A Rolling Horizon Heuristic with Optimality Guarantee for an On-Demand Vehicle Scheduling Problem

Johann Hartleb
Rotterdam School of Management, Erasmus University Rotterdam, The Netherlands
Institute for Road and Transport Science, University of Stuttgart, Germany
hartleb@rsm.nl

Marie Schmidt
Rotterdam School of Management, Erasmus University Rotterdam, The Netherlands
schmidt2@rsm.nl

Abstract

We consider a basic vehicle scheduling problem that arises in the context of travel demand models: Given demanded vehicle trips, what is the minimal number of vehicles needed to fulfill the demand? In this paper, we model the vehicle scheduling problem as a network flow problem. Since instances arising in the context of travel demand models are often so big that the network flow model becomes intractable, we propose using a rolling horizon heuristic to split huge problem instances into smaller subproblems and solve them independently to optimality. By letting the horizons of the subproblems overlap, it is possible to look ahead to the demand of the next subproblem. We prove that composing the solutions of the subproblems yields an optimal solution to the whole problem if the overlap of the horizons is sufficiently large. Our experiments show that this approach is not only suitable for solving extremely large instances that are intractable as a whole, but it is also possible to decrease the solution time for large instances compared to a comprehensive approach.

2012 ACM Subject Classification Applied computing → Transportation; Theory of computation → Network flows; Mathematics of computing → Network flows

Keywords and phrases Rolling Horizon Heuristic, Vehicle Scheduling, Network Flow

Digital Object Identifier 10.4230/OASIcs.ATMOS.2020.15

1 Introduction

On-demand transport services are becoming more and more popular among travelers and they have the potential to replace a significant part of the traditional public transport services in the near future. To be able to react on and regulate such services in a meaningful way, it is important for infrastructure managers and local authorities to model and estimate the impact of on-demand services on the utilization of streets. Recently, many microscopic approaches ([11], [2], [6]) have been proposed to model on-demand services. These rely on simulation of individual agents to obtain a virtual traffic volume and estimate the impact of on-demand vehicles on the infrastructure. In contrast to that, macroscopic approaches such as [13] model vehicle and traveler movements as flows to estimate the utilization of streets.

In this paper, we focus on macroscopic approaches and discuss a simple vehicle scheduling model for on-demand vehicles: Given demanded vehicle trips, what is the minimal number of vehicles needed to fulfill the demand? The resulting vehicle schedule describes vehicle itineraries and yields both the required size of the vehicle fleet and the positions of the vehicles over time. With this information, the utilization of streets can be estimated.

Most existing vehicle scheduling approaches are developed for operational purposes to find an assignment of vehicles to planned trips ([7], [5], [3]). Recent vehicle scheduling approaches focus on the integration of further operational aspects, for example, the integration of

20th Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2020).
Editors: Dennis Huisman and Christos D. Zaroliagis; Article No. 15; pp. 15:1–15:18
OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

related planning steps ([14], [4]) or the integration of recharging strategies of battery electric vehicles ([16], [12]). Compared to these approaches, the application of vehicle scheduling to estimate the impact of on-demand services in macroscopic models brings two differences: (1) The demanded vehicle trips are not planned trips but correspond to expected passenger demand in a macroscopic model. Hence, they might be of fractional value. (2) Compared to scheduled public transport, the amount of on-demand vehicle trips can be extremely large. Especially the second difference makes many existing optimization approaches unsuitable as corresponding problems easily exceed the size of tractable instances. In [10], the vehicle scheduling problem is modeled as a network flow problem that computes the size of the necessary vehicle fleet and their itineraries. Due to the large instances in realistic applications, [10] solve the vehicle scheduling model with a simple heuristic approach that constructs the vehicle schedule chronologically. While this approach scales well also with very large instances, no guarantee on the solution quality is given. It remains unknown whether the found vehicle schedule is optimal or how far it is from an optimal solution.

The contribution of this paper is a rolling horizon approach to solve the vehicle scheduling model introduced in [10] to optimality. The rolling horizon approach is a heuristic that splits instances into tractable subproblems and solves them independently. By enforcing overlap of the horizons of these subproblems, it is possible to look ahead to the demand of the next horizon and include that information while solving the current subproblem. As a consequence, the decisions taken in the current subproblem are well suited for the next subproblem and the overall solution quality can be improved. For a sufficient length of the overlap, we prove that a globally optimal solution can be found by composing the locally optimal solutions for the horizons. In numerical experiments we could solve intractable instances from [10] to optimality. Furthermore, we could show that using the rolling horizon approach can bring a speed-up in solution time for certain instance size compared to a comprehensive approach.

The remainder of this paper is structured as follows. In Section 2, the vehicle scheduling problem from [10] is introduced and in Section 3 we describe the rolling horizon heuristic in detail as our proposed solution approach. For a sufficiently long overlap of the horizons, we provide an optimality guarantee for the rolling horizon heuristic in Section 4. By modifying the formulation, we can strengthen the conditions for optimality. In Section 5, we show in numerical experiments that using the rolling horizon heuristic can help to speed up the solution process for large instances and Section 6 concludes the paper.

2 Problem setting

The problem discussed in [10] is to find a minimal vehicle schedule, a feasible routing of a minimal number of vehicles meeting all given demand. A macroscopic model is considered in which neither the demand nor the resulting size of the vehicle fleet need to be integer.

In this setting, the considered time frame and observation area are discretized into a set of *time intervals* T and a set of *traffic zones* Z. The *distance* $\delta_{z_o z_d}$ between two zones is given as multiples of time intervals, i.e., $\delta_{z_o z_d} = n$ means that driving from zone z_o to z_d can be done in n time intervals. It is assumed that vehicle trips within a zone can be performed in one time interval, i.e., $\delta_{zz} = 1 \; \forall z \in Z$. The passenger *demand* is given aggregated to demanded vehicle trips $d_{z_o z_d t}$ between origin z_o and destination zone z_d, starting at the beginning of time interval t. The trips end at the beginning of time interval $t + \delta_{z_o z_d}$, determined by the distance between origin and destination zone and the start time. Vehicle trips either correspond to demanded person trips or, in applications with trip pooling, comprise multiple person trips. For carsharing or ridesharing applications, we assume that the pooling of person trips was done in a preceding step, for example, by the approach of [8].

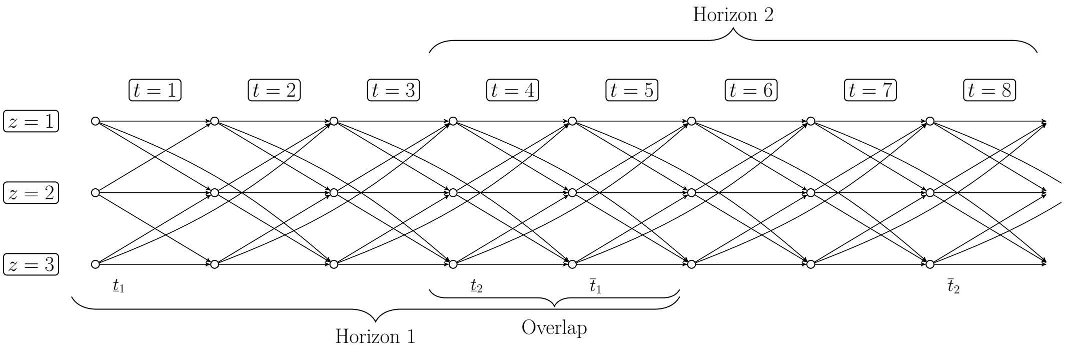


Figure 1 Instance with 3 zones and 8 time intervals. A possible assignment of time intervals to two overlapping horizons $\{\underline{t}_1, \dots, \overline{t}_1\} = \{1, \dots, 5\}$ and $\{\underline{t}_2, \dots, \overline{t}_2\} = \{4, \dots, 8\}$ is indicated.

We denote an instance consisting of a set of zones Z, a set of time intervals T, a distance function $\delta_{z_o z_d}$ and aggregated demand $d_{z_o z_d t}$ by $\mathcal{I} = (Z, T, \delta, d)$. For an instance $\mathcal{I}$, the aim is to compute a vehicle schedule with a minimal number of vehicles that meet the demand. The fleet size and the vehicle routes in the schedule can be used to estimate the infrastructure utilization. Deadheading trips are allowed to relocate vehicles, and vehicles waiting in a traffic zone can be modeled as empty trips within a zone.

