

Complexity of scheduling for DARP with soft ride times

Janka Chlebíková¹, Clément Dallard¹, and Niklas Paulsen²

¹ School of Computing, University of Portsmouth, United Kingdom

{janka.chlebkova,clement.dallard}@port.ac.uk

² Institut für Informatik, Christian-Albrechts Universität zu Kiel, Germany

npau@informatik.uni-kiel.de

Abstract. The Dial-a-Ride problem may contain various constraints for pickup-delivery requests, such as time windows and ride time constraints. For a tour, given as a sequence of pickup and delivery stops, there exist polynomial time algorithms to find a schedule respecting these constraints, provided that there exists one. However, if no feasible schedule exists, the natural question is to find a schedule minimising constraint violations. We model a generic fixed-sequence scheduling problem, allowing lateness and ride time violations with linear penalty functions and prove its APX-hardness. We also present an approach leading to a polynomial time algorithm if only the time window constraints can be violated (by late visits). Then, we show that the problem can be solved in polynomial time if all the ride time constraints are bounded by a constant. Lastly, we give a polynomial time algorithm for the instances where all the pickups precede all the deliveries in the sequence of stops.

Keywords: dial-a-ride · scheduling · Vehicle Routing Problem · NP-hardness · ride times · time windows

1 Introduction

The Dial-A-Ride Problem (DARP) is a well studied variant of the Vehicle Routing Problem. The DARP, with its various restrictions, serves as a model for many real-world problems from logistics, *e.g.* passenger transportation, or pickup-delivery of perishable goods. For a review of DARPs, we refer the readers to [2].

The study of the Dial-A-Ride Problem can be split into three main subproblems: the *clustering* of the requests into tours, the *routing* of the stops within each tour into a sequence, and the *scheduling* of the stops inside the tours [4]. These problems are the source of major research topics in operation research, each of them intensively studied. To get a better understanding of the inherent complexity of the problems, and eventually obtain faster algorithms, many restricted models have been studied.

In this paper, we focus on the scheduling subproblem, where the input of the problem is a tour with fixed sequence of stops, a set of pickup-delivery requests, and time constraints with their corresponding penalty functions. Each pickup-delivery request is represented by two stops, the first one as a pickup, and the

second one as a delivery. The visit of each stop has to be performed within a given *time window*. Furthermore, the time between the scheduled pickup and delivery of a same request is bounded by a given *ride time*.

Time windows and ride times constraints are naturally arising when scheduling pickups and deliveries. When both constraints must be respected in the solution, there exist efficient algorithms [6,9,11]. However, in all these approaches, ride time and time window constraints are *hard* in the sense that a solution must respect all the constraints.

In case there is no feasible schedule, one may look for a schedule “close” to a feasible one with minimal penalties. Therefore, variants of the problem with *soft* constraints, in which the violation of constraints is allowed but penalised, have been introduced. Depending on the type of constraints and their corresponding penalty functions, various results can be obtained. For instance, when the only constraints are time windows for the stops, Dumas et al. [5] proposed a linear programming approach for convex penalty functions with a linear time complexity, but their algorithm does not incorporate ride time constraints. To the best of our knowledge, the complexity of the problem with soft ride times constraints was previously unknown.

In this paper, we propose a systematic study of the complexity of the problem when allowing lateness at stops (scheduling after the time windows) and ride time violation.

2 Problem Statement

In the following, we assume that $0 \in \mathbb{N}$. For $\Delta \in \{\leq, <, \geq, >\}$, and X a well ordered set, let $X_{\Delta x} := \{y \in X : y \Delta x\}$ and $X[i]$ be the i -th smallest element of X .

We are given a sequence $\mathbf{S} = (1, 2, \dots, 2n)$, $n \in \mathbb{N}$, of $2n$ stops in the order in which their visits must be scheduled (in case of no ambiguity $s \in \mathbf{S}$ also represents an integer). Each stop $s \in \mathbf{S}$ is associated with a time interval (a_s, b_s) , $0 \leq a_s \leq b_s$, representing a time window in which a visit of the stop s should take place (without loss of generality we suppose that $a_1 = 0$). Furthermore, we have a set \mathcal{P} of n requests representing the pairs (p, d) of stops from \mathbf{S} , $p < d$, where p is a pickup and d is a delivery stop. Each request (p, d) has a time constraint $r_{p,d}$ ($r_{p,d} \geq 0$) on the ride time: a visit at stop d should be scheduled at most $r_{p,d}$ time units after the visit at stop p . Each stop s serves exactly one request (either as a pickup or as a delivery stop) and all times are represented as non-negative integers.

