C} \mid \gamma(c) \in \{i-1, i+1\} \wedge \left((M \circ C)^{-1}(c) \cap \mathcal{T}_i \right) \neq \emptyset \right\}$$

and the route set

$$\mathcal{M}_i^{\mathrm{R}} := \left\{ m \in \mathcal{M} \mid C(m) \in \mathcal{C}_i^{\mathrm{R}} \wedge m \subset \mathcal{T}_i \right\}$$

$\mathcal{C}_i^{\mathrm{R}}$ are all customers represented in $\mathcal{T}_i$ but initially treated in another partial instance, $\mathcal{M}_i^{\mathrm{R}}$ are all routes of these customers where all trips are in $\mathcal{T}_i$.
We regard the possibility to revise a previous route choice if we find a better alternative in partial instance i. For this, we think about the cost saving for subsequent trip deletion. As in Section 4.3.2, the cost function $C_1(m)$ is used for cost estimation.

Costs for Trip Replacement

We want to regard the possibility of deleting an already chosen route in partial instance I_i. We therefore consider customer $c \in \mathcal{C}_i^{\mathrm{R}}$, i. e. the customer has trips in an adjacent splitting and this partial instance is solved before. For this, we introduce the following notation.

Definition 11. Let $c \in \mathcal{C}_i^{\mathrm{R}}$ and let $S_{\gamma(c)} = (\bar{x}, \bar{z}, \bar{e}, \bar{u})$ be the partial solution of the previously solved partial instance, where the route of c has been chosen. Let $\bar{m}(c) \in C^{-1}(c)$ be the unique route with $\bar{u}_m = 1$
Let $s_1(t) \in \left(\left\{ d^{\mathrm{s}}_{\gamma(c)} \right\} \dot{\cup}\, \hat{\mathcal{V}}_{\gamma(c)} \,\dot{\cup}\, \mathcal{T}_{\gamma(c)} \right)$, $s_2(t) \in \left(\mathcal{T}_{\gamma(c)} \,\dot{\cup}\, \hat{\mathcal{P}}_{\gamma(c)} \,\dot{\cup} \left\{ d^{\mathrm{e}}_{\gamma(c)} \right\} \right)$ be the unique trips with $\bar{x}_{s_1,t} = \bar{x}_{t,s_2} = 1$ for all $t \in (\bar{m}(c) \backslash \mathcal{T}_i)$.
Here, $d^{\mathrm{s}}_{\gamma(c)}, d^{\mathrm{e}}_{\gamma(c)} \in \overline{V}_{\gamma(c)}$ are the respective source and sink node of the partial task graph $\overline{G}_{\gamma(c)}$.

The route $\bar{m}(c)$ denotes the route that is chosen for customer $c \in \mathcal{C}_i^{\mathrm{R}}$ by the partial solution, $s_1(t)$ and $s_2(t)$ are the trips that are directly before and after this trip in its respective duty.
By assuming $c^{\mathrm{d}}_{d^{\mathrm{s}},t} = c^{\mathrm{d}}_{t,d^{\mathrm{e}}} =: 0$ for all $t \in \mathcal{T}_{\gamma(c)}$, the cost saving for deleting a trip t in partial instance $I_{\gamma(c)}$ is

$$c^{\mathrm{d}}_{s_1(t),t} + c^{\mathrm{t}}_t + c^{\mathrm{d}}_{t,s_2(t)} - c^{\mathrm{d}}_{s_1(t),s_2(t)}$$

Adaption of the Model

In the following, we adapt the formulation (TMILP$_i$) in order to allow a belated route replacement. The restriction (4.27) is required for this formulation. It is also necessary that at least one adjacent partial instance is already solved. In this procedure, the partial instance is solved before, with the additional ability to replace already chosen routes under strong restrictions: If for customer $c \in \mathcal{C}$ a route has been chosen in a previously solved partial instance $I_{\gamma(c)}$ and there are routes whose trips are all in the partial trip set $\mathcal{T}_i$, then either one of these routes is chosen or the previous route decision is confirmed. The choice of another route is not possible since this would require an insertion of trips into an already existing partial solution. We call this formulation (RTMILP$_i$). The underlying partial task graph $\overline{G}_i$ is not modified.
We introduce new decision variables u^c for $c \in \mathcal{C}_i^{\mathrm{R}}$. They indicate whether the route choice for customer c is confirmed or not. If the route choice is not confirmed, adding of routes of the same customer is necessary. This is only possible, if all trips of this route are in $\mathcal{T}_i$. We therefore introduce decision variables u_m for all $m \in \mathcal{M}_i^{\mathrm{R}}$.
For every $c \in \mathcal{C}_i^{\mathrm{R}}$, either the previous choice must be confirmed or a new route is chosen. This is ensured by

$$u^c + \sum_{\substack{m \in \mathcal{M}_i^{\mathrm{R}} \\ C(m)=c}} u_m = 1 \qquad \text{for all } c \in \mathcal{C}_i^{\mathrm{R}} \tag{4.38}$$

We have $\overline{\mathcal{C}}_i = \mathcal{C}_i^{\mathrm{R}}$ since (4.27). The constraint (4.32) ensures the route decisions of the previous partial instances. It is replaced by

$$\sum_{s \in \mathrm{N}^-_{\overline{G}_i}(t)} x_{s,t} = u^c \qquad \text{for all } c \in \mathcal{C}_i^{\mathrm{R}}, t \in \bar{m}(c) \cap \mathcal{T}_i \tag{4.39}$$

$$\sum_{s \in \mathrm{N}^-_{\overline{G}_i}(t)} x_{s,t} = u_m \qquad \text{for all } t \in M^{-1}\left(\mathcal{M}_i^{\mathrm{R}}\right) \tag{4.40}$$

$$\sum_{s \in \mathrm{N}^-_{\overline{G}_i}(t)} x_{s,t} = 0 \qquad \text{for all } c \in \mathcal{C}_i^{\mathrm{R}}, t \in M^{-1}\left(C^{-1}(c) \setminus \left(\mathcal{M}_i^{\mathrm{R}} \cup \{\bar{m}(c)\}\right)\right) \cap \mathcal{T}_i \tag{4.41}$$

The constraint (4.39) ensures the route satisfaction for the previously decided route, (4.40) for all route completely in $\mathcal{T}_i$ and (4.41) for all other trips in $\mathcal{T}_i$. Note that $\mathcal{C}_i \cap \mathcal{C}_i^{\mathrm{R}} = \emptyset$. Hence, (4.29) and (4.30) are not influenced by them. We contract (4.30) and (4.40) to

$$\sum_{s \in \mathrm{N}^-_{\overline{G}_i}(t)} x_{s,t} = u_m \qquad \text{for all } m \in \mathcal{M}_i^{\mathrm{R}} \cup C^{-1}(\mathcal{C}_i), t \in m \cap \mathcal{T}_i \tag{4.42}$$

Finally, we replace (4.31) by

$$u_m \in \{0,1\} \qquad \text{for all } m \in C^{-1}(\mathcal{C}_i) \cup \mathcal{M}_i^{\mathrm{R}} \tag{4.43}$$

$$u^c \in \{0,1\} \qquad \text{for all } c \in \mathcal{C}_i^{\mathrm{R}} \tag{4.44}$$

Cost Function

We have to consider additional contributions to the cost function. If the route choice is not confirmed, the trips of $\bar{m}(c)$ are deleted and the route cost and the saved cost are subtracted from the cost function. We define these cost as

$$\hat{c}_c := c^{\mathrm{r}}_{\bar{m}(c)} + \sum_{t \in (\bar{m}(c) \setminus \mathcal{T}_i)} \left(c^{\mathrm{d}}_{s_1(t),t} + c^{\mathrm{t}}_t + c^{\mathrm{d}}_{t,s_2(t)} - c^{\mathrm{d}}_{s_1(t),s_2(t)} \right) \qquad \text{for } c \in \mathcal{C}_i^{\mathrm{R}}$$

Note, that the costs c^{t} and c^{d} belong to the partial instance $I_{\gamma(i)}$ and are not part of the considered partial task graph $\overline{G}_i$. The term $\hat{c}_c$ describes the cost saving for not confirming the route choice of c and is completely given in advance.
In order to determine the route costs for all $c \in \mathcal{C}_i^{\mathrm{R}}$, we add the following term to the objective function.

$$\sum_{m \in \mathcal{M}_i^{\mathrm{R}}} u_m c^{\mathrm{r}}_m - \sum_{c \in \mathcal{C}_i^{\mathrm{R}}} (1 - u^c)\, \hat{c}_c$$

In summary, the formulation (RTMILP$_i$) is given by (SMILP$_i$) with the constraints (4.29), (4.38), (4.39), (4.41), (4.42), (4.43) and (4.44) and the objective function

$$\left(\sum_{s \in \mathcal{T}_i \cup \hat{\mathcal{P}}_i} x_{d^{\mathrm{s}},s} - \sum_{s \in \hat{\mathcal{P}}_i} x_{s,d^{\mathrm{e}}} \right) c^{\mathrm{v}} + \sum_{m \in C^{-1}(\mathcal{C}_i)} u_m \hat{c}^{\mathrm{r}}_m + \sum_{m \in \mathcal{M}_i^{\mathrm{R}}} u_m c^{\mathrm{r}}_m + \sum_{c \in \mathcal{C}_i^{\mathrm{R}}} (u^c - 1)\, \hat{c}_c$$
$$+ \sum_{t \in \mathcal{T}_i \cup \hat{\mathcal{P}}_i} \sum_{s \in \mathrm{N}^-_{\overline{G}_i}(t) \setminus \{d^{\mathrm{s}}\}} \left[x_{s,t} \left(c^{\mathrm{d}}_{s,t} + c^{\mathrm{t}}_t \right) + \sum_{r \in \mathcal{R}_{s,t}} z_{s,r,t} \left(c^{\mathrm{d}}_{s,r} + c^{\mathrm{d}}_{r,t} - c^{\mathrm{d}}_{s,t} \right) \right] \tag{RTMILP$_i$}$$

4.3.4 Improvements

This is listing of comments regarding this section. It includes some small changes that can be made in the formulations in order to improve the performance, some new considerations and some small mistakes, where the developed methods work inaccurately but there is not an easy handling.

1. (4.27) does not hold in general: It is possible that each customer is represented in at most two splittings although (4.27) does not hold. If this is not the case, one

can possibly deviate some split points by a small value s. t. the condition holds. If only a small number of customers exceeds the splitting length, their exclusion from the formulations (HSP_c) or (RTMILP_i) still promises good results.

2. Create Preprocessing: One can create the split points via an own problem. The goal of this preprocessing is minimizing the customers represented in several splittings, constraints are a minimal and a maximal splitting length. If there are only few customers in more than one splitting, the solution behavior is improved.

3. Strategy for route choice in Section 4.3.2: In I_i, it is not beneficial to choose a route with $m \cap \mathcal{T}_i = \emptyset$. If such a route is the most suitable one, one can leave the choice open and choose in the next partial instance among all routes with $m \cap \mathcal{T}_i = \emptyset$. In I_i is is not necessary to choose the route because there are no trips to cover, thus it is not beneficial fix the route choice already there.

4. Strategy for route choice in Section 4.3.3: In $I_{\gamma(c)}$, it is beneficial to choose a route with $m \cap \mathcal{T}_{\gamma(c)} \neq \emptyset$, even is another choice, i. e. $m' \in \mathcal{M}_i^{\mathrm{R}}$ is more beneficial there. If m is a bad choice at the end, it will be reversed in I_i. If m is a good choice against expectation, it will be confirmed in I_i.

5. Deleting subsequent trips: If in (RTMILP_i) two subsequent trips of the same duty are deleted, the value for cost saving is wrong. For four subsequent trips t_1, t_2, t_3, t_4 where t_2, t_3 are deleted, the difference between the real cost saving and the computed cost saving is $c^{\mathrm{d}}_{t_1,t_4} + c^{\mathrm{d}}_{t_2,t_3} - c^{\mathrm{d}}_{t_1,t_3} - c^{\mathrm{d}}_{t_2,t_4}$. If there a subsequent trips of the same route in the same duty, the term can be adapted. An exact solution would be the introduction of a decision variable for each combination of route deletions.

Chapter 5

Optimal Approach

In this chapter, we develop a solution method in order to solve our problem to optimality. We expect this to require a very efficient algorithm and a lot of computation power since the problem is $\mathcal{NP}$-hard. The solution method should cope with multi-leg cover constraints as defined in Chapter 2. The approach is based on the underlying master theses which have already found methods to solve a simplified version of our problem. [Kai16] provide an optimal algorithm for the problem with single-leg cover constraints. This problem setting assumes that each customer has a set of alternative trips, where one of them has to be fulfilled each. We want our algorithm to produce a result in reasonable time, therefore we require already a very good solution as an initial solution. For receiving a good initial solution, we apply the Successive Heuristics as developed in Chapter 4.
In order to tackle the problem, we introduce a path flow formulation which is different to the arc flow formulation of Section 3.2. Since the entire solution consists of separate vehicle duties, we decompose the problems into these single duties. This concept is an application of Dantzig-Wolfe Decomposition. There are only a few constraints connecting these subproblems, namely the cover constraints. First we regard the LP relaxation of this problem and solve this via column generation. The resulting subproblem for each duty is a shortest path problem with resource constraints (SPPRC), which is also $\mathcal{NP}$-hard. For solving this subproblem, we have both a heuristic and an exact algorithm. In order to receive a total solution, we apply branch-and-price. We provide and discuss some branching strategies used for this procedure.
Most of the procedure is already developed by [Kai16]. We show the crucial results for the algorithm and discuss our adaptions in the path flow formulation, the algorithm solving the subproblems and the branch-and-price procedure. This adaptions make the optimal approach also cope with multi-leg cover constraints.

5.1 Path Flow Formulation

We apply Dantzig-Wolfe Decomposition in order to create a path flow formulation of our problem. This is advantageous since the size of the arc flow formulation grows very fast with increasing problem size. We give only a short outline on the general procedure and then show the application of our problem.

5.1.1 Dantzig-Wolfe Decomposition

Dantzig-Wolfe Decomposition can be used in order to deal with large mixed-integer linear programs. It breaks the problem into smaller subproblems if the structure is suitable. This is the case if a large subset of the variables can be partitioned in a way such that the sets of occurring variables are disjoint for most of the constraints. The structure of the matrix for such a linear program looks as follows:

$$
\left(\begin{array}{cccccccccccc}
\star & \cdots & \star & \star & \cdots & \star & & & & \star & \cdots & \star \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & & \cdots & & \vdots & \ddots & \vdots \\
\star & \cdots & \star & \star & \cdots & \star & & & & \star & \cdots & \star \\
\hline
\star & \cdots & \star & 0 & \cdots & 0 & & & & 0 & \cdots & 0 \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & & \cdots & & \vdots & \ddots & \vdots \\
\star & \cdots & \star & 0 & \cdots & 0 & & & & 0 & \cdots & 0 \\
0 & \cdots & 0 & \star & \cdots & \star & & & & & & \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & & \ddots & & & \vdots & \\
0 & \cdots & 0 & \star & \cdots & \star & & & & & & \\
 & & & & & & & & & 0 & & 0 \\
 & \vdots & & & \ddots & & & \ddots & & \vdots & \ddots & \vdots \\
 & & & & & & & & & 0 & & 0 \\
0 & \cdots & 0 & & & & 0 & \cdots & 0 & \star & \cdots & \star \\
\vdots & \ddots & \vdots & & \cdots & & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & 0 & & & & 0 & \cdots & 0 & \star & \cdots & \star
\end{array}\right)
$$

The subproblems emerge by considering only the constraints of a single set of this partition. The other constraints concerning the whole variable set are called linking constraints as they link the respective subproblems. The master problem considers the objective function in connection with the linking constraints. We then apply column generation for each of the subproblems separately. Starting with only a small set of feasible solutions, we successively generate further feasible solutions and include them to the master problem. Each feasible solution represents a column of the matrix representing the linear program. In the master problem, the actual formulation of the subproblems is not needed. Thus, we can extract the subproblems and solve them with specialized algorithms if they have an appropriate structure.
The column generation method is only able to solve linear program. Thus, we have to restate the integrality afterwards. How this is done is discussed later.