2.1 Network representation

This problem can be visualized in a time-space network, see Figure 1 for an instance with 3 zones and 8 time intervals. For each combination of z and t in the planning horizon, we introduce a node (z, t). These nodes are displayed in a grid structure with time intervals t on the horizontal and traffic zones z on the vertical. Each node (z, t) represents a traffic zone z at the *beginning of a time interval* t. The distance $\delta_{z_o z_d}$ between two zones z_o and z_d is represented by the horizontal length of the arcs between the corresponding nodes. An arc (z_o, z_d, t) between two nodes represents a possible vehicle trip between two zones z_o and z_d, starting at time interval t. The arrival time $t + \delta_{z_o, z_d}$ results from the start time t and the distance δ_{z_o, z_d}. For readability, we omit the arrival time in the notation of an arc. The demanded vehicle trips $d_{z_o z_d t}$ are modeled as lower bounds on the arc (z_o, z_d, t). A minimal flow in this network corresponds to a vehicle schedule with a minimal number of vehicles.

2.2 Model

In [10] it is proposed to model the vehicle scheduling problem as a network flow problem ([1], [15], [3]) on the time-space network as introduced in Section 2.1. They formulate the following linear program VS to find vehicle tours with a minimal number of vehicles.

$$\begin{aligned}
\min \quad & \sum_{z_o \in Z} \sum_{z_d \in Z} f_{z_o z_d 1} && && \text{(1a)}\\
\text{s.t.} \quad & f_{z_o z_d t} \geq d_{z_o z_d t} && \forall z_o, z_d \in Z,\ \forall t \in T && \text{(1b)}\\
& \sum_{\substack{z_o \in Z:\\ t - \delta_{z_o z} \geq 1}} f_{z_o z (t - \delta_{z_o z})} = \sum_{z_d \in Z} f_{z z_d t} && \forall z \in Z,\ \forall t \in T \setminus \{1\} && \text{(1c)}\\
& f_{z_o z_d t} \in \mathbb{R}_+ && \forall z_o, z_d \in Z,\ \forall t \in T && \text{(1d)}
\end{aligned}$$

The flow variables $f_{z_o z_d t} \in \mathbb{R}_+$ denote the number of vehicle trips from zone z_o to zone z_d, starting at the beginning of time interval t. The objective (1a) is to minimize the total number of vehicles, expressed by the number of vehicle trips starting in the first time interval. The resulting number of vehicles for a flow f is also referred to as flow value $|f|$. The first set of constraints (1b) ensures that the demand is satisfied. If $f_{z_o z_d t} > d_{z_o z_d t}$, that is, if there are more vehicle trips than demanded, this can be interpreted as empty trips for vehicle relocation or waiting in a traffic zone if $z_o = z_d$. To obtain a feasible vehicle flow, the second set of constraints (1c) requires that the number of vehicles is preserved in each zone z at the beginning of each time interval t. The domains of the flow variables f in (1d) show that the vehicle trips do not need to be integer valued.

2.3 Difficulty of problem

The problem VS is a continuous linear program and can therefore be solved efficiently with available solvers for moderately sized instances. The coefficient matrix is totally unimodular, hence, the problem of finding integer flows is polynomially solvable for integer demand $d_{z_o z_d t} \in \mathbb{Z}$ (see [9], problem [ND37], second comment). However, to determine the impact of on-demand vehicles on traffic and infrastructure in realistic cases of application, the number of time intervals and traffic zones may be enormous and yield intractable instances.

[10] discuss an application instance for the city area of Stuttgart. In this instance, the observation area is separated into $|Z| = 1175$ traffic zones and the time frame of one full day is segmented in $|T| = 96$ time intervals of 15 min. The numbers of variables and constraints of this instance is in the order of 10^8 and the corresponding optimization model could not be build with the general purpose solver Fico Xpress 8.8 on a laptop with 32GB RAM[1].

To handle extremely large instance sizes, [10] propose a simple heuristic that chronologically processes the nodes in the network and gradually constructs a vehicle schedule. By backtracking and repairing the vehicle schedule during construction, they can achieve good solutions for huge instances. However, the algorithm does not provide an approximation guarantee for the constructed vehicle schedules (see [17] for more information on approximation algorithms). That means, it cannot be guaranteed how close the solution is to an optimal one. Furthermore, no optimality gap is provided by design of the algorithm.

3 Rolling horizon heuristic

In this paper, we propose using a rolling horizon heuristic to solve the model VS. The idea is to divide the considered time frame into shorter time horizons and solve one subproblem for each time horizon. The solutions to the subproblems can then be composed to a solution for the full time frame.

3.1 Generalization of linear program

To be able to compose the solutions of the subproblems to a feasible global solution, the optimization model VS from [10] is generalized. We consider an additional input of vehicles a_{zt} that become available in zone z at the beginning of time interval t. Available vehicles to be considered in one subproblem are a result of the flow fixed in previous subproblems. We

[1] Hardware: Intel® Core™ i7-6700HQ CPU with 32GB of RAM; OS: Windows 10 Enterprise 2015 64-bit

denote the generalized input by $\overline{\mathcal{I}} = (Z, T, \delta, d, a)$ and generalize the optimization program to $\overline{VS}$:

$$
\begin{aligned}
\min \quad & \sum_{z_o \in Z} \sum_{z_d \in Z} f_{z_o z_d 1} - a_{z_o 1} && && \text{(2a)}\\
\text{s.t.} \quad & f_{z_o z_d t} \geq d_{z_o z_d t} && \forall z_o, z_d \in Z,\ \forall t \in T && \text{(2b)}\\
& \sum_{\substack{z_o \in Z:\\ t-\delta_{z_o z} \geq 1}} f_{z_o z (t-\delta_{z_o z})} + a_{zt} = \sum_{z_d \in Z} f_{z z_d t} && \forall z \in Z,\ \forall t \in T \setminus \{1\} && \text{(2c)}\\
& \sum_{z_d \in Z} f_{z z_d 1} \geq a_{z1} && \forall z \in Z && \text{(2d)}\\
& f_{z_o z_d t} \in \mathbb{R}_+ && \forall z_o, z_d \in Z,\ \forall t \in T && \text{(2e)}
\end{aligned}
$$

As in the program VS, the objective (2a) is to minimize the total number of vehicles needed to serve the demand. Since the flow variables f comprise all moving vehicles (including those that are given as available vehicles), available vehicles a in the first time interval are subtracted in the objective. This corresponds to minimizing the number of *additional vehicles* needed for serving the demand. The constraint (2b) ensuring that all demand is satisfied remains unchanged and is the same as constraint (1b). It is necessary to generalize the flow conservation constraints (2c) by treating available vehicles a_{zt} as incoming flow in nodes (z, t). Furthermore, an additional set of constraints (2d) ensures that the outgoing flow in the first interval considers all available vehicles a since this time interval is not covered in the flow conservation constraints (2c). Note, that for $a = 0$ the program $\overline{VS}$ coincides with the program VS.

3.2 Overlapping horizons

The idea of the rolling horizon heuristic is to divide the time frame into horizons and solve one subproblem for each horizon. By letting the horizons overlap it is further possible to look ahead to the demand of the next horizon. That means, the demand in the overlap with the next horizon is considered in the subproblem of the current horizon. However, the vehicle trips to satisfy this demand is not yet fixed in the solution to the current subproblem, but in the solution to the next subproblem.

The longer the overlap, the more demand can be considered, which allows to move vehicles to positions that are well-suited for meeting demand in the next horizon. These positions of vehicles are considered as available vehicles a_{zt} in the subproblem for the next horizon.

A possible division of a time frame into two overlapping horizons is indicated in the example network in Figure 1. In this example, the first horizon $\{1, \ldots, 5\}$ spans over the first five time intervals. In the first subproblem, all demand starting in these five intervals is considered. However, only the flow starting before the overlap, that is, starting in the first three time intervals is fixed in the solution of the first subproblem. The flow in the overlap is fixed by solving the subproblem of the second horizon $\{4, \ldots, 8\}$.