As it has been mentioned in Section 1, it is not always possible to schedule the visits for all stops (in a given order) with respect to their time windows and ride time constraints. Therefore we introduce the model in which the time window and ride time constraints can be violated for penalties (soft constraints).

In order to model soft constraints: (i) each stop $s \in \mathbf{S}$ is associated with a penalty function $\sigma_s^L : \mathbb{N} \rightarrow \mathbb{Q}$, mapping visit times which are later than the time window bounds to a non-negative penalty, and (ii) each request $(p, d) \in \mathcal{P}$ is

associated with a penalty function $\sigma_{p,d}^{RT} : \mathbb{N} \rightarrow \mathbb{Q}$, mapping ride times exceeding ride time constraints to a non-negative penalty.

In this paper we suppose that all penalty functions are linear non-decreasing functions. We consider a restricted model where earliness at stops is not allowed, hence a stop must be either scheduled within or after its time window. Therefore, for each stop $s \in \mathcal{S}$ and the visit time x at stop s , then $x \geq a_s$. Moreover, we have the function $\sigma_s^L(x)$ such that $\sigma_s^L(x) = 0$ for $x \leq b_s$ and otherwise $\sigma_s^L(x) = \alpha_s \cdot (x - b_s) + \beta_s$ for given $\alpha_s, \beta_s \in \mathbb{Q}_{\geq 0}$. Analogously, for each request $(p, d) \in \mathcal{P}$ we have the function $\sigma_{p,d}^{RT}$ with $\sigma_{p,d}^{RT}(x) = 0$ for $x \leq r_{p,d}$ and otherwise $\sigma_{p,d}^{RT}(x) = \alpha_{p,d} \cdot (x - r_{p,d}) + \beta_{p,d}$ for given $\alpha_{p,d}, \beta_{p,d} \in \mathbb{Q}_{\geq 0}$.

A *schedule* $\mathbf{t} = (t_1, \dots, t_{2n})$ is a sequence of visit times for all the stops in \mathcal{S} (in the given order), where we say that \mathbf{t} schedules a stop $s \in \mathcal{S}$ at time t_s . We say that a schedule \mathbf{t} is *feasible* if and only if for all stops $s \in \mathcal{S}$, $t_s \geq a_s$ and for any $s \in \mathcal{S}_{<2n}$, $t_s \leq t_{s+1}$. Obviously, each time window and ride time constraint is either *violated* or *satisfied* by a schedule \mathbf{t} . The *cost* $c(\mathbf{t})$ of a schedule \mathbf{t} is the sum of the penalties of violated constraints:

$$c(\mathbf{t}) = \sum_{s \in \mathcal{S}} \sigma_s^L(t_s) + \sum_{(p,d) \in \mathcal{P}} \sigma_{p,d}^{RT}(t_d - t_p). \quad (1)$$

When we solve the MIN PICKUP-DELIVERY SCHEDULING problem, we look for a feasible schedule \mathbf{t} with a minimum cost.

MIN PICKUP-DELIVERY SCHEDULING(MIN PDS)

Input An instance I of MIN PDS.

Task Find a feasible schedule \mathbf{t} of I such that $c(\mathbf{t})$ is minimum.

We also consider two special cases of the main problem:

MIN PICKUP-DELIVERY SCHEDULING WITH HARD RIDE TIME CONSTRAINTS(MIN PDS-HRT)

Input An instance I of MIN PDS.

Task Find a feasible schedule \mathbf{t} of I respecting all ride time constraints and minimising $c(\mathbf{t})$.

MIN PICKUP-DELIVERY SCHEDULING WITH HARD TIME WINDOWS(MIN PDS-HTW)

Input An instance I of MIN PDS.

Task Find a feasible schedule \mathbf{t} of I respecting all time windows constraints and minimising $c(\mathbf{t})$.

Throughout the paper, we label by (*) the statements for which the omitted proofs can be found in Appendix.

Table 1. Overview of complexity results classified by constraints. Arrows mean results are inferred.