5.1.2 Application of the Decomposition

In the original problem formulation, we regard only a single set of variables which model the entire flow of the vehicles. For the arc flow formulation, a single variable set is advantageous as the corresponding task graph stays small. In contrast to this, we extend the variable set in order to define smaller subproblems.

Identification of the Subproblems

Consider a solution of the (MMILP). This solution can be decomposed in a set of separate vehicle duties. For each of these duties, the time and fuel restrictions can be applied individually. The only requirements that do not occur in the respective duties individually are the cover constraints. They guarantee that for each customer exactly one route is fulfilled and for each route, if it is fulfilled, each of its trips are fulfilled. Therefore the duty of each vehicle is a natural choice for the subproblem. We introduce (x^v, z^v, e^v) for $v \in \mathcal{V}$ as the specific variables for each vehicle. With this we define the set of feasible vehicle duties as X_v for $v \in \mathcal{V}$.

$$
\begin{aligned}
X_v := \Big\{ & (x, z, e) \in \{0,1\}^A \times \{0,1\}^{(A \cap (\mathcal{V} \cup \mathcal{T})^2) \times \mathcal{R}} \times [0,1]^{\mathcal{V} \cup \mathcal{T}} \,\Big| \\
& \sum_{t \in \mathrm{N}_G^-(s)} x_{t,s} = \sum_{t \in \mathrm{N}_G^+(s)} x_{s,t} && \text{for all } s \in V \setminus \{d^{\mathrm{s}}, d^{\mathrm{e}}\} && (3.3) \\
& \sum_{s \in \mathrm{N}_G^-(t)} x_{s,v} = 1 && && (5.1) \\
& \sum_{s \in \mathrm{N}_G^-(t)} x_{s,t} = 0 && \text{for all } t \in \mathcal{V} \setminus \{v\} && (5.2) \\
& \sum_{r \in \mathcal{R}_{s,t}} z_{s,r,t} \leq x_{s,t} && \text{for all } t \in \mathcal{T}, s \in \mathrm{N}_G^-(t) && (3.5) \\
& e_s \leq f_s^0 && \text{for all } s \in \mathcal{V} && (3.6) \\
& 0 \leq e_s - \sum_{r \in \mathcal{R}_{s,t}} z_{s,r,t} f_{s,r}^{\mathrm{d}} && \text{for all } t \in \mathcal{T}, s \in \mathrm{N}_G^-(t) && (3.7) \\
& e_t \leq 1 - f_t^{\mathrm{t}} - \sum_{r \in \mathcal{R}_{s,t}} z_{s,r,t} f_{r,t}^{\mathrm{d}} && \text{for all } t \in \mathcal{T}, s \in \mathrm{N}_G^-(t) && (3.8) \\
& e_t \leq e_s - x_{s,t} \left(f_{s,t}^{\mathrm{d}} + f_t^{\mathrm{t}} \right) - \sum_{r \in \mathcal{R}_{s,t}} z_{s,r,t} \left(f_{s,r}^{\mathrm{d}} + f_r^{\mathrm{t}} + f_{r,t}^{\mathrm{d}} - f_{s,t}^{\mathrm{d}} \right) + (1 - x_{s,t}) \\
& && \text{for all } t \in \mathcal{T}, s \in \mathrm{N}_G^-(t) && (3.9) \\
\Big\} &
\end{aligned}
$$

Constraints (5.1) and (5.2) ensure that exactly vehicle v is used in this formulation.

We denote the set of feasible duties for any vehicle by $X := \bigcup_{v \in \mathcal{V}} X_v$. Any feasible solution of (MMILP) can be decomposed into vehicle duties. This is guaranteed by (3.4) which forces the duties of the vehicles to be disjoint with respect to the trips. The only variables that are not considered in X_v are the route variables u_m which can be determined by the arc variables $x_{s,t}$. The objective function is additive with respect to the decomposition except for the route cost which we then consider explicitly. We write the cost for a configuration (x^v, z^v, e^v) as $g\,(x^v, z^v, e^v)$.

The only constraints that are not ensured in X_v are the cover constraints (3.1) and (3.2). These are the linking constraints for the various subproblems. In summary, we can rewrite (MMILP) as

$$
\begin{aligned}
\min \quad & \sum_{v \in \mathcal{V}} g\,(x^v, z^v, e^v) + \sum_{m \in \mathcal{M}} u_m c^{\mathrm{r}}_m & & \\
\text{s.t.} \quad & \sum_{m \in C^{-1}(c)} u_m = 1 & \text{for all } c \in \mathcal{C} \qquad & (3.1) \\
& \sum_{v \in \mathcal{V}} \sum_{s \in \mathrm{N}^-_G(t)} x^v_{s,t} = u_m & \text{for all } m \in \mathcal{M}, t \in m \qquad & (5.3) \\
& (x^v, z^v, e^v) \in X_v & \text{for all } v \in \mathcal{V} & \\
& u_m \in \{0,1\}^{\mathcal{M}} & &
\end{aligned}
$$

Reduction of the Master Problem

Because of the introduction of variables for each vehicle, the resulting problem size is very large. For maintaining the master problem, not all information of X_v are needed. In order to fulfill (5.3) we need only $\sum_{s \in \mathrm{N}^-_G(t)} x_{s,t}$, which is the set of trips served by a specific vehicle. Therefore, we define the linear mapping

$$
\psi : X \to \{0,1\}^{\mathcal{T}} \qquad (x, z, e) \mapsto \left(\sum_{s \in \mathrm{N}^-_G(t)} x_{s,t} \right)_{t \in \mathcal{T}}
$$

The dimension of the codomain of ψ is much smaller than the dimension of the domain.

We can rewrite (MMILP) by using $y^v := \psi\left(x^v, z^v, e^v\right)$:

$$\begin{aligned}
\min \quad & \sum_{v \in \mathcal{V}} \min g\left(\psi^{-1}\left(y^v\right) \cap X_v\right) + \sum_{m \in \mathcal{M}} u_m c^{\mathrm{r}}_m \\
\text{s.t.} \quad & \sum_{m \in C^{-1}(c)} u_m = 1 && \text{for all } c \in \mathcal{C} \qquad (3.1)\\
& \sum_{v \in \mathcal{V}} y^v_t = u_m && \text{for all } m \in \mathcal{M}, t \in m \\
& y^v \in \psi\left(X_v\right) && \text{for all } v \in \mathcal{V} \\
& u_m \in \{0,1\} && \text{for all } m \in \mathcal{M}
\end{aligned}$$

The mapping ψ is not injective in general. Thus, there is more than one feasible duty that serves exactly the trips of y^v. These duties can have different cost. We therefore use the minimal resulting cost

$$\min g\left(\psi^{-1}\left(y^v\right) \cap X_v\right) = \min\left\{g\left(x^v\right) | x^v \in X_v, \psi\left(x^v\right) = y^v\right\}$$

This is the smallest cost of a vehicle duty that serves exactly the trips as indicated by the incidence vector y^v. We do not have to determine these costs now. As we will see later, the costs are a byproduct of solving the subproblems.

Column Generation

We apply column generation to our problem. For every $v \in \mathcal{V}$, let $\mathcal{I}_v$ be an index set for the finitely many points in $\psi\left(X_v\right)$ and let the columns of $Y^v \in \mathbb{R}^{\mathcal{T} \times \mathcal{I}_v}$ be exactly those points. Let $G^v \in \mathbb{R}^{1 \times \mathcal{I}}$ be the respective values of $\min g\left(\psi^{-1}(\cdot) \cap X_v\right)$. Then we can reformulate the master problem as

$$\begin{aligned}
\min \quad & \sum_{v \in \mathcal{V}} G^v \lambda^v + \sum_{m \in \mathcal{M}} u_m c^{\mathrm{r}}_m && \qquad \text{(IMP)}\\
\text{s.t.} \quad & \sum_{v \in \mathcal{V}} Y^v_{t,\cdot} \lambda^v = u_m && \text{for all } m \in \mathcal{M}, t \in m \\
& \sum_{m \in C^{-1}(c)} u_m = 1 && \text{for all } c \in \mathcal{C} \\
& \sum_{i \in \mathcal{I}_v} \lambda^v_i = 1 && \text{for all } v \in \mathcal{V} \\
& \lambda^v \in \{0,1\}^{\mathcal{I}_v} && \text{for all } v \in \mathcal{V} \\
& u_m \in \{0,1\} && \text{for all } m \in \mathcal{M}
\end{aligned}$$

Then we regard the LP-relaxation of (IMP) by dropping the integrality constraints:

$$\begin{aligned}
\min \quad & \sum_{v\in\mathcal{V}} G^v\lambda^v + \sum_{m\in\mathcal{M}} u_m c^{\mathrm{r}}_m & & \text{(LMP)}\\
\text{s.t.} \quad & \sum_{v\in\mathcal{V}} Y^v_{t,\cdot}\lambda^v = u_m & \text{for all } m\in\mathcal{M}, t\in m &\\
& \sum_{m\in C^{-1}(c)} u_m = 1 & \text{for all } c\in\mathcal{C} &\\
& \sum_{i\in\mathcal{I}_v} \lambda^v_i = 1 & \text{for all } v\in\mathcal{V} &\\
& \lambda^v \in \mathbb{R}^{\mathcal{I}_v}_{\geq 0} & \text{for all } v\in\mathcal{V} &\\
& u_m \geq 0 & \text{for all } m\in\mathcal{M} &
\end{aligned}$$

As next step, we reduce the size of the problem by considering only subsets $\mathcal{J}_v \subset \mathcal{I}_v$ of the feasible solutions for all $v \in \mathcal{V}$ and formulate the relaxed restricted master problem:

$$\begin{aligned}
\min \quad & \sum_{v\in\mathcal{V}} G^v_{\mathcal{J}_v}\lambda^v + \sum_{m\in\mathcal{M}} u_m c^{\mathrm{r}}_m & & \text{(LRMP)}\\
\text{s.t.} \quad & \sum_{v\in\mathcal{V}} Y^v_{t,\mathcal{J}_v}\lambda^v = u_m & \text{for all } m\in\mathcal{M}, t\in m &\\
& \sum_{m\in C^{-1}(c)} u_m = 1 & \text{for all } c\in\mathcal{C} &\\
& \sum_{i\in\mathcal{J}_v} \lambda^v_i = 1 & \text{for all } v\in\mathcal{V} &\\
& \lambda^v \in \mathbb{R}^{\mathcal{J}_v}_{\geq 0} & \text{for all } v\in\mathcal{V} &\\
& u_m \geq 0 & \text{for all } m\in\mathcal{M} &
\end{aligned}$$

Finally, we regard the dual relaxed restricted master problem. For this, we introduce dual variables $\gamma \in \mathbb{R}^{\mathcal{T}}$, $\mu \in \mathbb{R}^{\mathcal{V}}$ and $\eta \in \mathbb{R}^{\mathcal{C}}$. The dual problem is:

$$\begin{aligned}
\max \quad & \sum_{c\in\mathcal{C}} \eta_c + \sum_{v\in\mathcal{V}} \mu_v & & \text{(DLRMP)}\\
\text{s.t.} \quad & \sum_{t\in\mathcal{T}} Y^v_{t,i}\gamma_t + \mu_v \leq G^v_i & \text{for all } v\in\mathcal{V}, i\in\mathcal{J}_v & \quad (5.4)\\
& \eta_{C(m)} - \sum_{t\in m} \gamma_t \leq c^{\mathrm{r}}_m & \text{for all } m\in\mathcal{M} & \quad (5.5)\\
& \gamma \in \mathbb{R}^{\mathcal{T}} & &\\
& \mu \in \mathbb{R}^{\mathcal{V}} & &\\
& \eta \in \mathbb{R}^{\mathcal{C}} & &
\end{aligned}$$

5.1.3 Solving the Relaxed Master Problem

The size of the index set $\mathcal{I}_v$ of all feasible solutions of X_v can be exponential in the input size. Therefore, the formulation (IMP) is hard, even the relaxed version (LMP) is hard. In order to tackle the problem, we first consider a small subset $\mathcal{J}_v \subset \mathcal{I}_v$ of the index set. We solve the problem (LRMP) where only duties from the restricted set are allowed. Since $\mathcal{J}_v$ is small, it is easier to solve the problem. Originating from this solution, we iteratively enlarge $\mathcal{J}_v$ and solve (LRMP) until the solution is an optimal solution of (LMP). For this method arise the following questions.

1. Does this procedure come up with an optimal solution in finitely many steps?

2. How do we find columns to add?

3. How do we check for optimality in (LMP)?

If we iteratively add columns to $\mathcal{J}_v$, we finally have $\mathcal{J}_v = \mathcal{I}_v$ after a finite number of steps since $\mathcal{I}_v$ is a finite set. When this is reached, the problems (LRMP) and (LMP) are equivalent and thus the solution is optimal. Obviously, this behavior is not desirable as we do not want to solve the unrestricted problem. Thus, we hope to receive an optimal solution earlier.
We can check for optimality and find columns to add by using the dual problems of the restricted and the unrestricted problem. Consider a solution $(\lambda^v)_{v\in\mathcal{V}}$ of (LRMP) and its corresponding dual solution $(\gamma^*, \mu^*, \eta^*)$ which is feasible in (DLRMP). We want to check whether $(\lambda^v)_{v\in\mathcal{V}}$ is an optimal solution of the unrestricted problem (LMP). Due to strong duality, this is the case if and only if $(\gamma^*, \mu^*, \eta^*)$ is feasible in the unrestricted dual problem (DLMP).
We therefore consider the constraints of the dual problems. (5.5) are equivalent in both formulations. The constraints (5.4) read as follows in the unrestricted case

$$\sum_{t\in\mathcal{T}} Y^v_{t,i}\gamma_t + \mu_v \leq G^v_i \qquad \text{for all } v \in \mathcal{V}, i \in \mathcal{I}_v \tag{5.6}$$

Since $(\gamma^*, \mu^*, \eta^*)$ is a solution of the restricted problem, (5.6) is fulfilled for all $i \in \mathcal{J}_v$. It remains to check $\mathcal{I}_v \backslash \mathcal{J}_v$ which leads to the subproblem

$$\text{Find } i \in \mathcal{I}_v \backslash \mathcal{J}_v \qquad \text{s.t.} \qquad \sum_{t\in\mathcal{T}} Y^v_{t,i}\gamma^*_t + \mu^*_v > G^v_i \qquad \text{for } v \in \mathcal{V} \tag{5.7}$$

Identification of the Subproblem

Recall the definitions $G_i^v = \min g\left(\psi^{-1}\left(Y_i^v\right) \cap X_v\right)$ and $Y_i^v = \psi(x,z,e)$ for the respective $(x,z,e) \in X_v$. Using this, we can rewrite the subproblem to

$$\begin{aligned} \min \quad & g\left(x,z,e\right) - \sum_{t \in \mathcal{T}} \sum_{s \in \mathrm{N}_G^-(t)} x_{s,t}\gamma_t^* - \mu_v^* \\ \text{s.t.} \quad & (x,z,e) \in X_v \end{aligned} \tag{SP$_v$}$$

Actually, the term $g(x,z,e)$ would be $\min g\left(\psi^{-1}\left(\psi(x,z,e)\right) \cap X_v\right)$. The following result shows that this distinction is not necessary as this is done implicitly be solving the subproblem.