3.3 Algorithm

With the generalized optimization model $\overline{VS}$ we can define the rolling horizon heuristic. Its general idea is to solve the problem for smaller horizons that may be overlapping and compose the partial solutions to a solution for the full problem. Let h denote the number of time intervals in each horizon and let o denote the number of overlapping time intervals in the rolling horizon heuristic. Naturally, we require $0 \leq o < h$.

Algorithm 1 *Rolling horizon heuristic.*

```
1  Input: Instance $\mathcal{I} = (Z, T, \delta, d)$, length of horizon $h$, horizon overlap $o$
2  Output: Vehicle flow $f$
   # Initialize first horizon from time interval 1 to $h$, initialize variables for
     available vehicles $a$ and flow $f$ with 0;
3  Initialize: $i \leftarrow 1$, $\underline{t}_i \leftarrow 1$, $\bar{t}_i \leftarrow h$, $a_{zt} \leftarrow 0 \quad \forall z \in Z, t \in T$,
     $f_{z_o z_d t} \leftarrow 0 \quad \forall z_o, z_d \in Z, t \in T$;
   # Iterate through horizons until end of time frame is reached;
4  while $\bar{t}_i < |T|$ do
       # Solve the subproblem corresponding to the current horizon $i$ and get flow $f$;
5      $f \leftarrow$ Solve subproblem$(\mathcal{I}, \{\underline{t}_i, \dots, \bar{t}_i\}, a, f)$;
       # Update number of available vehicles from fixed flow for next horizons;
6      $a_{zt} \leftarrow \sum_{z_o \in Z:\ 1 \leq t - \delta_{z_o z} < \bar{t}_i - o} f_{z_o z (t - \delta_{z_o z})} \quad \forall z \in Z,\ t \in \{\bar{t}_i - o, \dots, \bar{t}_i - o + \max \delta_{z_o z_d} - 1\}$;
       # Update bounds of next horizon and goto next horizon by increasing counter $i$;
7      $\underline{t}_{i+1} \leftarrow \underline{t}_i + h - o$;
8      $\bar{t}_{i+1} \leftarrow \bar{t}_i + h$;
9      $i \leftarrow i + 1$;
   # When reached end of time frame, truncate last horizon at $|T|$;
10 $\bar{t} \leftarrow |T|$;
   # Solve the subproblem corresponding to the last horizon $i$ and get flow $f$;
11 $f \leftarrow$ Solve subproblem$(\mathcal{I}, \{\underline{t}_i, \dots, \bar{t}_i\}, a, f)$;
12 Return $f$;
```

The pseudocode for the heuristic is presented in Algorithm 1. This algorithm processes one horizon $\{\underline{t}_i, \dots, \bar{t}_i\}$ after another (Line 4), with the first horizon starting in the first time interval (Line 3). The horizons span h time intervals (Line 8) and each two consecutive horizons have an overlap of o time intervals (Line 7). For each iteration, the subproblem corresponding to the current horizon is called (Line 5 & 11), which is explained in detail in Algorithm 2. Afterwards, the available vehicles are updated to communicate information from the solution of one subproblem to the next (Line 6). Available vehicles are a mean to model fixed vehicle trips that started before the overlap. By definition, these trips end at latest $\max \delta_{z_o z_d} - 1$ time intervals after the beginning of the overlap.

Algorithm 2 *Solve subproblem.*

```
Input: Instance $\mathcal{I}$, horizon $\{\underline{t}, \dots, \bar{t}\}$, available vehicles $a$, (partial) vehicle flow $f$
Output: Updated vehicle flow $f$
# Initialize sub-instance $\overline{\mathcal{I}}$ constrained to the horizon;
Initialize: $T' \leftarrow \{\underline{t}, \dots, \bar{t}\}$, $\overline{\mathcal{I}} \leftarrow (Z, T', \delta|_{T'}, d|_{T'}, a|_{T'})$;
do
# Solve generalized optimization problem and get optimal flow $f'$ for horizon $T'$;
$f' \leftarrow \overline{VS}(\overline{\mathcal{I}})$;
# Update total vehicle flow with solution from subproblem;
$f_{z_o z_d t} \leftarrow f'_{z_o z_d t} \quad \forall z_o, z_d \in Z,\ t \in T'$;
# Add additional vehicles to time intervals before the horizon to conserve flow;
$f_{zzt} \leftarrow f_{zzt} + \min\{\sum_{z_d \in Z} f'_{z z_d \underline{t}} - a_{z\underline{t}}, 0\} \quad \forall z \in Z,\ t < \underline{t}$;
Return $f$;
```

In Algorithm 2, the optimization model $\overline{VS}$ is called (Line 5) to find an optimal vehicle schedule for the subinstance $\overline{\mathcal{I}}$ that is constrained to the current horizon (Line 3). After each optimization call, the the total flow f is updated with the partial solution found (Line 6). Note, that this overwrites the vehicle flow in the overlap with the previous horizon. This procedure is equivalent to considering all demand in the current horizon, but just fixing the flow before the overlap, as described in Section 3.2. The flow in the overlap is discarded and then fixed with the solution of the next subproblem. If more vehicles were necessary to serve the demand in the current horizon than in the previous horizons, additional vehicles are added to the flow f (Line 7). This can be interpreted as introducing waiting vehicles in a traffic zone during all previous horizons.

Algorithm 1 keeps the structure of the rolling horizon heuristic, and for each horizon the total flow is extended by the vehicle flow found in Algorithm 2. At the end, Algorithm 1 returns a vehicle flow for the whole time frame that is composed by the optimal partial flows for the horizons.

4 Quality of the solution

In this section, we prove that the vehicle schedules found by the rolling horizon heuristic in Algorithm 1 are optimal for certain choices of the overlap o. We start with the argument that the rolling horizon heuristic finds a feasible solution.

▶ **Definition 1.** *A flow f is called* feasible *for an instance $\mathcal{I}$ if it satisfies all demand and fulfills the flow conservation in each vertex, i.e., if constraints (1b) and (1c) hold.*

Since the partial solutions are optimal and hence feasible solutions to the flow problems per horizon, the demand is satisfied by the composed vehicle flow. By carrying over vehicles to next horizon with the help of available vehicles a, and by introducing additional waiting vehicles in previous horizons, the flow conservation holds.

▶ **Observation 2.** *Hence, the composed vehicle schedule found by the rolling horizon algorithm is a feasible vehicle schedule for the whole time frame of an instance $\mathcal{I}$.*

▶ **Theorem 3.** *The rolling horizon heuristic finds an optimal solution for an instance $\mathcal{I}$ if*

$$o \geq 2 \cdot \max_{z_o, z_d \in Z} \delta_{z_o z_d} - 1.$$

Sketch of proof. The main idea of the proof for Theorem 3 is simple: Since the overlap is long enough, any vehicle trip that was fixed in the solution of a previous horizon can be corrected by another vehicle trip in any desired direction, if necessary. We prove Theorem 3 by induction and start with the observation that the flow f^1 for the first horizon is optimal.

Next, we consider a flow for the first i horizons, denoted by f^i, and assume that it is optimal for the first i horizons. That means, it is optimal for the instance $\mathcal{I} = (Z, T^i, \delta, d)$ with restricted time frame $T^i := \{1, \ldots, \overline{t}_i\}$. Hence, the vehicle schedule f^i is a feasible flow that meets all demand starting at latest at time interval $\overline{t}_i$, the end of the i^{th} horizon. Remember that the flow in the overlap $\{\overline{t}_i - o, \ldots, \overline{t}_i\}$ is not fixed yet but will be overwritten in the next iteration of the rolling horizon heuristic. The key is to show that fixing vehicle trips that start up to $t < \overline{t}_i - o$, the beginning of the overlap, does not prevent the rolling horizon heuristic to find an optimal solution f^{i+1} for the first $i+1$ horizons if $o \geq 2 \cdot \max \delta_{z_o z_d} - 1$.

To do the induction step, we first consider the demand in the overlap, and, afterwards, vehicles that are not necessary to meet demand in the overlap. Since f^i is a feasible flow for the first i horizons, all demand in the overlap is satisfied. Of course, this demand is also met in any feasible solution for $i+1$ horizons. With some adaptions on the flow in the overlap, it can be shown that any optimal solution f^i for i horizons can be extended by any optimal solution after the overlap. Since these adaptions require extensive notation, We give a detailed proof for this in Appendix A. The underlying idea of the proof is, that it is not important *which* vehicles meet the demand, but it is ensured that *sufficient* vehicles are available to meet the demand in the overlap.