<i>Ride time constraints</i>	<i>Time window constraints</i>	
	Hard	Soft
Hard	$\mathcal{O}(n)$ [6]	$\mathcal{O}(n)$ [Section 3.2]
$0, \beta_{p,d} = 0$	$\mathcal{O}(n)$	$\leftarrow \mathcal{O}(n)$ [Section 4, from [5]]
Soft, bounded values	P	\leftarrow P [Section 4]
Soft, unbounded values	NP-hard, APX-hard [Section 3.1]	\rightarrow NP-hard, APX-hard

2.1 Our Contribution

The paper investigates how penalisation of the time window and ride time constraints contributes to the computational complexity of the problem. We show that an essential factor for the complexity are soft maximum ride time constraints. In Subsection 2.2, we give some remarks on our model. An overview of complexity results is shown in Table 1. We prove the NP-hardness of the main problem MIN PICKUP-DELIVERY SCHEDULING and its special case with hard time window constraints, MIN PDS-HTW (Section 3.1). Nevertheless, we show that the problem can be solved in polynomial time in case of hard ride time constraints, MIN PDS-HRT (Section 3.2). Further underlining the role of ride time constraints, we give a parameterised algorithm that solves MIN PICKUP-DELIVERY SCHEDULING in polynomial time if all ride time constraints are bounded by a constant (Section 4). In Section 5 we show that some structural properties in the sequence of the stops can be exploited to find a polynomial time algorithm. Namely, we present an $\mathcal{O}(n^4)$ time algorithm when all pickups precede all the deliveries in the sequence.

2.2 Remarks on the Model

Driving times, (un)loading times (*). In favour of simplicity, our model neglects times needed to travel between stops as well as loading or unloading times. We emphasise that this is not restrictive, since we focus on the scheduling of fixed sequences. An instance with given driving and (un)loading times can be transformed to an equivalent instance of our form with a simple preprocessing.

Waiting times. Constraints w_s on the time to wait between two consecutive stops s and $s + 1$ (as in [6]) are omitted in our model since they can be expressed by ride time constraints: assume for $s \in \mathcal{S}_{<2n}$ the constraint $\mathbf{t}_s + w_s \geq \mathbf{t}_{s+1}$ is given for schedules \mathbf{t} , $w_s \geq 0$. Simply insert two additional stops: the stop p immediately before s and the stop d immediately after $s + 1$ into \mathcal{S} and add a request (p, d) to \mathcal{P} with $r_{p,d} = w_s$. Replacing all waiting time constraints leads to an equivalent instance with at most $6n - 2 \in \mathcal{O}(n)$ requests.

Increasing time windows opening times. Since earliness is not allowed in our model, we expect that any instance of $2n$ stops has $a_s \leq a_{s+1}$ for all

$s \in S_{<2n}$. If for a stop s , $s \in S_{<2n}$, we have $a_s > a_{s+1}$, for any feasible schedule \mathbf{t} it holds $\mathbf{t}_s \geq a_s$ and $\mathbf{t}_{s+1} \geq \mathbf{t}_s$ and therefore $\mathbf{t}_{s+1} < a_s$ cannot hold for any feasible scheduling. We can therefore preprocess the instance in such a way that for all $s \in S_{<2n}$, $a_{s+1} \leftarrow \max\{a_s, a_{s+1}\}$. Notice that due to this property and the fact that the last stop $2n$ is a delivery stop, it always exists an optimal schedule \mathbf{t}^* such that $\mathbf{t}_{2n}^* = a_{2n}$.

As soon as possible deliveries. All deliveries can be scheduled at a time as soon as possible without increasing costs. Let d be a delivery of a request $(p, d) \in \mathcal{P}$ and \mathbf{t} a feasible schedule. We define a schedule \mathbf{t}' with $\mathbf{t}'_s := \mathbf{t}_s$ for all stops $s \in S \setminus \{d\}$ and $\mathbf{t}'_d := \max\{a_d, t_{d-1}\}$. Clearly, \mathbf{t}' is feasible. Obviously, \mathbf{t}'_d can only decrease the lateness at d as well as the ride time for $(p, d) \in \mathcal{P}$ with no changes in scheduling of the other stops.

3 Complexity study

3.1 Min Pickup-Delivery Scheduling with Hard Time Windows

In this subsection we study the variant of MIN PICKUP-DELIVERY SCHEDULING in which ride time constraints may be violated in return for a penalty (soft constraints), but the time windows must be respected (hard constraints). We show that such a problem, called MIN PICKUP-DELIVERY SCHEDULING WITH HARD TIME WINDOWS (MIN PDS-HTW), is NP-hard and APX-hard even in case of restricted time windows and very circumscribed penalty functions. The proof is based on a reduction from the MAXIMUM DICUT problem which is known to be NP-hard and APX-hard when restricted to directed acyclic graphs (DAG) [7, 10].