Lemma 4. *For $v \in \mathcal{V}$, an optimal solution $(x^*,z^*,e^*) \in X_v$ to the subproblem (SP$_v$) fulfills*

$$g\left(x^*,z^*,e^*\right) = \min g\left(\psi^{-1}\left(\psi\left(x^*,z^*,e^*\right)\right) \cap X_v\right).$$

In other words, the duty (x^*,z^*,e^*) has the smallest possible cost under all duties which serve the same set of trips.

This lemma is proven by [Kai16, pp. 42-43] and holds for our case, too.
How the subproblem (SP$_v$) is solved, is shown in Section 5.2. As mentioned before, the cost G_i^v are also determined by solving the subproblem. We receive a solution (x,z,e) of (SP$_v$) for all $v \in \mathcal{V}$ and simultaneously the cost $g(x,z,e)$. If we then add the corresponding duty to $\mathcal{J}_v$, we can easily use the determined cost for G^v.

Updating the Index Set

The value of a duty as determined in the subproblem is called reduced cost. As long as there exists a violated constraint in the dual problem, there exists a column with negative reduced cost. This is used for deciding if a duty is added to the index set. First, we solve (DLRMP) and receive a solution (γ^*,μ^*,η^*). With this we solve (SP$_v$) for all $v \in \mathcal{V}$ and receive solutions (x^v,z^v,e^v). We know that all of these duties with negative reduced cost correspond to a violated constraint in (DLMP). Thus we consider these duties in the next step. For all $v \in \mathcal{V}$ with $\text{value}\left(x^v,z^v,e^v\right) < 0$ we update the index set

$$\mathcal{J}_v \leftarrow \mathcal{J}_v \cup \{i\} \qquad Y_{\cdot,i}^v \leftarrow \psi\left(x^v,z^v,e^v\right) \qquad G_i^v \leftarrow g\left(x^v,z^v,e^v\right)$$

If $\text{value}\left(x^v,z^v,e^v\right) \geq 0$ for all $v \in \mathcal{V}$, then the dual solution (γ^*,μ^*,η^*) is feasible in (DLMP) and the corresponding primal solution $(\lambda^v)_{v \in \mathcal{V}}$ is an optimal solution of the relaxed master problem (LMP).

Initial Solution

For starting the column generation method, an initial index set is required for $\mathcal{J}_v$ for all $v \in \mathcal{V}$. The index sets have to be feasible, i. e. each occurring duty is feasible and there is a solution satisfying the cover constraints (3.1) and (3.2) using only duties out of $\bigcup_{v\in\mathcal{V}} \mathcal{J}_v$. Otherwise the restricted problem is infeasible and its dual problem is unbounded. Then we do not receive a solution (x^*, z^*, e^*) of (DLRMP) with which we define the subproblems. As an initial solution we use a heuristical solution of the problem, as we have developed in Chapter 4.
Note that this procedure only provides a solution of the LP-relaxation of the master problem. In Section 5.3 we show how we receive a solution of (IMP).

5.2 Solving of the Subproblems

In every step of the master problem, we solve the subproblem (SP_v) for each vehicle $v \in \mathcal{V}$. This subproblem is equivalent to the Shortest Path Problem with Resource Constraints (SPPRC). A vehicle duty is expressed as d^{s}-d^{e}-path whose first vertex is the respective vehicle v. The main resource is the fuel state of the vehicle, where refuel stations have negative fuel consumption. The goal is to find a feasible path with negative reduced cost which is then the new duty. Besides fuel, in the algorithm are used additional resources which describe what multimodal routes are served in this duty.

5.2.1 Shortest Path Problem with Resource Constraints

In this section, we summarize the crucial results in order to show a label-setting algorithm which solves the (SPPRC) optimally. The respective definitions and the algorithm are shown in detail in [Kai16] and [ID05]. The (SPPRC) is a generalization of the Shortest Path Problem and is $\mathcal{NP}$-hard, as shown in [HZ80, p.307]. The problem is given by a graph, a set of resources and a relation on every arc that specifies the change of resources along its way.

Definition 12 (Graph with resource constraints)**.** We call $H := (V_H, A_H, \sqsubseteq, I, \mathrm{REF})$ a graph with resource constraints for a set of resources $\mathcal{U}$ if

1. (V_H, A_H) is a directed graph with vertex set V_H and arc set A_H.

2. $\sqsubseteq \in \{\leq, =, \geq\}^{\mathcal{U}}$ is a vector of resource relations and is called the resource dominance relation. For two resources $r, \tilde{r} \in \mathbb{R}^{\mathcal{U}}$, we write $r \sqsubseteq \tilde{r}$ if $r_u \sqsubseteq_u \tilde{r}_u$ for all $u \in \mathcal{U}$ and say that $\tilde{r}$ dominates r. The subset of maximal vectors of a set

$R \subseteq \mathbb{R}^{\mathcal{U}}$ with respect to $\sqsubseteq$ is denoted by

$$\max_{\sqsubseteq} R := \{r \in R \mid \forall \tilde{r} \in R : r \sqsubseteq \tilde{r} \Rightarrow r = \tilde{r}\}$$

The closed cone of resource vectors less than or equal to zero with respect to $\sqsubseteq$ is denoted by

$$\mathbb{R}^{\mathcal{U}}_{\sqsubseteq 0} := \left\{r \in \mathbb{R}^{\mathcal{U}} \mid r \sqsubseteq 0_{\mathcal{U}}\right\}$$

3. $I \subseteq \mathbb{R}^{\mathcal{U}}$ is the Cartesian product of closed intervals of $\mathbb{R}$. The projection onto a single resource $u \in \mathcal{U}$ denoted by $\Pi_u(I)$ is called its resource window. If $\Pi_u(I) = \mathbb{R}$ for some resource $u \in \mathcal{U}$ it is called unrestricted.

4. $\mathrm{REF} = (\mathrm{REF}_{u,w})_{(v,w)\in A_H}$ is a vector of binary relations $\mathrm{REF}_{v,w} \subseteq I \times I$ for all $(v, w) \in A_H$ such that the set of vectors related to some $r^v \in I$

$$\mathrm{REF}_{v,w}\left(r^v\right) := \{r^w \in I \mid (r^v, r^w) \in \mathrm{REF}_{v,w}\}$$

is closed, has a finite set of maximal vectors $\max_{\sqsubseteq} \mathrm{REF}_{v,w}\left(r^v\right)$ and fulfills

$$\forall r^w, \tilde{r}^w \in I, r^w \sqsubseteq \tilde{r}^w : \tilde{r}^w \in \mathrm{REF}_{v,w}\left(r^v\right) \Rightarrow r^w \in \mathrm{REF}_{v,w}\left(r^v\right).$$

$\mathrm{REF}_{v,w}$ is called the resource extension function with respect to $\sqsubseteq$ on the arc $(v, w) \in A_H$.

The resource vectors are assigned to the vertices of the graph. They describe the absolute amount of available resources at that vertex.
The resource dominance relation is a partial order on $\mathbb{R}^{\mathcal{U}}$. If two resource vectors are comparable, then the dominating vector is always preferable to the other. If it is desirable to have a high quantity of a resource, the resource relation is set to $\leq$, e. g. for the fuel resource. Otherwise it is set to $\geq$, e. g. for the modeling cost resource. If no general relation holds, it is set to $=$. The resource extension function models the change of the resource vectors along the arcs. It relates a resource vector to all possible outcomes when traveling along this arc.

Definition 13 (Monotone resource extension function)**.** A resource extension function $\mathrm{REF}_{v,w} \subseteq I \times I$ with respect to $\sqsubseteq$ is called monotone if

$$\forall r^v, \tilde{r}^v \in I, r^v \sqsubseteq \tilde{r}^v : \mathrm{REF}_{v,w}\left(r^v\right) \subseteq \mathrm{REF}_{v,w}\left(\tilde{r}^v\right)$$

holds.

The monotonicity is important for the consistency. If a resource vector is dominated by another, then there are not more possible outcomes than for the dominating one. After introducing the graph, we define resource-feasible paths on this graph.

Definition 14 (Resource-feasible path)**.** Let $H = (V_H, A_H, \sqsubseteq, I, \mathrm{REF})$ be a graph with resource constraints. A path $P := (v_0, \ldots, v_n)$ of length $n \in \mathbb{N}_0$ in H is called resource-feasible if

$$\exists r^{v_i} \in I, i \in \{0, \ldots, n\} : (r^{v_{i-1}}, r^{v_i}) \in \mathrm{REF}_{v_{i-1},v_i} \, \forall i \in \{1, \ldots, n\}$$

holds. We say $(r^v)_{v \in P}$ witnesses resource-feasibility of P.

The witnessing resource vectors $(r^v)_{v \in \mathcal{P}}$ are the resources along this path, e. g. the fuel state of the respective trips of a vehicle duty.

Contraction and Inversion

We define the actions "contraction of an arc" and "inversion of a graph" in order to apply them to our problem.

Definition 15 (Contraction)**.** $H = (V_H, A_H, \sqsubseteq, I, \mathrm{REF})$ be a graph with resource constraints, $(v, w) \in A_H$ be the only arc leaving some vertex $v \in V_H$ and $\mathrm{REF}_{v,w}$ be monotone.

1. For $(u, v) \in A_H$, the concatenation of resource extension functions is defined as

$$\begin{aligned} \mathrm{REF}_{v,w} \circ \mathrm{REF}_{u,v} := \{\, (r^u, r^w) \in I \times I \mid \exists r^v \in I : \\ (r^u, r^v) \in \mathrm{REF}_{u,v} \wedge (r^v, r^w) \in \mathrm{REF}_{v,w} \} \end{aligned}$$

2. The graph with resource constraints $\widehat{H} := \left(V_{\widehat{H}}, A_{\widehat{H}}, \sqsubseteq, I, \widehat{\mathrm{REF}}\right)$ which results from H by contracting the arc (v, w) is defined by the vertex set $V_{\widehat{H}} := V_H \backslash \{v\}$, the arc set

$$A_{\widehat{H}} := \left(A_H \cap V_{\widehat{H}}^2\right) \cup \{(u, w) \mid (u, v) \in A_H\},$$

and the resource extension function $(\mathrm{REF}_a)_{a \in A_{\widehat{H}}}$, where

$$\widehat{\mathrm{REF}}_a := \begin{cases} \mathrm{REF}_{u,w} \cup (\mathrm{REF}_{v,w} \circ \mathrm{REF}_{u,v}) & \text{if } \exists u \in V_{\widehat{H}} : a = (u, w) \\ \mathrm{REF}_a & \text{otherwise} \end{cases}$$

for all $a \in A_H$. $\mathrm{REF}_{u,w}$ and $\mathrm{REF}_{u,v}$ are considered to be the empty relation $\emptyset$ if $(u, w) \notin A_H$ or $(u, v) \notin A_H$, respectively.

In [Kai16, p. 79] is proven that for a resource-feasible path P in the original graph H there is a resource-feasible path $\widehat{P}$ in the contracted graph $\widehat{H}$ and vice versa such that P and $\widehat{P}$ cover the same vertices of $V_H \backslash \{v\}$. We need contraction since we have resources on both the vertices and the arcs in our problem.

Definition 16 (Inversion). 1. A resource extension function $\mathrm{REF}_{v,w}$ with respect to $\sqsubseteq$ is called invertible if the inverted relation

$$\mathrm{REF}_{v,w}^{-1} := \{(r^w, r^v) \mid (r^v, r^w) \in \mathrm{REF}_{v,w}\}$$

is a resource extension function with respect to the inverted dominance relation $\sqsupseteq$. $\mathrm{REF}_{v,w}^{-1}$ is called the inversion of $\mathrm{REF}_{v,w}$.

2. Let $H := (V_H, A_H, \sqsubseteq, I, \mathrm{REF})$ be a graph with resource constraints and invertible resource extension functions.

   The inversion of H is defined to be the graph

$$H^{-1} := \left(V_H, A_H^{-1}, \sqsupseteq, I, \mathrm{REF}^{-1}\right)$$

   with inverted arc set $A_H^{-1} := \left\{(w,v) \in V_H^2 \mid (v,w) \in A_H\right\}$ and inverted resource extension functions $\mathrm{REF}^{-1} := \left(\mathrm{REF}_{v,w}^{-1}\right)_{(w,v)\in A_H^{-1}}$.

Theorem 5 (Feasibility-conservation of inversions). *Let $H := (V_H, A_H, \sqsubseteq, I, \mathrm{REF})$ be a graph with resource constraints. Further, let all the resource extension functions of* REF *be invertible.*
Then a path $P := (v_0, \dots, v_n)$ of length $n \in \mathbb{N}_0$ in H is resource-feasible with witnessing resource vectors $(r^v)_{v\in P}$ if and only if $P^{-1} := (v_n, \dots, v_0)$ is a resource-feasible path in the inverted graph H^{-1} with witnessing resource vectors $(r^v)_{v\in P^{-1}}$.

This theorem is already proven in [Kai16, p. 83]. We need inversions in order to improve the behavior of the algorithm. Using inversions we can apply the algorithm once for all subproblems and do not have to solve each subproblem separately.

Label-Setting Algorithm

We solve the (SPPRC) via a label-setting algorithm. Previously in this section, we have added resources to the graph in order to restrict the set of feasible paths. Since we do not have a cost function, finding an optimal path is not straight-forward. We do not have a shortest path but multiple resource-feasible paths. We use the resource dominance relation $\sqsubseteq$ to compare different paths. If a path is dominated by another, it is not preferable and therefore not considered in the solution. If two paths are not comparable, we cannot decide which one is more preferable. This leads to the concept of Pareto-optimality.

Definition 17 (Pareto-optimal paths). Let $H := (V_H, A_H, \sqsubseteq, I, \mathrm{REF})$ be a graph with resource constraints. Let P be a resource-feasible v-w-path in H.

P is called Pareto-optimal if there exist witnesses $(r^u)_{u \in P}$ for the resource-feasibility of P such that for every resource-feasible v-w-path Q in H with witnessing resource vectors $(\tilde{r}^u)_{u \in Q}$ fulfilling $\tilde{r}^v = r^v$

$$r^w \sqsubseteq \tilde{r}^w \Rightarrow r^w = \tilde{r}^w$$

holds. We say that the resource vectors $(r^u)_{u \in P}$ witness Pareto-optimality of P.

In general, there can be exponentially many paths in a graph with respect to its size. It is further possible that the witnessing resource vectors are not comparable. Hence, there can be an exponential number of Pareto-optimal paths in a graph. This fact gives a feeling why (SPPRC) is $\mathcal{NP}$-hard.
The idea of the algorithm is to start with a trivial path, consisting of one single vertex. We then extend the paths while we maintain resource-feasibility and Pareto-optimality for all paths. Finally, we receive Pareto-optimal paths from the starting vertex to all vertices. The algorithm works on the concept of Dynamic Programming.