For the remaining vehicles, we focus on vehicle trips in f^i that start before, and end at or after the beginning of the overlap $\bar{t}_i - o$. Vehicle trips that end earlier do not interfere with the next horizon, and vehicle trips that start later are overwritten by the solution for the next horizon. Hence, these trips in f^i do not restrict the solution for the $(i+1)^{\text{st}}$ horizon.

The trips under consideration start before the overlap, hence at $t \leq \bar{t}_i - o - 1$ and end at latest at $t \leq \bar{t}_i - o - 1 + \max \delta_{z_o z_d}$. Relocating the vehicles that have executed these trips to an arbitrary traffic zone from their current location takes at most another $\max \delta_{z_o z_d}$ time intervals. Hence, these vehicles are able to meet demand starting from any zone at time interval $t \leq \bar{t}_i - o - 1 + 2 \cdot \max \delta_{z_o z_d}$. For $o \geq 2 \cdot \max \delta_{z_o z_d} - 1$, the vehicles are able to meet demand just after the overlap at $t \leq \bar{t}_i$.

Together, this shows that fixing vehicle trips starting before the overlap in the solution of one subproblem and carrying this decision over to the next subproblem by the means of available vehicles, does not prevent finding a globally optimal solution. There are sufficient vehicles to meet demand starting in the overlap and the remaining vehicles can be relocated to any zone to meet demand after the overlap. ◀

Theorem 3 gives a lower bound on the overlap to ensure finding optimal solutions. It is further possible to show that this lower bound is minimal.

▶ **Lemma 4.** *For $o < 2 \cdot \max \delta_{z_o z_d} - 1$, optimality of the rolling horizon heuristic cannot be guaranteed.*

Proof. We consider an example instance containing two zones and five intervals. The maximum distance between two zones is $\delta_{12} = \delta_{21} = 2$. There is only demand of 1.0 vehicle trips within the first zone starting in time intervals 1 and 5, i.e., $d_{111} = d_{115} = 1.0$ and $d_{z_o z_d t} = 0.0$ otherwise. An outline of the underlying network can be found in Figure 2a.

One optimal solution is to use one vehicle that stays within the first zone all the time and satisfies the demand during the first and the fifth time interval. This solution is

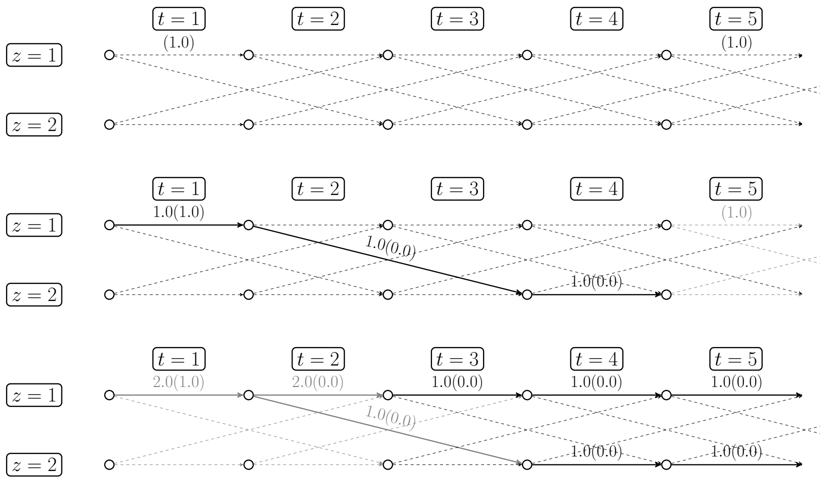
$$f_{11t} = 1.0 \ \forall t \in \{1, \dots, 5\} \text{ and } f_{z_o z_d t} = 0.0 \text{ otherwise,}$$

with an objective value of $|f| = 1.0$. Applying the rolling horizon heuristic to this instance with a horizon length of $h = 4$ and an overlap of $o = 2 = 2 \max \delta_{z_o z_d} - 2$, optimality cannot be guaranteed.

When considering the first horizon $\{1, \dots, 4\}$, demand $d_{115} = 1.0$ lies outside the horizon and is not considered yet. Therefore, routing 1.0 vehicles to the second zone after meeting demand d_{111} is an optimal solution to the first subproblem. This partial solution

$$f_{111} = f_{122} = f_{224} = 1.0 \text{ and } f_{z_o z_d t} = 0.0 \text{ otherwise}$$

with an objective value of $|f| = 1.0$ is depicted in Figure 2b. Given this partial solution, it is impossible to satisfy the demand in the fifth time interval with the same vehicles. When considering the second horizon $\{3, \dots, 5\}$, the vehicle trip f_{122} to the second zone cannot be reverted since it started before the overlap, and it is impossible to send the 1.0 vehicles



(a) Instance: Network with 2 zones and 5 time intervals, indicated demand and visualized distances.

(b) An optimal solution to first horizon $\{1, 2, 3, 4\}$. This prevents finding a globally optimal solution.

(c) An optimal solution to second horizon $\{3, 4, 5\}$. Previously fixed flow enforces use of additional vehicles.

Figure 2 Example network with 2 zones and 5 time intervals and suboptimal solution found by the rolling horizon heuristic for an overlap of $o = 2$ time intervals. Dashed lines indicate potential vehicle trips, thick edges positive flow, and grey lines and vertices are outside of the considered horizon. Numbers in brackets on edges show demand, numbers without brackets show vehicle flow.

back to the first zone in time. In this case, another 1.0 vehicles have to be added to satisfy demand d_{115} in the fifth interval, yielding a suboptimal global solution. This solution

$$f_{111} = 2.0,\ \ f_{11t} = 1.0\ \forall 2 \leq t \leq 5,\ \ f_{122} = f_{224} = f_{225} = 1.0 \text{ and } f_{z_o z_d t} = 0.0 \text{ otherwise}$$

has an objective value of $|f| = 2.0$ and can be seen in Figure 2c.

Note, that this example can be generalized to provide a counterexample for any maximal distance between two zones. Consider a network with the same pattern: two zones and a distance of $\delta_{12} = \delta_{21}$ between these two zones. Define the demand by $d_{111} = d_{11(2 \max \delta_{z_o z_d}+1)} = 1.0$ and $d_{z_o z_d t} = 0.0$ otherwise. Then, the rolling horizon with a setting of $h = 2 \max \delta_{z_o z_d}$ and $o = 2 \max \delta_{z_o z_d} - 2$ might fail to find the optimal solution with the same argumentation. This generalization shows that a choice of $o = 2 \max \delta_{z_o z_d} - 1$ is indeed the smallest value for the horizon overlap that ensures an optimal solution for Algorithm 1. ◀

The counter example in the proof of Lemma 4 abuses the fact that an unreasonable decision to route an empty vehicle to another zone can appear in an optimal solution. By preventing this kind of unreasonable vehicle trip, the condition for the optimality guarantee of the rolling horizon heuristic can be strengthened. In this context, a vehicle trip to a different zone is considered unreasonable if it does not satisfy any demand $d_{z_o z_d t}$ or if it is not performed to satisfy any demand in the zone of its destination. Staying within zones is never considered to be unreasonable as it is also used to model waiting vehicles during a time interval. This is formalized in the following definition.

▶ **Definition 5.** *A flow f is called **unreasonable** if for an arc (z_o, z_d, t) with $z_o \neq z_d$ none of the two conditions holds*

1. *Flow $f_{z_o z_d t}$ satisfies demand $d_{z_o z_d t}$, i.e., $f_{z_o z_d t} = d_{z_o z_d t}$.*
2. *Flow $f_{z_o z_d t}$ is performed to have enough vehicles available to meet demand $d_{z_d z t'}$ starting*

in zone z_d at $t' := t + \delta_{z_o z_d}$, and otherwise there were too few vehicles, i.e.,

$$f_{z_o z_d t} + \sum_{\substack{z_o \neq z \in Z:\\ t' - \delta_{z z_d} \geq 1}} f_{z z_d (t' - \delta_{z z_d})} = \sum_{z \in Z} d_{z_d z t'}.$$

As a consequence, if it is ensured that no flow is unreasonable, all vehicles stay in the destination zone z_d of their last satisfied demand $d_{z_o z_d t}$ unless they are needed to satisfy demand. In particular, sending 1.0 vehicles from the first to the second zone in the counter example is unreasonable. Preventing unreasonable flow helps to improve the condition for an optimality guarantee.