A directed cut (A, B) of a directed graph $G = (V, E)$ is a partition of V into two subsets A, B . Its size $s(A, B) := |\{(u, v) \in E : u \in A, v \in B\}|$ is the number of outgoing arcs from A to B . The MAXIMUM DICUT problem is defined as follows:

MAXIMUM DICUT

Input A directed graph $G = (V, E)$.

Task Find a directed cut (A, B) of maximum size in G .

Theorem 1. MIN PICKUP-DELIVERY SCHEDULING WITH HARD TIME WINDOWS is NP-hard.

Proof. Firstly, the decision version of MIN PDS-HTW is clearly in NP. Let $G = (V, E)$ be a connected DAG such that $|V| = n$, $|E| = m$. Since G is a DAG, the vertices of G can be labelled by $1, 2, \dots, n$ in a topological ordering in such a way that for any arc $(u, v) \in E$ it holds $\text{lab}(u) < \text{lab}(v)$, where $\text{lab}(z)$ represents the number used for labelling the vertex z [12].

In the following we show how the graph G can be transformed into an instance I of MIN PDS-HTW. The sequence of stops for I is defined as the concatenation $S := S^1 S^2 \dots S^n$, where each S^v represents a gadget of stops for each vertex $v \in V$.

Let $v \in V$ be fixed, then the gadget S^v contains the stop s_e^v for each arc e of G incident to v , *i.e.* $e = (v, u)$ or $e = (u, v)$ for $u \in V$. An example of such a gadget is depicted in Figure 1.

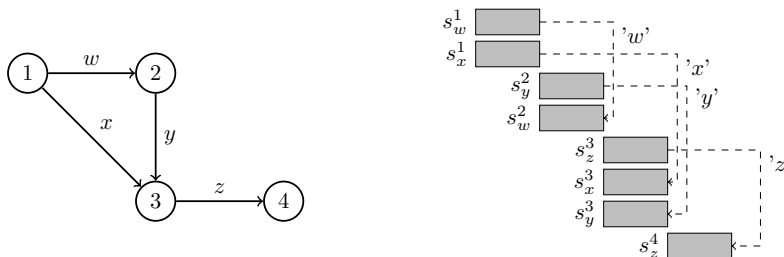


Fig. 1. A DAG (left) transformed into an instance of PDS (right). Gray boxes are time windows of length 1. Note there are two stops for every arc of G and the stops are grouped in gadgets for each vertex of G .

The stops within the gadget S^v are ordered in such a way that all stops belonging to outgoing arcs precede all stops belonging to ingoing arcs. Note that S has $2m$ stops. For each vertex $v \in V$ the time windows of all stops $s \in S^v$ are set to $a_s := lab(v) - 1$ and $b_s := lab(v)$. The requests correspond to the arcs in G , hence $\mathcal{P} = \{(s_e^u, s_e^v) : e = (u, v) \in E\}$ and for each $(p, d) \in \mathcal{P}$ the ride time is set to be $r_{p,d} = a_d - b_p$. Due to the specific numbering of vertices and sizes of windows, the stop p always precedes the stop d and $r_{p,d} \geq 0$ for all $(p, d) \in \mathcal{P}$. Setting the penalty coefficients $\alpha_{p,d} = 0$ and $\beta_{p,d} = 1$, the cost of a schedule corresponds to the number of violated ride time constraints. Obviously, the transformation from G to the instance I can be done in polynomial time.

Now we show that G has a directed cut of size at least $(m - k)$ if and only if there exists a schedule t violating at most k ride time constraints, for any $k \in \mathbb{N}$.

\Rightarrow Suppose there exists a directed cut (A, B) in G of size at least $(m - k)$. Define a schedule t such that for every vertex $v \in A$ and every stop $s \in S^v$ we set $t_s := b_s$ and for all other stops $t_s := a_s$. Clearly, t is a feasible schedule. Each arc $(u, v) \in E$ corresponds to the unique $(p, d) \in \mathcal{P}$ with $p \in S^u$ and $d \in S^v$. If $u \in A$ and $v \in B$, then $t_d - t_p = b_d - a_p \leq r_{p,d}$, hence the ride time constraint is respected. As we suppose $s(A, B) \geq m - k$, the previous holds for at least $(m - k)$ requests. With $|\mathcal{P}| = m$, it implies t violates at most k ride time constraints.