Algorithm 2: Label-setting algorithm for acyclic graphs with resource constraints

Input: graph with resource constraints $H := (V_H, A_H, \sqsubseteq, I, \mathrm{REF})$, topological sorting $v_0, \ldots, v_n$ of V_H and initial resource vector r^{v_0}
Output: shortest path tree rooted at (v_0, r^{v_0}) encoded by δ

$\mathcal{P}_{v_0} \leftarrow \{r^{v_0}\}$;
$\delta(v_0, r^{v_0}) \leftarrow \emptyset$;
foreach $i \in \{1, \ldots, n\}$ **do** $\mathcal{P}_{v_i} \leftarrow \emptyset$;
foreach $i = 0, \ldots, n$ **do**
    foreach $r^{v_i} \in \mathcal{P}_{v_i}$ **do**
        foreach $w \in \mathrm{N}_H^+(v_i)$ **do**
            $\mathcal{P} \leftarrow \max_{\sqsubseteq} \mathrm{REF}_{v_i,w}(r^{v_i})$;
            foreach $r^w \in \mathcal{P}$ **do** $\delta(w, r^w) \leftarrow (v_i, r^{v_i})$;
            $\mathcal{P}_w \leftarrow \mathcal{P}_w \cup \mathcal{P}$;
            $\mathcal{P}_w \leftarrow \max_{\sqsubseteq} \mathcal{P}_w$;
        end
    end
end
return δ

We have a graph with resource constraints, a topological sorting of the vertices and an initial resource vector as input. The graph has to be acyclic. A topological sorting $\{v_0, \ldots, v_n\}$ means that there are no $i < j$ with $(v_j, v_i) \in A_H$. Start with vertex v_0 and initial resource vector r^{v_0}, we treat the vertices successively in topological order. For each vertex $v \in V_H$ and for each computed Pareto-optimal v_0-v-path, we try to extend the path feasibly by a single vertex. If a path is found, we add a label to the extending vertex and update the mapping δ in order to identify the origin of the extension. At the end, we receive the mapping δ which identifies all resource-feasible Pareto-optimal v_0-v-paths for each vertex $v \in V_H$.

5.2.2 Strengthening Inequalities

Before we apply the previously stated algorithm to the subproblem, we want to insert additional valid inequalities. These are not necessary but may improve the column generation process. In the original problem, we have the linking constraints (3.1) and (3.2) which ensure that for every customer exactly one route is fulfilled. These constraints cannot be moved into the subproblems since the customers can be satisfied by different vehicles. It is even likely that two trips of the same route are fulfilled by different vehicles.
Nevertheless, we can identify duties that are infeasible with respect to the cover constraints. This is the case, if a vehicle fulfills two trips that belong to the same costumer but not to the same route. If this duty is part of the overall solution, it is not possible that (3.1) and (3.2) are fulfilled simultaneously. Therefore, we want to prevent such a duty from being added to the column set.
In order to use the inequalities, we have to introduce decision variables $u_m \in \{0, 1\}$ for $m \in \mathcal{M}$. We insert the following inequalities to (SP_v) for each $v \in \mathcal{V}$. They are not necessary since they are implied by the cover constraints of the master problem, but they strengthen formulation of the subproblem.

$$\sum_{m \in C^{-1}(c)} u_m \leq 1 \qquad \text{for all } c \in \mathcal{C} \tag{5.8}$$

$$\sum_{s \in \mathrm{N}_G^-(t)} x_{s,t} \leq u_m \qquad \text{for all } m \in \mathcal{M}, t \in m \tag{5.9}$$

These constraints ensure that for every customer at most one route can be fulfilled. Note that (5.8) is also valid with equality. Adding these constraints possibly makes the subproblems harder to solve, but improves the behavior in the master problem since there are no duties added that are infeasible from the very beginning. If the addition is beneficial for the overall process, is not known in advance.

5.2.3 Determination of the Resources

The only resource that we have used so far is the fuel resource. We use the index *fuel* in our resource vector. The fuel is in the interval $[0, 1]$ for the fuel level where 0 means that the vehicle has no fuel and 1 that the vehicle is completely fueled. A higher fuel level is preferable, hence the resource relation for fuel is $\leq$.

Since we have no objective function in this problem, we model the reduced cost as an additional resource. We introduce the index *redcost* in order to keep track of the reduced cost of a duty. The reduced cost is unrestricted and the reduced cost is minimized, thus we set the resource window to $\mathbb{R}$ and the resource relation to $\geq$.
An additional resource is the number of trips that a vehicle fulfills. We call it the length of a duty and use the index *length*. This resource is not necessary for the subproblem but advantageous for the branch-and-bound procedure as we see in Section 5.3. The length of a duty lies in the interval $[0, |\mathcal{T}|]$. Comparing two duties with different lengths, it is not clear which of them is preferable. Thus the resource relation is $=$.
In order to ensure the constraints (5.8) and (5.9), we use a resource for every multimodal route. We use the respective $m \in \mathcal{M}$ as index for the route resource. This resource indicates how many trips of this route can still be fulfilled within this duty. Initially, all trips of a route can be fulfilled, thus the resource window is $[0, |m|]$ for $m \in \mathcal{M}$. Similar to the duty length it is not possible to compare different route resources, therefore the resource relation is $=$.
Altogether, we consider resources $\mathcal{U} := \{\text{redcost}, \text{fuel}, \text{length}\} \cup \mathcal{M}$ in the resource window

$$I := \mathbb{R} \times [0,1] \times [0, |\mathcal{T}|] \times \bigotimes_{m \in \mathcal{M}} [0, |m|] \subseteq \mathbb{R}^{\mathcal{U}}$$

A resource vector $r \in I$ consists of the reduced cost $r_{\text{redcost}} \in \mathbb{R}$, the fuel level $r_{\text{fuel}} \in [0,1]$, the duty length $r_{\text{length}} \in [0, |\mathcal{T}|]$ and the route resources $r_m \in [0, |m|]$ for $m \in \mathcal{M}$ in this order. The resource relation vector is given by $\sqsubseteq := (\geq, \leq, =, =, \ldots, =)$.
These resources coincide in large parts with the formulation in [Kai16]. They use resources for each customer instead of route resources in order to ensure the single-leg cover constraints. This is the only adaption in this section and the next section.

5.2.4 Determination of the Resource Extension Function

In this section, we determine the resource extension function such that we receive a formulation equivalent to (SP_v). We first define a more complex graph with simple resource extension functions. Then we contract the graph in order to shift the complexity from the graph into the functions.

Extended Task Graph with Split Vertices

In the problem as defined in Section 5.2.1, changes of the resource vectors are only defined on the arcs. This means, only resources occurring on the arcs are considered. Since we have also a trip cost and fuel consumption at the vertices of the task graph, we modify the graph in order to deal with vertex resources. Therefore, we create a

graph by splitting the vertices, based on the extended task graph $G = (V, A)$ according to Definition 2.
Define $\widetilde{G} := \left(\widetilde{V}, \widetilde{A}\right)$ with vertex set

$$\widetilde{V} := \{d^{\mathrm{s}}, d^{\mathrm{e}}\} \cup \left\{s^-, s^+ \mid s \in V \setminus \{d^{\mathrm{s}}, d^{\mathrm{e}}\}\right\}$$

and arc set

$$\begin{aligned}\widetilde{A} := &\left\{(d^{\mathrm{s}}, s^-) \mid (d^{\mathrm{s}}, s) \in A\right\} \cup \left\{\left(s^+, d^{\mathrm{e}}\right) \mid (s, d^{\mathrm{e}}) \in A\right\} \\ &\cup \left\{\left(s^+, t^-\right) \mid (s, t) \in A\right\} \cup \left\{\left(s^-, s^+\right) \mid s \in V \setminus \{d^{\mathrm{s}}, d^{\mathrm{e}}\}\right\}\end{aligned}$$

We use $\sqsubseteq$, I as described in Section 5.2.3 and $\widetilde{V}$, $\widetilde{A}$ as described before. The values γ_t for $t \in \mathcal{T}$ and μ_v for the specific $v \in \mathcal{V}$ come from the dual solution of (DLRMP), from which the subproblem results. With this, we define the resource extension function $\widetilde{\mathrm{REF}}$.
In the arc between the split vertices of a vehicle $v \in \mathcal{V}$ occur the cost $-\mu_v$ and the fuel consumption $(1 - f_v^0)$ as the initial fuel is f_v^0. Thus we define the resource extension function for $v \in \mathcal{V}$ as

$$\widetilde{\mathrm{REF}}_{v^-,v^+}(c, e, l, b) := \left(\left(\begin{array}{c} c - \mu_v \\ e - (1 - f_v^0) \\ l \\ b \end{array}\right) + \mathbb{R}^{\mathcal{U}}_{\sqsubseteq 0}\right) \cap I$$

for $(c, e, l, b) \in I$. In refuel points we have fuel consumption f_r^{t}, thus we define the resource extension function for each refuel point r as

$$\widetilde{\mathrm{REF}}_{r^-,r^+}(c, e, l, b) := \left(\left(\begin{array}{c} c \\ e - f_r^{\mathrm{t}} \\ l \\ b \end{array}\right) + \mathbb{R}^{\mathcal{U}}_{\sqsubseteq 0}\right) \cap I$$

for $(c, e, l, b) \in I$.
In a trip $t \in \mathcal{T}$ occur the cost $c_t^{\mathrm{t}} - \gamma_t$, the fuel consumption is f_t^{t} and the length of the duty increases by 1. If trip t is fulfilled in this duty, no other trip of the same costumer must be fulfilled, unless from the same route. We therefore define the following auxiliary functions

$$\begin{aligned} M^{\mathrm{C}}(t) &= \{m \in \mathcal{M} \mid C(m) \neq (C \circ M)(t)\} \\ M^{\mathrm{R}}(t) &= \{m \in \mathcal{M} \mid C(m) = (C \circ M)(t) \wedge m \neq M(t)\} \end{aligned}$$

Note that $M^{\mathrm{C}}(t), \mathcal{M}^{\mathrm{R}}(t) \subseteq \mathcal{M}$ are subsets and $M(t) \in \mathcal{M}$ is an element of the route set. We define the resource extension function for $t \in \mathcal{T}$ as

Xme Note: l+1

$$\widetilde{\mathrm{REF}}_{t^-,t^+}(c,e,l,b) := \left(\begin{pmatrix} c + c^{\mathrm{t}}_t - \gamma_t \\ e - f^{\mathrm{t}}_t \\ l - 1 \\ b_{M^{\mathrm{C}}(t)} \\ 0_{M^{\mathrm{R}}(t)} \\ b_{M(t)} - 1 \end{pmatrix} + \mathbb{R}^{\mathcal{U}}_{\sqsubseteq 0} \right) \cap I \tag{5.10}$$

for $(c,e,l,b) \in I$ where $\mathbb{1}_{\overline{M}(t)}$ is the unit vector of the coordinate $\overline{M}(t)$. Between two trips, we have the cost and the fuel consumption for the deadhead trip and for $(s,t) \in A \cap \mathcal{T}^2$

$$\widetilde{\mathrm{REF}}_{s^+,t^-}(c,e,l,b) := \left(\begin{pmatrix} c + c^{\mathrm{d}}_{s,t} \\ e - f^{\mathrm{d}}_{s,t} \\ l \\ b \end{pmatrix} + \mathbb{R}^{\mathcal{U}}_{\sqsubseteq 0} \right) \cap I$$

for $(c,e,l,b) \in I$. Between a vehicle and a trip we additionally have to consider the fixed vehicle cost c^{v}, thus the resource extension function is for $(s,t) \in A \cap \mathcal{V} \times \mathcal{T}$ defined by

$$\widetilde{\mathrm{REF}}_{s^+,t^-}(c,e,l,b) := \left(\begin{pmatrix} c + c^{\mathrm{v}} + c^{\mathrm{d}}_{s,t} \\ e - f^{\mathrm{d}}_{s,t} \\ l \\ b \end{pmatrix} + \mathbb{R}^{\mathcal{U}}_{\sqsubseteq 0} \right) \cap I$$

for $(c,e,l,b) \in I$. For all arcs incident with d^{s} or d^{e}, the resource extension function is the function corresponding to the identity, i. e. $\left((c,e,l,b) + \mathbb{R}^{\mathcal{U}}_{\sqsubseteq 0}\right) \cap I$ for $(c,e,l,b) \in I$.

Extended Task Graph

We transform the developed task graph $\widetilde{G} = \left(\widetilde{V}, \widetilde{A}, \sqsubseteq, I, \widetilde{\mathrm{REF}}\right)$ by contracting the arcs $\left(t^-, t^+\right)$ for all $t \in V \setminus \{d^{\mathrm{s}}, d^{\mathrm{e}}\}$. Then we identify t^+ with t for all $t \in V \setminus \{d^{\mathrm{s}}, d^{\mathrm{e}}\}$ and receive the extended task graph $G = (V, A, \sqsubseteq, I, \mathrm{REF})$ with

$$\mathrm{REF}_{s,t} = \widetilde{\mathrm{REF}}_{t^-,t^+} \circ \widetilde{\mathrm{REF}}_{s^+,t^-}$$

for all $s,t \in V \setminus \{d^{\mathrm{s}}, d^{\mathrm{e}}\}$. For arcs starting from a refuel point or a trip and leading to a trip, i. e. $(s,t) \in A, s \notin \mathcal{V}, t \in \mathcal{T}$ we have

$$\mathrm{REF}_{s,t}(c,e,l,b) := \left(\begin{pmatrix} c + c^{\mathrm{d}}_{s,t} + c^{\mathrm{t}}_{s,t} - \gamma_t \\ e - f^{\mathrm{d}}_{s,t} - f^{\mathrm{d}}_{s,t} \\ l-1 \\ b - \mathbb{1}_{\overline{M}(t)} \end{pmatrix} + \mathbb{R}^{\mathcal{U}}_{\sqsubseteq 0} \right) \cap I$$

FiXme Note: adapt b-vector

since $f^{\mathrm{d}}_{s,t} \geq 0$ and $f^{\mathrm{t}}_{t} \geq 0$. The resources are independent from each other.
For arcs starting at a trip and leading to a refuel point, i. e. $(s,r),(s,t) \in A, s \in \mathcal{T}, r \in \mathcal{R}_{s,t}$ we have

$$\mathrm{REF}_{s,r}(c,e,l,b) := \begin{cases} \left(\begin{pmatrix} c + c^{\mathrm{d}}_{s,r} \\ e - f^{\mathrm{d}}_{s,r} - f^{\mathrm{d}}_{s,r} \\ l \\ b \end{pmatrix} + \mathbb{R}^{\mathcal{U}}_{\sqsubseteq 0} \right) \cap I & \text{if } e \geq f^{\mathrm{d}}_{s,r} \\ \emptyset & \text{otherwise} \end{cases}$$

For both cases, we can extend the evaluation to $s \in \mathcal{V}$ by adding c^{v} to the reduced cost resource. For arcs indicent with d^{s} or d^{e} we have $\mathrm{REF}_{d^{\mathrm{s}},s} = \widetilde{\mathrm{REF}}_{s^-,s^+}$ and $\mathrm{REF}_{s,d^{\mathrm{e}}} = \widetilde{\mathrm{REF}}_{s^+,d^{\mathrm{e}}}$.