▶ **Theorem 6.** *If it is ensured in each iteration that no vehicle flow is unreasonable, the rolling horizon heuristic finds an optimal solution for the whole time frame if*

$$o \geq \max_{z_o, z_d \in Z} \delta_{z_o z_d}.$$

Sketch of proof. The idea for the proof of Theorem 6 follows the structure of the one for Theorem 3. In this case, the vehicle trips that start before the overlap end at the beginning of the overlap $t = \bar{t}_i - o$, unless they meet demand. Then, after relocating them, the vehicles are able to meet demand at $t \leq \bar{t}_i - o + \max \delta_{z_o z_d}$, i.e., at $t \leq \bar{t}_i$ for $o \geq \max \delta_{z_o z_d}$.

Since the flow is not unreasonable, the fixed vehicle trips that end after $\bar{t}_i - o$ either meet demand or relocate vehicles to meet demand in the zone of destination. Hence, they are not a restriction to finding a globally optimal solution as these trips have to be performed in any feasible solution. With the same argumentation as in proof of Theorem 6, it follows that the rolling horizon heuristic finds an optimal solution under the given conditions. ◀

Again, it is possible to show that this value is minimal.

▶ **Lemma 7.** *For $o < \max \delta_{z_o z_d}$ optimality of the rolling horizon heuristic cannot be guaranteed, even if it is ensured in each iteration that no vehicle flow is unreasonable.*

Proof. To show that the rolling horizon heuristic might not be optimal if $o = \max \delta_{z_o z_d} - 1$ we again provide a counterexample. This instance has two zones and $\max \delta_{z_o z_d} + 2$ intervals, where the case of $\max \delta_{z_o z_d} = \delta_{12} = 2$ is depicted in Figure 3a. The only demand in this instance is $d_{111} = d_{22(\max \delta_{z_o z_d} + 2)} = 1.0$. An optimal solution is

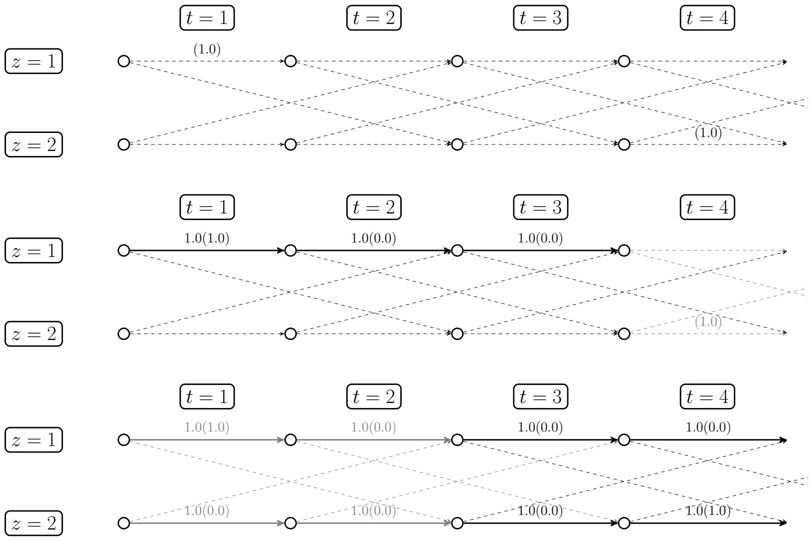
$$f_{111} = f_{122} = f_{22(\max \delta_{z_o z_d} + 2)} = 1.0 \text{ and } f_{z_o z_d t} = 0.0 \text{ otherwise}$$

with an optimal objective value of 1.0.

With the assumption of no unreasonable flow, the rolling horizon heuristic with parameter setting $h = \max \delta_{z_o z_d} + 1 = 3$ and $o = \max \delta_{z_o z_d} - 1 = 1$ cannot find an optimal solution for this instance. In the first horizon $\{1, \ldots, \max \delta_{z_o z_d} + 1\}$, demand $d_{22(\max \delta_{z_o z_d} + 2)}$ is not considered and flow $f_{111} = f_{112} = \cdots = f_{11(\max \delta_{z_o z_d} + 1)} = 1.0$ is fixed, see Figure 3b for the case $\max \delta_{z_o z_d} = \delta_{12} = 2$. In the second horizon $\{3, \ldots, \max \delta_{z_o z_d} + 2\}$ it is not possible any more to route the available vehicle from the first zone to the second zone to satisfy demand $d_{22(\max \delta_{z_o z_d} + 2)} = 1.0$. This situation can be seen in Figure 3c. It is necessary to introduce 1.0 additional vehicles in the second zone, which yields the suboptimal solution

$$f_{11t} = f_{22t} = 1.0 \; \forall t \in T \text{ and } f_{z_o z_d t} = 0.0 \text{ otherwise}$$

with the objective value 2.0. This shows that in case of no unreasonable flow the rolling horizon heuristic only is guaranteed to find an optimal solution for $o \geq \max \delta_{z_o z_d}$. ◀



(a) Instance: Network with 2 zones and 4 time intervals, indicated demand and visualized distances.

(b) An optimal solution to first horizon $\{1, 2, 3\}$. This prevents finding a globally optimal solution.

(c) An optimal solution to second horizon $\{3, 4\}$. Previously fixed flow enforces use of additional vehicles.

Figure 3 Example network with 2 zones and 4 time intervals and suboptimal solution found by the rolling horizon heuristic for an overlap of $o = 1$ time interval. Dashed lines indicate potential vehicle trips, thick edges positive flow, and grey lines and vertices are outside of the considered horizon. Numbers in brackets on edges show demand, numbers without brackets show vehicle flow.

Unreasonable flow can, for example, be prevented by using the objective function (3):

$$\sum_{z_o \in Z} \sum_{z_d \in Z} f_{z_o z_d 1} - a_{z_o 1} + \sum_{z_o \in Z} \sum_{z_d \in Z} c_{z_o z_d} \sum_{t \in T} f_{z_o z_d t}. \tag{3}$$

with artificial routing costs c. By setting $c_{zz} = 0$ and $0 < c_{z_o z_d} < \frac{1}{|T|}$ $\forall z_o \neq z_d \in Z$, waiting in a zone is always preferred to an unreasonable flow. The upper bound on $c_{z_o z_d}$ ensures that never additional vehicles are acquired to save artificial routing costs. This means, using objective (3) with that cost setting minimizes the number of vehicles and at the same time prevents unreasonable flow.

5 Numerical experiments

In their paper, [10] discuss instances with up to 10^8 vehicles trips which leads to a similar amount of variables. While it was not possible to build an optimization model VS for such huge instances with the solver FICO Xpress on a laptop with 32GB RAM, optimal solutions for these instances could be found with the rolling horizon heuristic on the same machine.

Besides the fact that huge instances become tractable, splitting the problem into subproblems can speed up the solution process for tractable instance sizes. We conduct experiments on randomly generated instances with $|T| = 96$ time intervals, a maximal distance of $\max \delta_{z_o z_d} = 10$ time intervals between zones, and a varying number of zones $|Z|$. The rolling horizon heuristic is used with the adjusted objective function (3) and a minimal overlap of $o = 10$ that ensures finding an optimal solution. For each instance size, that means, for each number of zones $|Z|$, five randomly generated instances are solved with various settings for the horizon length h. The horizon length h and the overlap o determine the number of subproblems p that need to be solved during the rolling horizon heuristic. Applying the rolling horizon heuristic with a horizon length of 96 time intervals means solving the whole problem at once and is considered as the base case.