Task Graph

We transform the extended task graph $G = (V, A, \sqsubseteq, I, \mathrm{REF})$ by contracting the arcs $(r,t) \in A$ for all $s,t \in \mathcal{V} \cup \mathcal{T}, s \prec t$ and $r \in \mathcal{R}_{s,t}$. This yields the task graph $\widehat{G} = \left(\widehat{V}, \widehat{A}, \sqsubseteq, I, \widehat{\mathrm{REF}}\right)$ which builds on the task graph $\widehat{G} = \left(\widehat{V}, \widehat{A}\right)$ from Definition 2.
For every arc $(s,t) \in \widehat{A}$ with $s \notin \mathcal{V}, t \in \mathcal{T}$ and $r \in \mathcal{R}_{s,t}$, we determine the resource vectors $\left(\mathrm{REF}_{r,t} \circ \mathrm{REF}_{s,r}\right)(c,e,l,b)$ as

$$\left.\begin{array}{ll} \left(\begin{pmatrix} c + c^{\mathrm{d}}_{s,r} + c^{\mathrm{d}}_{r,t} + c^{\mathrm{t}}_{t} - \gamma_t \\ \min\left(e - f^{\mathrm{d}}_{s,r} - f^{\mathrm{t}}_{r}, 1\right) - f^{\mathrm{d}}_{r,t} - f^{\mathrm{t}}_{t} \\ l-1 \\ b - \mathbb{1}_{\overline{M}(t)} \end{pmatrix} + \mathbb{R}^{\mathcal{U}}_{\sqsubseteq 0} \right) \cap I & \text{if } e \geq f^{\mathrm{d}}_{s,r} \\ \emptyset & \text{otherwise} \end{array}\right.$$

for $(c,e,l,b) \in I$. For $s \in \mathcal{V}$, we again add c^{v} to the reduced cost resource. The resource extension function is then according to Definition 15 given by

$$\widehat{\mathrm{REF}}_{s,t}(c,e,l,b) = \mathrm{REF}_{s,t}(c,e,l,b) \cup \bigcup_{r\in\mathcal{R}_{s,t}} \left(\mathrm{REF}_{r,t} \circ \mathrm{REF}_{s,r}\right)(c,e,l,b)$$

$$= \left[\left(\begin{pmatrix} c + c^{\mathrm{d}}_{s,t} + c^{\mathrm{t}}_{t} - \gamma_t \\ e - f^{\mathrm{d}}_{s,t} - f^{\mathrm{t}}_{t} \\ l-1 \\ b - \mathbb{1}_{\overline{M}(t)} \end{pmatrix} + \mathbb{R}^{\mathcal{U}}_{\sqsubseteq 0}\right) \cap I\right]$$

$$\cup \left[\left(\left\{\begin{pmatrix} c + c^{\mathrm{d}}_{s,r} + c^{\mathrm{d}}_{r,t} + c^{\mathrm{t}}_{t} - \gamma_t \\ \min\left(e - f^{\mathrm{d}}_{s,r} - f^{\mathrm{t}}_{r}, 1\right) - f^{\mathrm{d}}_{r,t} - f^{\mathrm{t}}_{t} \\ l-1 \\ b - \mathbb{1}_{\overline{M}(t)} \end{pmatrix} \mid r \in \mathcal{R}_{s,t}, e \geq f^{\mathrm{d}}_{s,r} \right\} + \mathbb{R}^{\mathcal{U}}_{\sqsubseteq 0}\right) \cap I\right]$$

for $s \notin \mathcal{V}$ and with an additional c^{v} in the first component of all vectors for $s \in \mathcal{V}$.

5.2.5 An Exact Algorithm for Solving the Subproblems

After specification of the resources and the resource extension function, we present an algorithm that solves the subproblem for every vehicle optimally.

Model Equivalence

First, we prove that the previously developed problem formulation is equivalent to the subproblem and that the values for the reduced cost coincide.

Theorem 6 (Correspondence to MMILP formulation)**.** *For every resource-feasible d^{s}-d^{e}-path $P := (d^{\mathrm{s}}, v, t_1, \ldots, t_n, d^{\mathrm{e}})$ with $n \in \mathbb{N}_0$ in the graph $\widehat{G} = \left(\widehat{V}, \widehat{A}, \sqsubseteq, I, \widehat{\mathrm{REF}}\right)$ with witnessing resource vectors $(r^v)_{v\in P}$, there is a feasible solution $(x,z,e,u) \in X_v$ satisfying (5.8) and (5.9) such that $g^v(x,z,e,u) \leq r^{d^{\mathrm{e}}}_{\mathrm{redcost}} - r^{d^{\mathrm{s}}}_{\mathrm{redcost}}$.*
Inversely, for every feasible solution $(x,z,e,u) \in X_v$ to some subproblem $v \in \mathcal{V}$ that satisfies (5.8) and (5.9), there is a resource-feasible d^{s}-d^{e}-path $P := (d^{\mathrm{s}}, v, t_1, \ldots, t_n, d^{\mathrm{e}})$, $n \in \mathbb{N}_0$ in $\widehat{G} = \left(\widehat{V}, \widehat{A}, \sqsubseteq, I, \widehat{\mathrm{REF}}\right)$ with witnessing resource vectors $(r^v)_{v\in P}$ such that the equation $g^v(x,z,e) = r^{d^{\mathrm{e}}}_{\mathrm{redcost}} - r^{d^{\mathrm{s}}}_{\mathrm{redcost}}$ holds.

This theorem is proven by [Kai16, pp. 96-99] for a slightly modified problem. The strengthening inequality is adapted to single-leg cover constraints and the route resource is a customer resource there. This has also an impact on the resource extension function. We do not present the complete proof, but only the parts that are different here. This concerns mainly the decision variable u_m and the route resources r_m.

Proof. We show that we can create a solution of (MMILP) out of a resource-feasible path and vice versa.

"⇒": Resource-feasible path to feasible solution

Let $P := (d^{\mathrm{s}}, t_0, t_1, \ldots, t_n, d^{\mathrm{e}}), n \in \mathbb{N}_0$ be a resource-feasible d^{s}-d^{e}-path in $\widehat{G} = \left(\widehat{V}, \widehat{A}, \sqsubseteq, I, \widehat{\mathrm{REF}}\right)$ with witnessing resource vectors $(r^s)_{s \in P}$. Construct the following solution (x, z, e, u). For $a \in \widehat{A}$, set x_a to 1 if the path P uses the arc a and 0 otherwise. Set $e_s := r^s_{\mathrm{fuel}}$ if $s \in P$ and $e_s := 0$ otherwise. For all $i \in [n]$ and $r \in \mathcal{R}_{t_{i-1},t_i}$, set $z_{t_{i-1},r,t_i} := 1$ if $r^{t_{i-1}} \in \left(\widehat{\mathrm{REF}}_{r,t_i} \circ \widehat{\mathrm{REF}}_{t_{i-1},r}\right)\left(r^{t_{i-1}}\right)$ and $z_{t_{i-1},r,t_i} := 0$ otherwise. If this holds true for more than one $r \in \mathcal{R}_{t_{i-1},t_i}$, change z_{t_{i-1},r,t_i} to 0 for all such r but one. For all other arcs $a \in A$, set $z_a := 0$. For all $m \in \mathcal{M}$ set $u_m := 1$ if there is an $i \in [n]$ such that $M(t_i) = m$ and $u_m := 0$ otherwise. The claim is that this $(x, z, e, u) \in X_v$ holds, i. e. (x, z, e, u) is a feasible solution to the subproblem of vehicle $v := t_0$.
The flow conservation and the fuel constraints hold as proven by [Kai16]. This applied directly since the respective parts of X_v and $\widehat{\mathrm{REF}}$ have not been modified. The claim concerning the reduced cost is also proven there.
Due to (3.3) and (3.10) we have $\sum_{s \in \mathrm{N}^-_{\widehat{G}}(t)} x_{s,t} \in \{0, 1\}$ for all $t \in \mathcal{T}$. From the previous definition follows that

$$\sum_{s \in \mathrm{N}^-_{\widehat{G}}(t)} x_{s,t} = 1 \qquad \Rightarrow \qquad t \in P \qquad \Rightarrow \qquad u_{M(t)} = 1$$

for all $t \in \mathcal{T}$ and therefore holds (5.9). For (5.8) we assume by contradiction that there is a $m \in \mathcal{M}$ with $\sum_{m \in C^{-1}(c)} u_m > 1$. Then there are $i, j \in [n], i < j$ such that $t_i, t_j \in P$ and $M(t_i) \neq M(t_j), (M \circ C)(t_i) = (M \circ C)(t_j)$. Since the definition of $\widehat{\mathrm{REF}}$ and $t_j \in M^{\mathrm{R}}(t_i)$ we have

$$r^{t_i}_{M(t_j)} = 0 \qquad \Rightarrow \qquad r^{t_{j-1}}_{M(t_j)} = 0 \qquad \Rightarrow \qquad r^{t_j}_{M(t_j)} = r^{t_{j-1}}_{M(t_j)} - 1 \notin I_{M(t_j)}$$

This leads to contradiction to the resource-feasibility of P and therefore holds (5.8).

"⇐": Feasible solution to resource-feasible path

Let $(x, z, e, u) \in X_v$ for some $v \in \mathcal{V}$. Set $t_0 := v$. By the definition of X_v, $x_{d^{\mathrm{s}},t_0} = 1$ holds. Based on the flow conservation (3.3) and construction of the task graph $\widehat{G}$, the existence of exactly one $t_1 \in \mathcal{T} \cup \{d^{\mathrm{e}}\}$ with $x_{t_0,t_1} = 1$ follows. This step can be repeated $n \in \mathbb{N}_0$ times for finding $t_i \in \mathcal{T} \cup \{d^{\mathrm{e}}\}, i = 2, \ldots, n+1$ until $t_{n+1} = d^{\mathrm{e}}$ is reached. This defines a path $P := (d^{\mathrm{s}}, t_0, \ldots, t_n, d^{\mathrm{e}})$ in the task graph. Set the resource vectors $(r^s)_{s \in P}$ along this path as follows. For all vertices $t_k, k \in \{0, \ldots, n\}$, set the reduced cost to

$$r^{t_k}_{\text{redcost}} := c^{\text{v}} + \sum_{i=1}^{k} \left[x_{t_{i-1},t_i} \left(c^{\text{d}}_{t_{i-1},t_i} + c^{\text{t}}_{t_i} - \gamma_{t_i} \right) + \sum_{r \in \mathcal{R}_{t_{i-1},t_i}} z_{t_{i-1},r,t_i} \left(c^{\text{d}}_{t_{i-1},r} + c^{\text{d}}_{r,t_i} - c^{\text{d}}_{t_{i-1},t_i} \right) \right] - \mu_v$$

the fuel level to $r^{t_k}_{\text{fuel}} := e_{t_k}$, the length to $r^{t_k}_{\text{length}} := k$ and the route resources to

$$r^{t_k}_m = \begin{cases} 0 & \text{if } \sum\limits_{\substack{i \in [k]: \\ m \in M^{\text{R}}(t_i)}} \sum\limits_{s \in \mathrm{N}^-_{\widehat{G}}} x_{s,t} > 0 \\ |m| - \sum\limits_{\substack{i \in [k]: \\ m = M(t_i)}} \sum\limits_{s \in \mathrm{N}^-_{\widehat{G}}} x_{s,t} & \text{otherwise} \end{cases}$$

for $m \in \mathcal{M}$. The route resource lies in I since $|m|$ is chosen large enough. The resource vectors at the source and sink are set to $r^{d^{\text{s}}} := \left(0, 1, 0, (|m|)_{m \in \mathcal{M}}\right)$ and $r^{d^{\text{e}}} := r^{t_n}$, respectively. We claim that these resource vectors $(r^s)_{s \in P}$ witness resource-feasibility of the path P in the task graph $\widehat{G}$.
The resource-feasibility for the resources *fuel*, *redcost* and *length* are already proven by [Kai16]. This is applied directly since the respective parts of X_v and $\widehat{REF}$ have not been modified.
For all $m \in \mathcal{M}$, we have $r^{t_0}_m = |m| \in I_m$ and therefore resource-feasibility. For $k \in [n]$, we distinguish the following cases for the resource-feasibility of $r^{t_k}_m$:

$$\left. \begin{array}{lcl} m \in M^{\text{R}}(t_k) & \Rightarrow & r^{t_k}_m = 0 \\ m \in M^{\text{C}}(t_k) & \Rightarrow & r^{t_k}_m = r^{t_{k-1}}_m \\ m = M(t_k) & \Rightarrow & r^{t_k}_m = r^{t^{k-1}_m} - 1, r^{t_k}_m \geq 0 \end{array} \right\} \Rightarrow r^{t_k}_m \in \widehat{\text{REF}}_{t_{k-1},t_k} \left(r^{t_{k-1}}_m \right)$$

We always have $r^{t_k}_m \geq 0$ since the initial value has been chosen large enough. Therefore, the above defined resource vector witnesses resource-feasibility for all $m \in \mathcal{M}$. This concludes the proof.

□

Inversion of the Graph

We have seen in Theorem 6 that the subproblem (SP_v) can be written as a Shortest Path Problem with Resource Constraints. Therefore, we can apply Algorithm 2 in order to solve the subproblem optimally. Since there is a subproblem for every vehicle

$v \in \mathcal{V}$, it is not advantageous to apply the algorithm for each subproblem separately. In order to improve this behavior we can exploit the symmetry of the problem. Remember, that the only differences of the subproblems are caused by the the dual variables (γ_t, μ_v) from the dual solution of (DLRMP). Thus, the subproblems vary only in the fact which vehicle is included. As we search d^{s}-d^{e}-paths in the subproblem, only the second vertex of such a path is affected.
From the results of Theorem 5 and Theorem 6 we can see that (SPPRC) given on the inverted task graph $\widehat{G}^{-1}$ is equivalent to the one given by $\widehat{G}$. As Algorithm 2 yields Pareto-optimal paths from the starting vertex to each vertex in the graph, it is profitable to apply the algorithm to $\widehat{G}^{-1}$ with starting vertex d^{e}. Thus, we receive Pareto-optimal d^{e}-v-paths for all $v \in \mathcal{V}$ in the inverted graph and we have to execute the algorithm only once. We can then extend the respective d^{e}-v-paths to d^{e}-d^{s}-paths and receive solutions for all the subproblems.
The following lemma shows that it is possible to invert the problem. It is a condition for Theorem 5 that the resource extension function is invertible.