Table 1 Solution times for varying instance sizes (number of zones $|Z|$) and length of horizon h. The top two rows indicate the length per horizon h and the corresponding number of subproblems p. The first two columns state the number of zones $|Z|$ and the resulting number of vehicle trips. The last column gives the best absolute solution time in seconds per instance size. The remaining columns show the solution times relative to the best solution time per instance size.

p		1	2	3	4	5	6	7	8	9	10	11	
h		96	53	39	32	28	25	23	21	20	19	18	
$\|Z\|$	trips	rel CPU											abs CPU [s]
20	$3.8\,10^4$	**1.0**	1.17	1.27	1.43	1.40	1.53	1.69	1.84	1.90	2.05	2.15	1.0
40	$1.5\,10^5$	1.02	1.01	**1.0**	1.10	1.16	1.23	1.25	1.36	1.37	1.42	1.65	4.3
60	$3.5\,10^5$	1.31	1.06	**1.0**	1.06	1.09	1.15	1.15	1.22	1.26	1.44	1.41	10.7
80	$6.1\,10^5$	1.42	1.11	**1.0**	1.03	1.04	1.14	1.12	1.19	1.20	1.21	1.27	22.1
100	$9.6\,10^5$	1.49	1.08	1.01	**1.0**	1.02	1.03	1.06	1.09	1.16	1.19	1.27	38.4
120	$1.4\,10^6$	1.78	1.18	**1.0**	1.00	1.10	1.04	1.12	1.16	1.15	1.21	1.20	62.2
140	$1.9\,10^6$	1.72	1.17	1.04	**1.0**	1.05	1.06	1.07	1.08	1.12	1.13	1.15	95.8
160	$2.5\,10^6$	1.76	1.18	1.03	1.03	**1.0**	1.05	1.04	1.06	1.10	1.14	1.12	142.2
180	$3.1\,10^6$	1.82	1.18	1.04	**1.0**	1.03	1.02	1.04	1.02	1.04	1.05	1.10	194.0
200	$3.8\,10^6$	1.95	1.42	1.17	1.12	1.07	**1.0**	1.04	1.09	1.05	1.07	1.11	241.7
220	$4.6\,10^6$	1.93	1.25	1.08	1.10	1.07	1.03	**1.0**	1.05	1.08	1.10	1.08	316.1
240	$5.5\,10^6$	1.81	1.19	1.02	**1.0**	1.03	1.04	1.07	1.06	1.09	1.05	1.07	437.2

Table 1 shows relative and best absolute solution times for finding a globally optimal solution, averaged over five random instances for each instance size. Both the number of trips and the average absolute solving time increase exponentially with the number of zones, indicating that large instances are hard to solve.

A value of 1.0 in the top left corner indicates that it is fastest to solve the instances with 20 zones at once, i.e., with a horizon length of $h = 96$. With decreasing length of the horizon, and thus increasing number of subproblems, the solution times increase. For example, solving the same instances by splitting them up into 11 horizons spanning 18 time intervals each, thus solving 11 (smaller) subproblems, takes more than twice as long as the fastest option.

The larger the instances, the more it pays off to solve a larger number of small subproblems instead of only few but large subproblems. With the tendency to increase further, solving instances with number of trips in the order of magnitude of 10^6 at once took almost twice as long as solving them with the rolling horizon approach in the best setting. Comparably low computation times could be achieved with various settings for the horizon length.

6 Conclusion and Outlook

6.1 Conclusion

This paper presents an alternative way to solve a simple vehicle scheduling problem as it occurs, for example, in the context of traffic estimation. The aim is to meet given demand with the least amount of vehicles possible. For certain applications such as on-demand services, the number of demanded trips can be extremely large, making real-world instances intractable.

We propose a rolling horizon heuristic to solve large instances of this problem. The principle is to split the considered time frame into small horizons and solve a vehicle scheduling problem for each horizon. For a sufficient overlap of the horizons, we prove that a solution composed by the partial solutions of the horizons is globally optimal. By introducing artificial routing costs, we could further relax the condition on the optimality criterion which makes finding optimal solutions less expensive.

In experiments we find that the rolling horizon approach has a runtime advantage over solving a full model already for moderately sized instances, which illustrates the benefit of our approach also for instances of medium size.

6.2 Outlook

The presented rolling horizon approach was motivated with and developed for the application of vehicle scheduling in macroscopic demand models. However, the underlying theory of the solution approach is more general. The vehicle scheduling problem VS is modeled as a general network flow problem on a directed cycle-free graph. Hence, the presented rolling horizon approach is also applicable to a wider set of applications that can be modeled similarly.

Furthermore, it would be interesting to investigate whether the basic idea of the proof can be adjusted to be used in an even broader range of applications. The key ingredient of the proof is that decisions do not influence the remote future, which is the case in many applications with time-space networks, for example. Therefore, it might be possible to prove that a rolling horizon solution approach is capable of finding optimal solutions in other applications as well. This could be especially interesting for applications of online optimization where information is revealed successively.

References

1 Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. *Network flows.* Working paper (Sloan School of Management); 2059-88. Cambridge, Mass.: Alfred P. Sloan School of Management, Massachusetts Institute of Technology, 1988. URL: https://dspace.mit.edu/handle/1721.1/49424.

2 Joschka Bischoff, Michal Maciejewski, and Kai Nagel. City-wide shared taxis: A simulation study in Berlin. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 275–280. IEEE, 2017.

3 Stefan Bunte and Natalia Kliewer. An overview on vehicle scheduling models. *Public Transport*, 1(4):299–317, 2009.

4 Samuela Carosi, Antonio Frangioni, Laura Galli, Leopoldo Girardi, and Giuliano Vallese. A matheuristic for integrated timetabling and vehicle scheduling. *Transportation Research Part B: Methodological*, 127:99–124, 2019.

5 A El-Azm. The minimum fleet size problem and its applications to bus scheduling. In *Computer scheduling of public transport 2*, pages 493–512, 1985.

6 Daniel J Fagnant and Kara M Kockelman. Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in Austin, Texas. *Transportation*, 45(1):143–158, 2018.

7 Brian A Foster and David M Ryan. An integer programming approach to the vehicle scheduling problem. *Journal of the Operational Research Society*, 27(2):367–384, 1976.

8 Markus Friedrich, Maximilian Hartl, and Christoph Magg. A modeling approach for matching ridesharing trips within macroscopic travel demand models. *Transportation*, 45(6):1639–1653, 2018. doi:10.1007/s11116-018-9957-5.

9 Michael R Garey and David S Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979.

10 Johann Hartleb, Markus Friedrich, and Emely Richter. Umlaufbildung für on demand-fahrzeugflotten in makroskopischen nachfragemodellen (preprint). In *HEUREKA'20: Optimierung in Verkehr und Transport*, FGSV 002/127, 2020.

11 Michael Heilig, Tim Hilgert, Nicolai Mallig, Martin Kagerbauer, and Peter Vortisch. Potentials of autonomous vehicles in a changing private transportation system – a case study in the Stuttgart region. *Transportation research procedia*, 26:13–21, 2017.

12 Tao Liu and Avishai Avi Ceder. Battery-electric transit vehicle scheduling with optimal number of stationary chargers. *Transportation Research Part C: Emerging Technologies*, 114:118–139, 2020.

13 Emely Richter, Markus Friedrich, Alexander Migl, and Johann Hartleb. Integrating ridesharing services with automated vehicles into macroscopic travel demand models. In *2019 6th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, pages 1–8. IEEE, 2019.

14 Anita Schöbel. An eigenmodel for iterative line planning, timetabling and vehicle scheduling in public transportation. *Transportation Research Part C: Emerging Technologies*, 74:348–365, 2017.

15 Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer Science & Business Media, 2003.

16 Min Wen, Esben Linde, Stefan Ropke, P Mirchandani, and Allan Larsen. An adaptive large neighborhood search heuristic for the electric vehicle scheduling problem. *Computers & Operations Research*, 76:73–83, 2016.

17 David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.

A Proof of optimality

To proof Theorem 3, some additional definitions and observations are helpful. From Observation 2 we know that a composed solution for an instance $\mathcal{I}$ found by the rolling horizon heuristic is feasible. It remains to prove that the solution does not require more vehicles than an optimal solution to the program VS. To this end, we introduce necessary notation to examine the vehicle flow per vehicle tour.