Lemma 5 (Invertibility of $\widehat{\mathrm{REF}}$)**.** *The resource extension function* $\widehat{\mathrm{REF}}$ *is invertible.*

Proof. We will only prove that the resource extension function $\widetilde{\mathrm{REF}}$ restricted to the route constraints and to the split vertices of trips is invertible. [Kai16] has already shown the invertibility of the resource extension function for the simplified problem setting and that invertibility is maintained if arcs are contracted. Since the parts of $\widetilde{\mathrm{REF}}$ are independent from each other, the invertibility of $\widehat{\mathrm{REF}}$ follows directly.
For readability, we write the restriction of $\widetilde{\mathrm{REF}}$ to the route resources as F. The concerned resource window is $I_{\mathcal{M}} := \bigotimes_{m \in \mathcal{M}} [0, |m|]$ and the resource dominance relation is $\sqsubseteq_{\mathcal{M}} := (=, \ldots, =)$. For every $t \in \mathcal{T}$ we have the partition $\mathcal{M} = M^{\mathrm{C}}(t) \cup M^{\mathrm{R}}(t) \cup \{M(t)\}$. Using this partition, we split the resource vectors as follows:

$$b = \left(b_{M^{\mathrm{C}}(t)}, b_{M^{\mathrm{R}}(t)}, b_{M(t)}\right) \in I_{M^{\mathrm{C}}(t)} \times I_{M^{\mathrm{R}}(t)} \times I_{M(t)} = I_{\mathcal{M}}$$

According to (5.10) we have for $t \in \mathcal{T}$

$$F_{t^-,t^+}\left(b_{M^{\mathrm{C}}(t)}, b_{M^{\mathrm{R}}(t)}, b_{M(t)}\right) := \left(\begin{pmatrix} b_{M^{\mathrm{C}}(t)} \\ 0 \\ b_{M(t)} - 1 \end{pmatrix} + \mathbb{R}^{\mathcal{M}}_{\sqsubseteq_{\mathcal{M}} 0}\right) \cap I_{\mathcal{M}}$$

for $b \in I_{\mathcal{M}}$. Because of the definition of $\sqsubseteq_{\mathcal{M}}$ and $b \in I_{\mathcal{M}}$, this is equivalent to

$$F_{t^-,t^+}\left(b_{M^{\mathrm{C}}(t)}, b_{M^{\mathrm{R}}(t)}, b_{M(t)}\right) := \left\{\left(b_{M^{\mathrm{C}}(t)}, 0, b_{M(t)} - 1\right) \mid b_{M(t)} \geq 1\right\}$$

Recall the inverted relation according to Definition 16. The inverted relation is given by

$$F^{-1}_{t^-,t^+} := \left\{\left(r^{t^+}, r^{t^-}\right) \mid \left(r^{t^-}, r^{t^+}\right) \in F_{t^-,t^+}\right\}$$

for all $t \in \mathcal{T}$. Applying the respective definitions we get

$$\begin{aligned} F_{t^-,t^+} &= \left\{ \left(\left(b_{M^C(t)}, b_{M^R(t)}, b_{M(t)}\right), \left(b_{M^C(t)}, 0, b_{M(t)} - 1\right) \right) \mid b \in I_{\mathcal{M}}, b_{M(t)} \geq 1 \right\} \\ \Rightarrow F^{-1}_{t^-,t^+} &= \left\{ \left(\left(b_{M^C(t)}, 0, b_{M(t)} - 1\right), \left(b_{M^C(t)}, b_{M^R(t)}, b_{M(t)}\right) \right) \mid b \in I_{\mathcal{M}}, b_{M(t)} \geq 1 \right\} \\ &= \left\{ \left(\left(b_{M^C(t)}, 0, b_{M(t)}\right), \left(b_{M^C(t)}, b_{M^R(t)}, b_{M(t)} + 1\right) \right) \mid b \in I_{\mathcal{M}}, b_{M(t)} \leq |m| - 1 \right\} \end{aligned}$$

and therefore

$$F^{-1}_{t^-,t^+}(b) = \begin{cases} \left(b_{M^C(t)}, I_{M^R(t)}, b_{M(t)} + 1\right) & \text{if } b_{M^R(t)} = 0_{M^R(t)}, b_{M(t)} \leq |m| - 1 \\ \emptyset & \text{otherwise} \end{cases}$$

We show that $F^{-1}_{t^-,t^+}$ is a resource extension function with respect to $\sqsupseteq_{\mathcal{M}}$ for all $t \in \mathcal{T}$. Since we have equality everywhere in the resource dominance relation, we have $(\sqsupseteq_{\mathcal{M}}) = (\sqsubseteq_{\mathcal{M}})$. The respective conditions from Definition 12 and Definition 13 are obviously fulfilled since for $r^v, \tilde{r}^v \in I_{\mathcal{M}}$ we have $r^v \sqsupseteq \tilde{r}^v \Leftrightarrow r^v = \tilde{r}^v$.
Since considering the other resources and contraction of vertices does not destroy invertibility follows that $\widehat{\text{REF}}$ is invertible.

□

5.3 Solving of the Master Problem

With the methods developed in Section 5.1 and Section 5.2 we are able to solve the relaxed master problem. Having a solution of this relaxed problem, it is not guaranteed that the solution is an integer solution. But this is a condition for feasibility in the master problem. If the solution is a fractional solution, we apply the “Branch-and-Bound” rule in order to receive an integer solution. We describe how Branch-and-Bound works in general and how we can apply this to our problem. Then we discuss a number of possible branching rules. Finally, we show how the branching rules are chosen. The branching rules are mainly taken over by the branching rules created by [Kai16]. The branching rule concerning the choice of routes is modified.

5.3.1 Branch-and-Bound

The set of feasible solutions for the relaxed problem is divided into two separate sets by a hyperplane. We express this restriction as an inequality. We evaluate both sets individually. An optimal solution of one of these sets is an optimal solution for the entire problem. If we repeat this procedure iteratively, we get smaller problems. This procedure is called “Branching”. We receive a tree of smaller problems where the root is the original master problem and the respective child nodes are the smaller problems resulting by branching.

The overall process works as follows: Starting with the root, we solve the respective relaxed problem. If we receive a fractional solution, we branch the problem and create child nodes. We continue this procedure for each child until we either receive a feasible integer solution or the problem becomes infeasible. After this, we iteratively assign to each node the feasible solution of its child nodes with the smallest objective value, or infeasible, if both child nodes are infeasible. Therefore, the solution assigned to the root is the optimal solution of the entire problem.
Depending on the problem and the branching strategy, the decision tree might become quite large. Therefore we have to think about methods to improve this behavior. If we already have an initial feasible solution, then the value of this solution is an upper bound to the optimal solution value. As we know, the value of the relaxation is a lower bound to the optimal value of this problem. Therefore, if the value of the relaxation for a subtree is greater than the value of the initial solution, we know that the optimal solution does not lie in this subtree. Thus we can completely neglect this subtree. This method is called “Bounding”.

Application to the Master Problem

A branching decision is always an inequality that we add to the master problem. As suggested by [Kai16], we only use inequalities written in terms of $(x, z, e, u) \in X_v, v \in \mathcal{V}$ such that the structure of the subproblems is not changed. Further, we do not want to affect the symmetry of the subproblems since we exploit this when we solve the subproblems. Thus, we discuss the decision rules with respect to keeping the symmetry. If an inequality concerns only a single subproblem, we move this restriction completely to the subproblem. This has the advantage that we still create feasible columns. However, moving the inequalities to the subproblems might lead to a change of their structure and therefore might make it harder to solve. In Section 5.3.2 we discuss several branching rules.

5.3.2 Branching Rules

In the following, we present a number of branching rules. In order to keep the branching tree small, we try to create decisions that lead to a balanced branching. The branching is used for enforcing integrality in the master problem. In order to discuss the branching rules, we introduce for $v \in \mathcal{V}$ the set of vertices that can be reached from v and the set of vertices from which v can be reached:

$$\mathrm{N}_G^{++}(v) := \{w \in V \mid \exists v\text{-}w\text{-path in } G\} \qquad \mathrm{N}_G^{--}(v) := \{w \in V \mid \exists w\text{-}v\text{-path in } G\}$$

Assignment of Trips

We first consider a branching on the components of the image with respect to ψ. This is a branching on the single variables of the master problem and thus the most specific branching rule we consider. We fix a value for the expression

$$\psi\left(x^v, z^v, e^v\right)_t = \sum_{s \in \mathrm{N}_G^-(t)} x_{s,t}^v$$

for some $v \in \mathcal{V}, t \in \mathcal{T}$. This decision can be interpreted as determining whether trip t is fulfilled by vehicle v. This is a very specific decision since it implicitly chooses the route $M(t)$ for customer $(M \circ C)(t)$ and assigns it to vehicle v.
Branching down means setting $\sum_{s \in \mathrm{N}_G^-(t)} x_{s,t}^v = 0$ for some $v \in \mathcal{V}, t \in \mathcal{T}$. This can be implemented in (SP$_v$) by setting

$$x_{s,t}^v = 0 \qquad \text{for all } s \in \mathrm{N}_G^-(t)$$

This corresponds to deleting the respective arcs in the task graph.
Branching up means demanding $\sum_{s \in \mathrm{N}_G^-(t)} x_{s,t}^v = 1$ for some $v \in \mathcal{V}, t \in \mathcal{T}$. With the constraints (3.3), (5.1), (5.2) and the fact that G is acyclic, this is equivalent to setting

$$x_{s,u}^v = 0 \qquad \text{for all } (s,u) \in A \cap \left(\left(\mathrm{N}_G^{--}(t) \setminus \{t\}\right) \times \left(\mathrm{N}_G^{++}(t) \setminus \{t\}\right)\right)$$

This corresponds to deleting all arc skipping trip t.
Branching up excludes a lot of possible assignments, while branching only forbids one such assignment. This leads to an quite unbalanced tree. Another disadvantage is that this rule destroys symmetry between the subproblems as it concerns only one subproblem. Nevertheless, there is always a branching decision that can be made if the previous solution was not integral. This means, this rule is enough to completely ensure integrality in the master problem.

Length of Vehicle Duties

Another suggestion is to consider not a single trip, but the sum up over all trips for the components of the image of ψ. We fix a value for the expression

$$\sum_{t \in \mathcal{T}} \psi\left(x^v, z^v, e^v\right)_t = \sum_{t \in \mathcal{T}} \sum_{s \in \mathrm{N}_G^-(t)} x_{s,t}^v$$

for some $v \in \mathcal{V}$. This can be interpreted as determining the duty length for vehicle v. As the length of the duty can be expressed linearly in terms of (x^v, z^v, e^v) and in terms of images of ψ it would be possible to include this decision in the master problem without changing the subproblems. But since this decision affects only one specific vehicle, we move it to the subproblem.

This leads to the following general setting of the subproblem using a lower bound l^{LB} and an upper bound l^{UB} for the vehicle length.

$$\begin{aligned} \min \quad & g^v\left(x^v, z^v, e^v\right) \\ \text{s.t.} \quad & l^{\mathrm{LB}} \leq \sum_{t\in\mathcal{T}} \sum_{s\in \mathrm{N}_G^-(t)} x_{s,t}^v \leq l^{\mathrm{UB}} \\ & (x^v, z^v, e^v) \in X_v \end{aligned} \tag{5.11}$$

As proven by [Kai16], this problem can be solved by using the resource length as introduced before. We modify the resource extension function for $(c, e, l, b) \in I$ as follows:

$$\widehat{\mathrm{REF}}_{t,d^e}(c, e, l, b) := \left(\left\{(c, e, l, b) \mid l^{\mathrm{LB}} \leq l \leq l^{\mathrm{UB}}\right\} + \mathbb{R}^{\mathcal{U}}_{\sqsubseteq 0}\right) \cap I$$

This is a coarser branching rule as the assignment of trips, but still destroys the symmetry between the subproblems. In contrast to before, this rule alone is not sufficient to completely ensure integrality in the master problem.

Choice of Multimodal Routes

The next approach is to sum up over all vehicles for the components of the image of ψ. We fix a value for the expression

$$\sum_{v\in\mathcal{V}} \psi\left(x^v, z^v, e^v\right)_t = \sum_{v\in\mathcal{V}} \sum_{s\in \mathrm{N}_G^-(t)} x_{s,t}^v$$

for some $t \in \mathcal{T}$. This can be interpreted as deciding whether trip t is fulfilled by some vehicle. Because of the cover constraint (5.3), this decision directly determines the route variable u_m for $m := M(t)$ and therefore all other trips $t' \in M^{-1}(m)$. Because of this implication, we directly regard the case of branching on u_m.
Branching down means setting $u_m = 0$ for some $m \in \mathcal{M}$. From this follows $\sum_{s\in \mathrm{N}_G^-(t)} x_{s,t}^v = 0$ for all $t \in M^{-1}(m), v \in \mathcal{V}$ and therefore

$$x_{s,t}^v = 0 \qquad \text{for all } v \in \mathcal{V}, t \in M^{-1}(m), s \in \mathrm{N}_G^-(t)$$

This corresponds to deleting the respective arcs in all task graphs.
Branching up means setting $u_m = 1$ for some $m \in \mathcal{M}$ and therefore demanding $\sum_{s\in \mathrm{N}_G^-(t)} x_{s,t}^v = 0$ for all $t \in M^{-1}(m), v \in \mathcal{V}$. Due to the cover constraint (3.1) this is equivalent to setting $u_{m'} = 0$ for all $m' \in (C^{-1} \circ C)(m)\backslash\{m\}$. We therefore set

$$x_{s,t}^v = 0 \qquad \text{for all } v \in \mathcal{V}, m' \in \left(C^{-1} \circ C\right)(m)\backslash\{m\}, \\ t \in M^{-1}(m'), s \in \mathrm{N}_G^-(t)$$

In other words, stating that a multimodal route is fulfilled by some vehicles is equivalent to stating that all trips belonging to other routes of the same customer are not fulfilled by any vehicle, if (3.1) and (5.3) hold. Again, this can be realized by deleting the respective arcs in all task graphs.

This branching rule maintains the symmetry for the various subproblems since for all $v \in \mathcal{V}$ the same arcs are deleted. Therefore, we can still solve all subproblems by just one executing of the algorithm. Further, the tree is more balanced than before. Similar to the duty length, this rule alone is not sufficient to completely ensure integrality in the master problem.

Number of Used Vehicles

Finally, we regard the branching on the number of used vehicles. We say, a vehicle is used if it fulfills at least one trip. The only duty that a vehicle can have to be unused is the duty with no trip, uniquely described by the path $(d^{\mathrm{s}}, v, d^{\mathrm{e}})$ in G. For a duty $(x, z, e) \in X_v$ for $v \in \mathcal{V}$ the term

$$\sum_{(s,t) \in A \cap (\mathcal{V} \times \mathcal{T})} x_{s,t}$$

is one if and only if the vehicles serves at least one trip. One possibility would be to decide if a specific vehicle is used or not. But the various subproblems are to similar as only the vehicle vertex is different and therefore this branching rule leads to an unbalanced tree.

Instead, we regard the number of used vehicles. For $(x^v, e^v, z^v) \in X_v$ the number of used vehicles can be expressed as

$$\sum_{v \in \mathcal{V}} \sum_{(s,t) \in A \cap (\mathcal{V} \times \mathcal{T})} x^v_{s,t}$$

Using this branching decision requires adding a suitable inequality to the master problem. As these inequalities concern the duties of all vehicles, they cannot be moved to the subproblems. We modify the master problem using a lower bound v^{LB} and an upper bound v^{UB} for the number of used cars as follows:

$$
\begin{aligned}
\min \quad & \sum_{v \in \mathcal{V}} g\left(x^v, z^v, e^v\right) + \sum_{m \in \mathcal{M}} u_m c_m^{\mathrm{r}} \\
\text{s.t.} \quad & \sum_{m \in C^{-1}(c)} u_m = 1 && \text{for all } c \in \mathcal{C} && (3.1) \\
& \sum_{v \in \mathcal{V}} \sum_{s \in \mathrm{N}_G^-(t)} x_{s,t}^v = u_m && \text{for all } m \in \mathcal{M}, t \in m && (5.3) \\
& v^{\mathrm{LB}} \leq \sum_{v \in \mathcal{V}} \sum_{(s,t) \in A \cap (\mathcal{V} \times \mathcal{T})} x_{s,t}^v \leq v^{\mathrm{UB}} && && (5.12) \\
& (x^v, z^v, e^v) \in X_v && \text{for all } v \in \mathcal{V} \\
& u_m \in \{0,1\}^{\mathcal{M}}
\end{aligned}
$$

5.3.3 Choosing Branching Decisions

5.4 Variants of the Problem Formulation

5.4.1 Master Problem with Route Choice Restriction

Remember the formulation from before:

$$
\begin{aligned}
\min \quad & \sum_{v \in \mathcal{V}} \min g\left(\psi^{-1}\left(y^v\right) \cap X_v\right) + \sum_{m \in \mathcal{M}} u_m c_m^{\mathrm{r}} \\
\text{s.t.} \quad & \sum_{m \in C^{-1}(c)} u_m = 1 && \text{for all } c \in \mathcal{C} && (3.1) \\
& \sum_{v \in \mathcal{V}} y_t^v = u_m && \text{for all } m \in \mathcal{M}, t \in m \\
& y^v \in \psi\left(X_v\right) && \text{for all } v \in \mathcal{V} \\
& u_m \in \{0,1\} && \text{for all } m \in \mathcal{M}
\end{aligned}
$$

The constraints (3.1) depend only on u_m. Therefore, we create another subproblem for the choice of routes. We define the set of feasible route choices as follows:

$$
\hat{X} := \left\{ \{0,1\}^{\mathcal{M}} | \sum_{m \in C^{-1}(c)} u_m = 1 \text{ for all } c \in \mathcal{C} \right\}
$$