▶ **Definition 8.** *A* v-vehicle tour *is a sequence*

$$\tau = ((z_{o1} z_{d1} t_1), \dots, (z_{on} z_{dn} t_n))$$

of n consecutive vehicle trips with a positive flow of value $v \in \mathbb{R}_+$. Consecutive trips are characterized by

$$z_{di} = z_{oi+1} \text{ and } t_i + \delta_{z_{oi} z_{di}} = t_{i+1} \qquad \forall 1 \leq i < n.$$

A v-vehicle tour can be imagined as a tour that is driven by exactly v vehicles. Obviously, a vehicle schedule consists of many vehicle tours:

▶ **Observation 9** ([15]). *A feasible flow f can be decomposed into a finite set of vehicle tours $\{\tau_k\}_k$ such that the sum of all vehicle tour values $\sum_k v_k$ equals the total flow $|f|$. Each of the vehicle tours spans the whole time frame, i.e.,*

$$t_1 = 1 \text{ and } t_n + \delta_{z_{on} z_{dn}} > |T|.$$

Such a decomposition is not unique.

Next, we introduce vehicle duties to keep track of which vehicle tour serves which demand.

▶ **Definition 10.** *Let $\mathcal{I} = (Z, T, \delta, d)$ be an instance and let f be a feasible vehicle schedule, decomposed to a set of vehicle tours. A mapping γ from a vehicle tour τ and a vehicle trip (z_o, z_d, t) to a positive value,*

$$\gamma \colon (\tau, (z_o, z_d, t)) \mapsto \mathbb{R}_+$$

is called vehicle duty *if the following three conditions hold:*

1. *The value is only positive if the vehicle trip is in the tour,*

$$\gamma\left(\tau, (z_o, z_d, t)\right) > 0 \Rightarrow (z_o, z_d, t) \in \tau.$$

2. *The value is at most the value v of the vehicle tour,*

$$\gamma\left(\tau, (z_o, z_d, t)\right) \leq v.$$

3. *The sum of values for one vehicle trip (z_o, z_d, t) sum up to the demand $d_{z_o z_d t}$ on that trip,*

$$\sum_{\tau:\ (z_o,z_d,t)\in\tau} \gamma\left(\tau, (z_o, z_d, t)\right) = d_{z_o z_d t} \quad \forall (z_o, z_d, t).$$

A vehicle duty can be interpreted as assigning all demand to vehicle trips that meet the demand.

▶ **Observation 11.** *For each feasible flow f decomposed to a set of vehicle tours, there exists a vehicle duty such that the tour value v of each vehicle tour equals the value of the last positive demand assigned to the tour. Demand $d_{z_o z_d t}$ is called the last demand assigned to the tour if there is no other demand $d_{z_o{}' z_d{}' t'}$ assigned to that tour with $t' > t$.*

For any vehicle duty, this can easily be constructed by iteratively splitting each v-vehicle tour not fulfilling this criterion into two v_1 and v_2 vehicle tours with the same sequence of vehicle trips where at least one tour fulfills the criterion.

▶ **Definition 12.** *Such a vehicle duty is called* maximal vehicle duty*.*

Proof of Theorem 3. We show that a composed vehicle schedule of the rolling horizon heuristic is optimal for the whole time frame by induction over the number of horizons.

Induction basis It is easy to see that the optimization program $\overline{VS}$ finds an optimal solution f^1 for the first horizon $\{\underline{t}_1, \ldots, \overline{t}_1\} = \{1, \ldots, h\}$.

Induction hypothesis We consider a solution f^i of the rolling horizon algorithm for the first i horizons and as induction hypothesis we assume that the solution is optimal. That means, it is not possible to satisfy all demand $d_{z_o z_d t}$ for $t \leq \overline{t}_i$ with less than $x^i := |f^i|$ vehicles. This flow f^i is fixed up to the beginning of the overlap and may not be changed by the solution of a future horizon. The flow in the overlap, $f^i_{z_o z_d t}$ for $t \geq \overline{t}_i - o$ is overwritten by the solution of the next horizon and may change.

Induction step Let f^* be an optimal solution for $i+1$ horizons, for example found by solving the optimization model VS. Our aim is to show that a solution from the next iteration of the rolling horizon heuristic with an overlap of $o \geq 2 \cdot \max \delta_{z_o z_d} - 1$ is optimal for $i+1$ horizons. This is done by constructing a feasible flow f^{i+1} for the first $i+1$ horizons that is identical to the flow f^i before the overlap and uses $x^* := |f^*|$ vehicles. Since we can construct such a solution, Algorithm 2 in the rolling horizon heuristic will find a solution that is at least as good.

First, we consider a decomposition of the flow f^i into finitely many vehicle tours τ^i, and a maximal vehicle duty γ^i assigning all demand to the vehicle tours. We 'cut off' each vehicle tour τ^i

I after meeting the last demand that starts in the overlap and that is assigned to that tour, or else,

II if no demand starting in the overlap is assigned to that tour in the vehicle duty, after the first vehicle trip that ends *in* the overlap.

To formalize, let

$$\tau^i = ((z_{o1}, z_{d1}, t_1), \dots, (z_{ok}, z_{dk}, t_k), (z_{ok+1}, z_{dk+1}, t_{k+1}), \dots, (z_{on}, z_{dn}, t_n))$$

be a vehicle tour in flow f^i. Let (z_{ok}, z_{dk}, t_k) be the last vehicle trip starting in the overlap with demand assigned to the tour τ^i, or else, be the first vehicle trip that ends in the overlap. Then, the rear part after this vehicle trip, starting with $(z_{ok+1}, z_{dk+1}, t_{k+1})$, is cut off, which yields the incomplete tour

$$\tau' = ((z_{o1}, z_{d1}, t_1), \dots, (z_{ok}, z_{dk}, t_k)).$$

This can be interpreted as letting all vehicles from vehicle tour τ^i *wait* in zone z_{dk} at time interval $t_k + \delta_{z_{ok}z_{dk}}$.

- **I** We denote the number of all vehicles waiting in vertex (z, t) after meeting demand that starts in the overlap by w^{I}_{zt} and initialize the set of vertices where these vehicles are waiting as

$$W^{\mathrm{I}} = \{(z,t) : \; w^{\mathrm{I}}_{zt} > 0, \; z \in Z, \; \bar{t}_i - o < t \leq \bar{t}_i + \max \delta_{z_o z_d}\}.$$

- **II** Equivalently, we denote the number of all vehicles waiting in vertex (z, t) after the first vehicle trip ending in the overlap by w^{II}_{zt}.

This leaves us with incomplete vehicle tours that start in the first time interval and end sometime after the beginning of the overlap with waiting vehicles.

Next, we use these incomplete vehicle tours as a basis for the flow f^{i+1} that we want to construct as a solution for the first $i+1$ horizons. We set

$$f^{i+1}_{z_o z_d t} := \sum_{\tau' : \, (z_o, z_d, t) \in \tau'} v(\tau') \quad \forall z_o, z_d \in Z, \; t \in \{1, \dots, \bar{t}_i\}$$

where $v(\tau')$ is the flow value of vehicle tour τ'. We want to highlight three characteristics of f^{i+1}:

1. Since the tours τ' are not cut off before the start of the overlap, f^{i+1} is identical to flow f^i up to the beginning of the overlap. This is required for the construction of f^{i+1} since all vehicle trips before the overlap are fixed by design of the rolling horizon heuristic.
2. Since it is defined by incomplete tours, f^{i+1} is *not* a feasible flow (yet). The flow conservation does not hold at some nodes. In this proof, we show that it is possible to extend it to a feasible flow at these nodes.
3. Since the tours τ' are cut off after meeting the last demand, f^{i+1} does meet all demand starting up to the end of the overlap.

Our goal is to show that we can complete f^{i+1} to a feasible flow for $i+1$ horizons while using x^* vehicles. The sum of all flow values of the incomplete tours is x^i, equal to the flow value of f^i. It holds that $x^i \leq x^*$, otherwise f^i is not optimal for the first i horizons as $f^*|_i$ would be a better solution, which contradicts the induction hypothesis. In case that $x^i < x^*$ we add $(x^* - x^i)$ more vehicles to f^{i+1} at an arbitrary zone, for example by letting them stay in the first zone until the beginning of the overlap:

$$f^{i+1}_{11t} := f^i_{11t} + (x^* - x^i) \quad \forall t < \bar{t}_i$$

This increases the number of waiting vehicles $w^{\mathrm{II}}_{1\bar{t}_i}$ in node $(1, \bar{t}_i)$ by $(x^* - x^i)$. Then, the sum of all flow values in f^{i+1} is x^*, as in any optimal flow f^*.

Next, we consider an arbitrary but fixed optimal solution f^* for $i+1$ horizons, decomposed into finitely many vehicle tours τ^*, and a vehicle duty γ^* assigning all demand to the vehicle tours τ^*. Let $\mathcal{T}$ be the set of all tours τ^* in f^*. Our aim is to extend the incomplete tours in f^{i+1} with the rear parts of the tours in f^*.

- First, we consider waiting vehicles w^{I}_{zt} that met demand starting in the overlap. We extend f^{i+1} according to the following procedure:

  While W^{I} is not empty, we choose an arbitrary node $(z,t) \in W^{\mathrm{I}}$. By definition of W^{I}, at least w^{I}_{zt} demanded vehicle trips end in node (z,t). Hence, there exist vehicle tours $\tau^* \in \mathcal{T}$ that this demand was assigned to, otherwise f^* was infeasible. We extend f^{i+1} at node (z,t) with these tours τ^* from f^* until there are no more waiting vehicles w^{I}_{zt} in node (z,t):

    While $w^{\mathrm{I}}_{zt} > 0$, we take such a tour τ^* with tour value $v(\tau^*)$ and remove it from the set $\mathcal{T}$. If $v(\tau^*) > w^{\mathrm{I}}_{zt}$, we split the tour τ^* into two tours with the same sequence of vehicle trips as τ^*, one tour τ^*_w with flow value w^{I}_{zt}, and one tour τ^*_{v-w} with flow value $v(\tau^*) - w^{\mathrm{I}}_{zt}$. Else, for $v(\tau^*) \leq w^{\mathrm{I}}_{zt}$, we take the tour with the full value $v(\tau^*)$ and define $\tau^*_w := \tau^*$.

    We extend f^{i+1} at node (z,t) with tour τ^*_w and put τ^*_{v-w} back into the set $\mathcal{T}$. Extending f^{i+1} with tour τ^*_w means, we increase the flow value $f^{i+1}_{z_o' z_d' t'}$ for each vehicle trip (z_o', z_d', t') in tour τ^*_w after node (z,t) by the value $v(\tau^*_w)$,

    $$f^{i+1}_{z_o' z_d' t'} = f^{i+1}_{z_o' z_d' t'} + v(\tau^*_w) \quad \forall (z_o', z_d', t') \in \tau^*_w\colon \ t' \geq t.$$

    Based on this extension of f^{i+1}, we update the number of waiting vehicles:

      At the current node (z,t), there are $v(\tau^*_w)$ waiting vehicles less, hence, we set $w^{\mathrm{I}}_{zt} := w^{\mathrm{I}}_{zt} - v(\tau^*_w)$. Moreover, it might be that some further demand $d_{z_o' z_d' t'}$ starting in the overlap after node (z,t) was assigned to vehicle tour τ^*_w in the optimal flow f^*, i.e. $\gamma^*(\tau^*_w, (z_o', z_d', t')) > 0$ for $t < t' \leq \bar{t}_i$. Then, we assign this demand to the newly extended tour in f^{i+1} as well. In particular, we undo the assignment of this demand to another tour τ^{i+1} in f^{i+1}.

      If demand $d_{z_o' z_d' t'}$ was the last demand assigned to tour τ^{i+1}, this has two consequences: First, it caused waiting vehicles $w^{\mathrm{I}}_{z_d'(t' + \delta_{z_o' z_d'})}$ after the demanded vehicle trip (z_o', z_d', t'). We remove these waiting vehicles since the demand is met by the newly extended vehicle tour in f^{i+1} as well:

      $$w^{\mathrm{I}}_{z_d'(t'+\delta_{z_o' z_d'})} := \max\{w^{\mathrm{I}}_{z_d'(t'+\delta_{z_o' z_d'})} - \gamma^*(\tau^*_w, (z_o', z_d', t')), 0\}.$$

      Second, since the assignment of the last demand to tour τ^{i+1} is undone, either another demand $d_{z_o'' z_d'' t''}$ with $\bar{t}_i - o \leq t'' < t$ is the last demand, or no other demand that starts in the overlap is assigned to tour τ^{i+1}. In the first case, we increase the number of waiting vehicles $w^{\mathrm{I}}_{z_d''(t''+\delta_{z_o'' z_d''})}$ after that demand by $\gamma^*(\tau^*_w, (z_o', z_d', t'))$ since it is now the last demand. In the second case, we increase the number of waiting vehicles $w^{\mathrm{II}}_{\hat{z}\hat{t}}$ by $\gamma^*(\tau^*_w, (z_o', z_d', t'))$, where node $(\hat{z}, \hat{t})$ is the first node of tour τ^{i+1} in the overlap.

    If $w^{\mathrm{I}}_{zt} = 0$ for any node (z,t) after updating of the number of waiting vehicles, we remove it from W^{I} and continue with the next node in W^{I}.

  This procedure extends f^{i+1} with vehicle trips from f^* until there are no more waiting vehicles w^{I}_{zt} at node (z,t). During this construction, also the waiting vehicles at other nodes might be changed. We want to emphasize that this procedure is well-defined and finite. There exist sufficient vehicle tours in the set $\mathcal{T}$ to be chosen from in the procedure. For each node (z, t), at most vehicle tours with a total flow value of incoming demand at (z, t) are requested from set $\mathcal{T}$. Since all demand is met by the solution f^*, these tours exist. Furthermore, we reduce the waiting vehicles in all future nodes by the flow value of a tour, if a tour with assigned demand is removed from $\mathcal{T}$. Hence, taking a tour τ^* with assigned demand ending in (z, t) from set $\mathcal{T}$ is well-defined. In each update of the number of waiting vehicles, the total number of waiting vehicles never increases. Furthermore, it is impossible to process waiting vehicles caused by the same demand twice, which makes the procedure finite.

This procedure is applied to all nodes with waiting vehicles w^{I} until the set of waiting vehicles W^{I} is empty. After this procedure, we obtain an incomplete flow f^{i+1} with some complete tours that start in the first time interval and reach the end of the horizon, and some waiting vehicles w^{II} after the beginning of the overlap that were not treated yet.

II Second, we consider these waiting vehicles w^{II} after the beginning of the overlap.

We start with determining the amount of waiting vehicles w^{II}: Let x denote the sum of the flow values of the complete tours constructed in case I. Since these tours are based on flow f^i and extended with tours from the optimal solution f^*, the sum of the flow values of the vehicle tours left in the set of tours $\mathcal{T}$ is equal to the total number of waiting vehicles w^{II}, namely $(x^* - x)$. The waiting vehicles w^{II}_{zt} are present at nodes (z, t) after the first vehicle trip that starts before and ends in the overlap. Hence, the vehicles are waiting in zone z at the beginning of time interval t with

$$t \leq \bar{t}_i - o - 1 + \max \delta_{z_o z_d}.$$

It is important to note that all demand in the overlap is met by the complete tours constructed in case I and we do not need to take care of this.

We disconnect the remaining vehicle tours in the set $\mathcal{T}$ at the first node (z', t') after the overlap into two incomplete tours. Then, it is possible relocate the waiting vehicles to zone z' within at most $\max \delta_{z_o z_d}$ time intervals, that means the vehicles can be available in zone z' at latest at

$$t \leq \bar{t}_i - o - 1 + 2 \cdot \max \delta_{z_o z_d} \leq \bar{t}_i \leq t'.$$

That means, it is possible to relocate the waiting vehicles w^{II} within the overlap and extend f^{i+1} with the rear parts of the disconnected vehicle tours from f^*.

As a result, we obtain a feasible flow f^{i+1} for the first $i + 1$ horizons that uses x^* vehicles. This flow is identical to flow f^i before the overlap, and identical to flow f^* after the overlap. For the time intervals in the overlap, we constructed f^{i+1} in such way that it connects f^i and f^*. By construction, it is ensured that all demand is satisfied and with the help of waiting vehicles we could connect the flows ensuring flow conservation at each node.

Since it is possible to construct a flow f^{i+1} with these characteristics, the rolling horizon algorithm will find a vehicle schedule for $i + 1$ horizons that is at least as good. The theorem follows by induction. ◀