We introduce variable $\hat{u}$ and the respective route cost function $\hat{g}$ and rewrite (MMILP) again:

$$\begin{aligned}
\min \quad & \sum_{v\in\mathcal{V}} \min g\left(\psi^{-1}\left(y^v\right)\cap X_v\right) + \hat{g}\left(\hat{u}\right) & \\
\text{s.t.} \quad & \sum_{v\in\mathcal{V}} y^v_t = u_m & \text{for all } m\in\mathcal{M}, t\in m \\
& y^v \in \psi\left(X_v\right) & \text{for all } v\in\mathcal{V} \\
& \hat{u}\in\hat{X} &
\end{aligned}$$

Column Generation

For every $v\in\mathcal{V}$, let $\mathcal{I}_v$ be an index set for the finitely many points in $\psi\left(X_v\right)$ and let the columns of $Y^v\in\mathbb{R}^{\mathcal{T}\times\mathcal{I}_v}$ be exactly those points. Let $\hat{\mathcal{I}}$ be an index set for the finitely many points in $\hat{X}$ and let the columns of $\hat{Y}\in\mathbb{R}^{\mathcal{M}\times\hat{\mathcal{I}}}$ be exactly those points. Let $G^v\in\mathbb{R}^{1\times\mathcal{I}}$ be the respective values of $\min g\left(\psi^{-1}(\cdot)\cap X_v\right)$ and $\hat{G}\in\mathbb{R}^{1\times\hat{\mathcal{I}}}$ be the respective route costs. Then we can reformulate the master problem as

$$\begin{aligned}
\min \quad & \sum_{v\in\mathcal{V}} G^v\lambda^v + \hat{G}\hat{\lambda} & \\
\text{s.t.} \quad & \sum_{v\in\mathcal{V}} Y^v_{t,\cdot}\lambda^v = \hat{Y}_{m,\cdot}\hat{\lambda} & \text{for all } m\in\mathcal{M}, t\in m \\
& \sum_{i\in\mathcal{I}_v} \lambda^v_i = 1 & \text{for all } v\in\mathcal{V} \\
& \sum_{i\in\hat{\mathcal{I}}} \hat{\lambda}_i = 1 & \\
& \lambda^v\in\{0,1\}^{\mathcal{I}_v} & \text{for all } v\in\mathcal{V} \\
& \hat{\lambda}\in\{0,1\}^{\hat{\mathcal{I}}} &
\end{aligned} \tag{IMMP}$$

We regard the LP-relaxation by dropping the integrality constraints:

$$
\begin{aligned}
\min \quad & \sum_{v \in \mathcal{V}} G^v \lambda^v + \hat{G}\hat{\lambda} && && \text{(LMMP)} \\
\text{s.t.} \quad & \sum_{v \in \mathcal{V}} Y^v_{t,\cdot} \lambda^v = \hat{Y}_{m,\cdot} \hat{\lambda} && \text{for all } m \in \mathcal{M}, t \in m \\
& \sum_{i \in \mathcal{I}_v} \lambda^v_i = 1 && \text{for all } v \in \mathcal{V} \\
& \sum_{i \in \hat{\mathcal{I}}} \hat{\lambda}_i = 1 \\
& \lambda^v \in \mathbb{R}^{\mathcal{I}_v}_{\geq 0} && \text{for all } v \in \mathcal{V} \\
& \hat{\lambda} \in \mathbb{R}^{\hat{\mathcal{I}}}_{\geq 0}
\end{aligned}
$$

We reduce the size by considering only subsets $\mathcal{J}_v \subset \mathcal{I}_v$ and $\hat{\mathcal{J}} \subset \hat{\mathcal{I}}$ and formulate the relaxed restricted master problem:

$$
\begin{aligned}
\min \quad & \sum_{v \in \mathcal{V}} G^v_{\mathcal{J}_v} \lambda^v + \hat{G}_{\hat{\mathcal{J}}}\hat{\lambda} && && \text{(LRMMP)} \\
\text{s.t.} \quad & \sum_{v \in \mathcal{V}} Y^v_{t,\mathcal{J}_v} \lambda^v = \hat{Y}_{m,\hat{\mathcal{J}}} \hat{\lambda} && \text{for all } m \in \mathcal{M}, t \in m \\
& \sum_{i \in \mathcal{J}_v} \lambda^v_i = 1 && \text{for all } v \in \mathcal{V} \\
& \sum_{i \in \hat{\mathcal{J}}} \hat{\lambda}_i = 1 \\
& \lambda^v \in \mathbb{R}^{\mathcal{J}_v}_{\geq 0} && \text{for all } v \in \mathcal{V} \\
& \hat{\lambda} \in \mathbb{R}^{\hat{\mathcal{J}}}_{\geq 0}
\end{aligned}
$$

For the dual relaxed restricted master problem, we introduce dual variables $\gamma \in \mathbb{R}^{\mathcal{T}}$, $\mu \in \mathbb{R}^{\mathcal{V}}$ and $\alpha \in \mathbb{R}$. The dual problem is:

$$
\begin{aligned}
\max \quad & \sum_{v\in\mathcal{V}} \mu_v + \alpha && && \text{(DLRMMP)}\\
\text{s.t.} \quad & \sum_{t\in\mathcal{T}} Y^v_{t,i}\gamma_t + \mu_v \le G^v_i && \text{for all } v\in\mathcal{V}, i\in\mathcal{J}_v\\
& \alpha - \sum_{m\in\mathcal{M}}\sum_{t\in m} \hat{Y}_{m,i}\gamma_t \le \hat{G}_i && \text{for all } i\in\hat{\mathcal{J}}\\
& \gamma\in\mathbb{R}^{\mathcal{T}}\\
& \mu\in\mathbb{R}^{\mathcal{V}}\\
& \alpha\in\mathbb{R}
\end{aligned}
$$

Solving of the Relaxed Master Problem

Let $(\gamma^*, \mu^*, \alpha^*)$ be a solution of (DLRMMP) with $\mathcal{J}_v \subset \mathcal{I}_v$ for all $v \in \mathcal{V}$ and $\hat{\mathcal{J}} \subset \hat{\mathcal{I}}$. We want to find out whether $(\gamma^*, \mu^*, \alpha^*)$ corresponds to an optimal solution of (LMMP). This is the case if it is feasible for the dual relaxed master problem, i.e. the following constraints hold for the entire sets $\mathcal{I}_v$ and $\hat{\mathcal{I}}$. This means, the following equations hold for $(\gamma^*, \mu^*, \alpha^*)$:

$$\sum_{t\in\mathcal{T}} Y^v_{t,i}\gamma^* + \mu^*_v \le G^v_i \qquad \text{for all } v\in\mathcal{V}, i\in\mathcal{I}_v \tag{5.13}$$

$$\alpha^* - \sum_{m\in\mathcal{M}}\sum_{t\in m} \hat{Y}_{m,i}\gamma^*_t \le \hat{G}_i \qquad \text{for all } i\in\hat{\mathcal{I}} \tag{5.14}$$

In order to find an optimal solution of (LMMP) we have to find indices $i \in \mathcal{I}_v$ or $j \in \hat{\mathcal{I}}$ where the previous constraints are violated. This leads to the following subproblems:

$$
\begin{aligned}
&\text{Find } i\in\mathcal{I}_v\backslash\mathcal{J}_v \text{ s.t.} \quad && \sum_{t\in\mathcal{T}} Y^v_{t,i}\gamma^*_t + \mu^*_v > G^v_i && \text{for } v\in\mathcal{V}\\
&\text{Find } i\in\hat{\mathcal{I}}\backslash\hat{\mathcal{J}} \text{ s.t.} \quad && \alpha^* - \sum_{m\in\mathcal{M}}\sum_{t\in m} \hat{Y}_{m,i}\gamma^*_t > \hat{G}_i
\end{aligned}
$$

The vehicle subproblem (SP_v) was already considered before. For the route choice, an additional subproblem arises.

Route Subproblem

The route subproblem for finding violated constraints (5.14) reads as follows:

$$
\begin{aligned}
\min \quad & \sum_{m \in \mathcal{M}} u_m \left(c^{\mathrm{r}}_m + \sum_{t \in m} \gamma^*_t \right) && \\
\text{s.t.} \quad & \sum_{m \in C^{-1}(c)} u_m = 1 && \text{for all } c \in \mathcal{C} \\
& u_m \in \{0,1\} && \text{for all } m \in \mathcal{M}
\end{aligned}
\tag{SP$^{\mathrm{m}}$}
$$

This problem is easy to solve: For every $c \in \mathcal{C}$ choose the multimodal route $m \in C^{-1}(c)$ with the smallest cost $c^{\mathrm{r}}_m + \sum_{t \in m} \gamma^*_t$.
Let $\bar{u}$ be an optimal solution of (SP$^{\mathrm{m}}$). If $\operatorname{val}(\bar{u}) < \alpha^*$ then add this to $\hat{\mathcal{J}}$ and continue the master problem.

Chapter 6

Instance Creation

6.1 Route Creation

We are not given the set of routes $\mathcal{M}$ in advance. For each customer $c \in \mathcal{C}$, we have start and end location $p_c^{\text{start}}, p_c^{\text{end}}$ and a start and end time $\hat{z}_c^{\text{start}}, \hat{z}_c^{\text{end}}$. All the trips of the customer lie in this interval, i.e.

$$\hat{z}_c^{\text{start}} \leq z_m^{\text{start}} \qquad z_m^{\text{end}} \leq \hat{z}_c^{\text{end}} \qquad \text{for all } m \in C^{-1}(c).$$

Basic Restrictions

To simplify the creation of the routes, we make some assumptions. For every route $m \in \mathcal{M}$ holds:

- There are not two car trips in a row.
- There is no car trip between two public transport trips.
- The number of public transport trips is restricted. Usually, one can reach every station with at most two changes.
- We define a walking distance d^{walk}. If the distance between the start position and the first station or between the last station and the end position, there is no car trip necessary.

We assume that we have some oracle that provides the set of feasible public transport routes for customer $c \in \mathcal{C}$:

$$M_c = \Big\{(s_1, z_1, s_2, z_2) \,|\, s_1, s_2 \in \mathcal{S}, \hat{z}_c^{\text{start}} \leq t_1 < t_2 \leq \hat{z}_c^{\text{end}}, \text{ there is a public transport route from } s_1 \text{ to } s_2 \text{ with start time } z_1 \text{ and end time } z_2\Big\}$$

The fact, whether the customer changes during his usage of public transport, has no effect on the model. Thus, we can consider each element in M_c as a public transport trip.

Route Creation

We create the set of multimodal routes $\mathcal{M}$. For this, we set a car trip before and after each public transport trip in order to bring the customer from his start to his destination, except when it is possible to walk the distance. We also have to consider the given time restrictions. Further, we create the pure car trips. How the set $\mathcal{M}$ is created in detail, is described in Algorithm 3.
Until now, we do not consider any changing times between a car trip and a public transport trip.
Further, we assume that the given customer start and and times are feasible, i.e. $\hat{z}_c^{\text{start}} + t_{p_c^{\text{start}}, p_c^{\text{end}}} \leq \hat{z}_c^{\text{end}}$ for all $c \in \mathcal{C}$.

Further Restrictions

If the routes are created as described in Algorithm 3, there are routes using every available station as long as it is feasible. Most of these routes are obviously bad for the customer since they cause a big detour. What is more, a large number of routes enlarge the problem size and leads to a bad performance for solving it. Therefore, we try to restrict the set of alternatives to a reasonable size.

Example 2. Let $\mathcal{S} = \{s_1, \ldots, s_n\}$ with a single public transport ride serving all stations. Let $\mathcal{C} = \{c_1, c_2\}$ with $p_{c_1}^{\text{end}} = s_n$ and $p_{c_2}^{\text{start}} = s_k$ for a certain $k \in [n-1]$. The alternative routes are

$$\mathcal{M} = \underbrace{\left\{\left(\left(p_{c_1}^{\text{start}}, s_i\right), (s_i, s_n)\right) | i \in [n-1]\right\}}_{\text{for } c_1} \cup \underbrace{\left\{\left(s_k, p_{c_2}^{\text{end}}\right)\right\}}_{\text{for } c_2}$$

with $(p_{c_1}^{\text{start}}, s_k) \prec \left(s_k, p_{c_2}^{\text{end}}\right)$ and $(p_{c_1}^{\text{start}}, s_i) \not\prec \left(s_k, p_{c_2}^{\text{end}}\right)$ for all $i \in [n] \backslash \{k\}$.
We get the only solution, where only one car is needed, when c_1 drives to s_k, wherever the station s_k is. Every route of c_1 can be the optimal route, considering the other customers. Therefore, an exact reduction of $\mathcal{M}$ is not possible without the the risk of cutting off the optimal solution.

It is not practicable to consider all possible multimodal routes due to computation reasons. But it is also not possible to reduce the number of routes without risking to lose the optimal solution. Hence, we try to make reasonable restrictions which keep the problem size small.

Pareto Optimality

Algorithm 3: Creation of the routes

Input: customer set $\mathcal{C}$; $p_c^{\text{start}}, p_c^{\text{end}}, \hat{z}_c^{\text{start}}, \hat{z}_c^{\text{end}}, M_c$ for all $c \in \mathcal{C}$

Output: set of routes $\mathcal{M}$, set of trips $\mathcal{T}_{\text{car}}, \mathcal{T}_{\text{public}}$

$\mathcal{T}_{\text{car}} \leftarrow \emptyset$;

$\mathcal{T}_{\text{public}} \leftarrow \emptyset$;

$\mathcal{M} \leftarrow \emptyset$;

foreach $c \in \mathcal{C}$ **do**

  foreach $(s_1, z_1, s_2, z_2) \in M_c$ **do**

    create public transport trip t;

    $p_t^{\text{start}} \leftarrow s_1, p_t^{\text{end}} \leftarrow s_2, z_t^{\text{start}} \leftarrow z_1, z_t^{\text{end}} \leftarrow z_2$;

    create car trips t_1, t_2;

    $p_{t_1}^{\text{start}} \leftarrow p_c^{\text{start}}, p_{t_1}^{\text{end}} \leftarrow s_1, z_{t_1}^{\text{start}} \leftarrow z_1 - t_{p_c^{\text{start}}, s_1}, z_{t_1}^{\text{end}} \leftarrow z_1$;

    $p_{t_2}^{\text{start}} \leftarrow s_2, p_{t_2}^{\text{end}} \leftarrow p_c^{\text{end}}, z_{t_2}^{\text{start}} \leftarrow z_2, z_{t_2}^{\text{end}} \leftarrow z_2 + t_{s_2, p_c^{\text{end}}}$;

    if $\hat{z}_c^{\text{start}} \leq z_{t_1}^{\text{start}} \wedge z_{t_2}^{\text{end}} \leq \hat{z}_c^{\text{end}}$ **then**

      create multimodal route m;

      $\mathcal{T}_{\text{public}} \leftarrow \mathcal{T}_{\text{public}} \cup \{t\}$;

      if $d_{p_c^{\text{start}}, s_1} \geq d^{\text{walk}}$ **then** $m \leftarrow (t_1, t)$; $\mathcal{T}_{\text{car}} \leftarrow \mathcal{T}_{\text{car}} \cup \{t_1\}$;

      else $m \leftarrow (t)$;

      if $d_{s_2, p_c^{\text{end}}} \geq d^{\text{walk}}$ **then** append t_2 to m; $\mathcal{T}_{\text{car}} \leftarrow \mathcal{T}_{\text{car}} \cup \{t_2\}$;

      $C(m) \leftarrow c$;

      $\mathcal{M} \leftarrow \mathcal{M} \cup \{m\}$;

    end

  end

  create car trips t_1, t_2;

  $p_{t_1}^{\text{start}} \leftarrow p_c^{\text{start}}, p_{t_1}^{\text{end}} \leftarrow p_c^{\text{end}}, z_{t_1}^{\text{start}} \leftarrow \hat{z}_c^{\text{start}}, z_{t_1}^{\text{end}} \leftarrow \hat{z}_c^{\text{start}} + t_{p_c^{\text{start}}, p_c^{\text{end}}}$;

  $p_{t_2}^{\text{start}} \leftarrow p_c^{\text{start}}, p_{t_2}^{\text{end}} \leftarrow p_c^{\text{end}}, z_{t_2}^{\text{start}} \leftarrow \hat{z}_c^{\text{end}} - t_{p_c^{\text{start}}, p_c^{\text{end}}}, z_{t_2}^{\text{end}} \leftarrow \hat{z}_c^{\text{end}}$;

  create multimodal routes m_1, m_2;

  $m_1 \leftarrow (t_1), m_2 \leftarrow (t_2)$;

  $\mathcal{T}_{\text{car}} \leftarrow \mathcal{T}_{\text{car}} \cup \{t_1, t_2\}, \mathcal{M} \leftarrow \mathcal{M} \cup \{m_1, m_2\}$;

end

return $\mathcal{M}, \mathcal{T}_{\text{car}}, \mathcal{T}_{\text{public}}$

The idea is to choose only Pareto optimal multimodal routes (cf. Kaiser/Knoll, cap. 3.2.2) in order to determine good routes.

Definition 18 (Pareto optimality)**.** Let $V \subset \mathbb{R}^n$.

1. The partial order $\leq$ on $\mathbb{R}^n$ is given by

$$v \leq w \qquad :\Leftrightarrow \qquad v_i \leq w_i \qquad \forall i \in [n] \qquad \text{for all } v, w \in \mathbb{R}^n$$

2. An element $w \in V$ is Pareto optimal in V if it is minimal with respect to $\leq$ in V, i.e.

$$v \leq w \qquad \Rightarrow \qquad v = w \qquad \text{for all } v \in V$$

3. The Pareto frontier of V with respect to $\leq$ is the set of Pareto optimal elements in V, i.e.

$$\min_{\leq} V := \{w \in V | \forall v \in V : v \leq w \Rightarrow v = w\}$$

Let $m \in \mathcal{M}$ be a multimodal route. We define

$$\varphi : \mathcal{M} \to \mathbb{R}^5 \qquad\qquad m \mapsto \begin{pmatrix} c^{\mathrm{r}} + \sum_{t \in m \cap \mathcal{T}_{\mathrm{car}}} c_t^{\mathrm{t}} \\ c^{\mathrm{r}} \\ |\mathcal{T}_{\mathrm{car}} \cap \{t \in m\}| \\ \sum_{t \in m \cap \mathcal{T}_{\mathrm{car}}} z_t^{\mathrm{end}} - z_t^{\mathrm{start}} \\ \sum_{t \in m \cap \mathcal{T}_{\mathrm{car}}} f_t^{\mathrm{t}} \end{pmatrix}$$

The function φ grades a route to their costs, their route costs, the number of cars needed, the time of a car needed and the fuel consumption.
From now on, we will use the Pareto frontier of $\varphi(\mathcal{M})$ as a restricted route set:

$$\hat{\mathcal{M}} := min_{\leq} \varphi(\mathcal{M}) \tag{6.1}$$

Previous Formulations

1 (MILP)

$$
\begin{aligned}
\min \sum_{s\in\mathcal{V}} \sum_{t\in \mathrm{N}^+_G(s)\setminus\{d^\mathrm{e}\}} & x_{s,t}c^\mathrm{v} \\
+\sum_{t\in\mathcal{T}} \sum_{s\in \mathrm{N}^-_G(t)} & \left[x_{s,t}\left(c^\mathrm{d}_{s,t}+c^\mathrm{t}_t\right)+\sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t}\left(c^\mathrm{d}_{s,r}+c^\mathrm{d}_{r,t}-c^\mathrm{d}_{s,t}\right)\right] && \text{(MILP)}\\
\text{s.t.} \sum_{t\in\mathrm{N}^-_G(s)} x_{t,s} &= \sum_{t\in\mathrm{N}^+_G(s)} x_{s,t} && \text{for all } s\in V\setminus\{d^\mathrm{s},d^\mathrm{e}\} && (1)\\
\sum_{s\in\mathrm{N}^-_G(t)} x_{s,t} &= 1 && \text{for all } t\in\mathcal{V} && (2)\\
\sum_{t\in C^{-1}(c)}\sum_{s\in\mathrm{N}^-_G(t)} x_{s,t} &= 1 && \text{for all } c\in\mathcal{C} && (3)\\
\sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t} &\le x_{s,t} && \text{for all } t\in\mathcal{T}, s\in\mathrm{N}^-_G(t) && (4)\\
e_s &\le f^0_s && \text{for all } s\in\mathcal{V} && (5)\\
0 &\le e_s-\sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t}f^\mathrm{d}_{s,r} && \text{for all } t\in\mathcal{T}, s\in\mathrm{N}^-_G(t) && (6)\\
e_t &\le 1-f^\mathrm{t}_t-\sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t}f^\mathrm{d}_{r,t} && \text{for all } t\in\mathcal{T}, s\in\mathrm{N}^-_G(t) && (7)\\
e_t &\le e_s-x_{s,t}\left(f^\mathrm{d}_{s,t}+f^\mathrm{t}_t\right)-\sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t}\left(f^\mathrm{d}_{s,r}+f^\mathrm{t}_r+f^\mathrm{d}_{r,t}-f^\mathrm{d}_{s,t}\right)+(1-x_{s,t})\\
& && \text{for all } t\in\mathcal{T}, s\in\mathrm{N}^-_G(t) && (8)\\
x_{s,t} &\in\{0,1\} && \text{for all } (s,t)\in A && (9)\\
z_{s,r,t} &\in\{0,1\} && \text{for all } t\in\mathcal{T}, s\in\mathrm{N}^-_G(t), r\in\mathcal{R}_{s,t} && (10)\\
e_s &\in[0,1] && \text{for all } s\in V\setminus\{d^\mathrm{s},d^\mathrm{e}\} && (11)
\end{aligned}
$$

2 (AMILP)

$$
\begin{aligned}
\min \sum_{s\in\mathcal{V}} \sum_{t\in\mathrm{N}^+_G(s)\setminus\{d^{\mathrm{e}}\}} & x_{s,t}c^{\mathrm{v}} \\
+\sum_{t\in\mathcal{T}} \sum_{s\in\mathrm{N}^-_G(t)} & \left[x_{s,t}\left(c^{\mathrm{d}}_{s,t}+c^{\mathrm{t}}_{t}\right)+\sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t}\left(c^{\mathrm{d}}_{s,r}+c^{\mathrm{d}}_{r,t}-c^{\mathrm{d}}_{s,t}\right)\right] && \text{(AMILP)}\\
\text{s.t.} \sum_{t\in\mathrm{N}^-_G(s)} x_{t,s} &= \sum_{t\in\mathrm{N}^+_G(s)} x_{s,t} && \text{for all } s\in V\setminus\{d^{\mathrm{s}},d^{\mathrm{e}}\} \quad (1)\\
\sum_{s\in\mathrm{N}^-_G(t)} x_{s,t} &= 1 && \text{for all } t\in\mathcal{T}\cup\mathcal{V} \quad (12)\\
\sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t} &\le x_{s,t} && \text{for all } t\in\mathcal{T}, s\in\mathrm{N}^-_G(t) \quad (4)\\
e_s &\le f^0_s && \text{for all } s\in\mathcal{V} \quad (5)\\
0 &\le e_s - \sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t}f^{\mathrm{d}}_{s,r} && \text{for all } t\in\mathcal{T}, s\in\mathrm{N}^-_G(t) \quad (6)\\
e_t &\le 1-f^{\mathrm{t}}_t-\sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t}f^{\mathrm{d}}_{r,t} && \text{for all } t\in\mathcal{T}, s\in\mathrm{N}^-_G(t) \quad (7)\\
e_t &\le e_s - x_{s,t}\left(f^{\mathrm{d}}_{s,t}+f^{\mathrm{t}}_t\right)-\sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t}\left(f^{\mathrm{d}}_{s,r}+f^{\mathrm{t}}_r+f^{\mathrm{d}}_{r,t}-f^{\mathrm{d}}_{s,t}\right)+(1-x_{s,t})\\
& && \text{for all } t\in\mathcal{T}, s\in\mathrm{N}^-_G(t) \quad (8)\\
x_{s,t} &\in \{0,1\} && \text{for all } (s,t)\in A \quad (9)\\
z_{s,r,t} &\in \{0,1\} && \text{for all } t\in\mathcal{T}, s\in\mathrm{N}^-_G(t), r\in\mathcal{R}_{s,t} \quad (10)\\
e_s &\in [0,1] && \text{for all } s\in V\setminus\{d^{\mathrm{s}},d^{\mathrm{e}}\} \quad (11)
\end{aligned}
$$

3 (LMILP)

$$
\begin{aligned}
\min & \sum_{s\in\mathcal{V}} \sum_{t\in \mathrm{N}^{+}_{\overline{G}}(s)\setminus\{d^{\mathrm{e}}\}} x_{s,t}c^{\mathrm{v}} + \sum_{s\in P}\sum_{t\in \mathrm{N}^{+}_{\overline{G}}(s)} x_{s,t}c^{\mathrm{t}}_t \\
& + \sum_{t\in\mathcal{T}\cup P}\sum_{s\in \mathrm{N}^{-}_{\overline{G}}(t)} \left[x_{s,t}\left(c^{\mathrm{d}}_{s,t}+c^{\mathrm{t}}_t\right) + \sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t}\left(c^{\mathrm{d}}_{s,r}+c^{\mathrm{d}}_{r,t}-c^{\mathrm{d}}_{s,t}\right)\right] && \text{(LMILP)}\\
\text{s.t.} & \sum_{t\in \mathrm{N}^{-}_{\overline{G}}(s)} x_{t,s} = \sum_{t\in \mathrm{N}^{+}_{\overline{G}}(s)} x_{s,t} && \text{for all } s\in\overline{V}\setminus\{d^{\mathrm{s}},d^{\mathrm{e}}\} && (13)\\
& \sum_{s\in \mathrm{N}^{-}_{\overline{G}}(t)} x_{s,t} = 1 && \text{for all } t\in\mathcal{T}\cup\mathcal{V} && (14)\\
& \sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t} \le x_{s,t} && \text{for all } t\in\mathcal{T}\cup P, s\in \mathrm{N}^{-}_{\overline{G}}(t)\setminus P && (15)\\
& e_s \le f^0_s && \text{for all } s\in\mathcal{V} && (16)\\
& 0\le e_s - \sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t}f^{\mathrm{d}}_{s,r} && \text{for all } t\in\mathcal{T}\cup P, s\in \mathrm{N}^{-}_{\overline{G}}(t)\setminus P && (17)\\
& e_t \le 1 - f^{\mathrm{t}}_t - \sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t}f^{\mathrm{d}}_{r,t} && \text{for all } t\in\mathcal{T}\cup P, s\in \mathrm{N}^{-}_{\overline{G}}(t)\setminus P && (18)\\
& e_t \le e_s - x_{s,t}\left(f^{\mathrm{d}}_{s,t}+f^{\mathrm{t}}_t\right) - \sum_{r\in\mathcal{R}_{s,t}} z_{s,r,t}\left(f^{\mathrm{d}}_{s,r}+f^{\mathrm{t}}_r+f^{\mathrm{d}}_{r,t}-f^{\mathrm{d}}_{s,t}\right) + (1-x_{s,t})\\
& && \text{for all } t\in\mathcal{T}\cup P, s\in \mathrm{N}^{-}_{\overline{G}}(t)\setminus P && (19)\\
& e_t \le e_s - x_{s,t}f^{\mathrm{t}}_t + (1-x_{s,t}) && \text{for all } s\in P, t\in \mathrm{N}^{+}_{\overline{G}}(s) && (20)\\
& x_{s,t}\in\{0,1\} && \text{for all } (s,t)\in\overline{A} && (21)\\
& z_{s,r,t}\in\{0,1\} && \text{for all } t\in\mathcal{T}\cup P, s\in \mathrm{N}^{-}_{\overline{G}}(t)\setminus P, r\in\mathcal{R}_{s,t} && (22)\\
& e_s\in[0,1] && \text{for all } s\in\overline{V}\setminus\{d^{\mathrm{s}},d^{\mathrm{e}}\} && (23)
\end{aligned}
$$

Bibliography

[Adl14] Jonathan D. Adler. *Routing and Scheduling of Electric and Alternative-Fuel Vehicles*. PhD thesis, Arizona State University, 2014.

[BCG87] A. A. Bertossi, P. Carraresi, and G. Gallo. On some matching problems arising in vehicle scheduling models. *Networks*, 17.3:271–281, 1987.

[BK09] Stefan Bunte and Natalie Kliewer. An overview on vehicle scheduling models. *Public Transport*, 1.4:299–317, 2009.

[CL07] J.-F. Cordeau and G. Laporte. The dial-a-ride problem: Models and algorithms. *Annals of Operations Research*, 153.1:29–46, 2007.

[DF54] G. B. Dantzig and D. R. Fulkerson. Minimizing the number of tankers to meet a fixed schedule. *Naval Research Logistics Quarterly*, 1.3:217–222, 1954.

[DP95] J. Daduna and J. Pinto Paixão. Vehicle scheduling for public mass transit - an overview. *Computer-Aided Transit Scheduling*, 430:76–90, 1995.

[EMH12] Sevgi Erdoğan and Elise Miller-Hooks. A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48.1:100–114, 2012.

[FP95] R. Freling and J. Pinto Paixão. Vehicle scheduling with time constraint. *Computer-Aided Transit Scheduling*, 430:130–144, 1995.

[Hau15] Jan Hauser. Amerika schaltet auf autopilot. *Frankfurter Allgemeine Zeitung*, 2015.

[HZ80] G. Y. Handler and I. Zang. A dual algorithm for the constrained shortest path problem. *Networks*, 10.4:293–309, 1980.

[ID05] S. Irnich and G. Desaulniers. *Column Generation*, chapter Shortest Path Problems with Resource Constraints, page 33–65. Boston, MA: Springer US, 2005.

[Kai16] Marcus Kaiser. Optimal integration of autonomous vehicles in car sharing: A decomposition approach in consideration of multimodal transport. Master's thesis, Technische Universität München, 2016.

[KK16] Marcus Kaiser and Martin Knoll. Optimal integration of autonomous vehicles in car sharing: Overview and basic models. Master's thesis, Technische Universität München, 2016. part 1.

[Kno16] Martin Knoll. Optimal integration of autonomous vehicles in car sharing: A decomposition approach and fastening heuristics. Master's thesis, Technische Universität München, 2016.

[LK81] J. K. Lenstra and A. H. G. Rinnooy Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11.2:221–227, 1981.

[nFORT14] Ángel Felipe, M. Teresa Ortuño, Giovanni Righini, and Gregorio Tirado. A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. *Transportation Research*, 71:111–128, 2014.

[Raf83] S. Raff. Routing and scheduling of vehicles and crews: The state of the art. *Computers and Operations Research*, 10.2:63–211, 1983.

[WLR+16] M. Wen, E. Linde, S. Ropke, P. Mirchandani, and A. Larsen. An adaptive large neighborhood search heuristic for the electric vehicle scheduling problem. *Computers and Operations Research*, 76:73–83, 2016.

List of Corrections