\Leftarrow Now suppose there exists a schedule t for I violating at most k ride time constraints. For each vertex $v \in V$ let $s[v]$ be the first delivery stop in S^v and if there is no such stop, then $s[v]$ be the last pickup stop in S^v . This allows us to define a partition of V in the following way: for each vertex $v \in V$, if $t_{s[v]} = b_{s[v]}$ then $v \in A$, otherwise $v \in B$. Fix a $(p, d) \in \mathcal{P}$ and let $u, v \in V$ be such vertices that p (resp. d) is from the gadget S^u (resp. S^v). If t satisfies the ride time constraint of (p, d) , then by the definition of I it must hold $t_p = b_p$ and

$t_d = a_d$. Since p is from the gadget S^u and \mathbf{t} is feasible, $t_{s[v]} \geq b_p$ and necessarily $b_{s[u]} = b_p$, hence $t_{s[u]} = b_{s[u]}$. Analogously, $t_{s[v]} = a_{s[v]}$. Therefore, $u \in A$ and $v \in B$. As we suppose that in the schedule \mathbf{t} at most k ride time constraints are violated, then at least $|\mathcal{P}| - k = m - k$ are satisfied. Since each satisfied ride time leads to a distinct arc going from A to B , $s(A, B) \geq m - k$.

The reduction defined in Theorem 1 is in fact a Strict-reduction [3] between the optimisation problems. As Lampis et al. [10] proved the MAXIMUM DICUT problem is APX-hard even when restricted on DAGs, the following result follows:

Corollary 1. MIN PICKUP-DELIVERY SCHEDULING WITH HARD TIME WINDOWS *is APX-hard*.

Now we argue that the main problem MIN PICKUP-DELIVERY SCHEDULING is NP-hard and APX-hard too. The idea is to set the penalties for lateness at each stop to such values that any optimal schedule must respect the time windows. Let I be an instance of MIN PDS-HTW with n requests. As mentioned in Subsection 2.2, there exists an optimal schedule \mathbf{t} of I such that $t_{2n} = a_{2n}$. Therefore the actual ride time of each request is bounded by the value a_{2n} . Let $k = \max_{(p,d) \in \mathcal{P}} \sigma_{p,d}^{RT}(a_{2n})$. Then the instance I can be transformed into an instance I' of MIN PDS by setting $\alpha_s = 0$ and $\beta_s = nk + 1$ for all $s \in \mathcal{S}$ (hence $\sigma_s^L(x) = nk + 1$ for $x > b_s$). The cost of any schedule respecting time windows is at most nk , hence there exists an optimal schedule of I' with cost strictly less than $nk + 1$. Such a schedule must respect time windows and therefore is also valid for the instance I of MIN PDS-HTW, hence MIN PDS-HTW can be seen as a special case of MIN PDS. Therefore we can conclude

Corollary 2. MIN PICKUP-DELIVERY SCHEDULING *is NP-hard and APX-hard*.

3.2 Min Pickup-Delivery Scheduling with Hard Ride Time Constraints

In this subsection we study the variant of MIN PICKUP-DELIVERY SCHEDULING in which the ride time constraints must be respected (hard constraints), while time windows may be violated in return for penalty (soft constraints). As it was mentioned in Section 1, we consider a model in which *lateness* is the only possible way to violate a time window restriction. We prove that this variant of the problem, called MIN PICKUP-DELIVERY SCHEDULING WITH HARD RIDE TIME CONSTRAINTS (MIN PDS-HRT), can be solved in linear time, compared to NP-hardness of MIN PICKUP-DELIVERY SCHEDULING WITH HARD TIME WINDOWS shown in Section 3.1.

When both ride times and time window constraints are hard, a linear time algorithm was proposed by Firat and Woeginger in [6]. It has also been adapted to handle additional *minimum* ride time constraints in [8]. We show how the same approach can be used to minimise lateness penalties in MIN PICKUP-DELIVERY SCHEDULING WITH HARD RIDE TIME CONSTRAINTS.

Our idea, similarly to the one used in [6, 8], is to formulate a difference constraint system (DCS) with variables of the schedule and interpret it as a graph in which the existence of negative weight cycles is equivalent to infeasibility of the DCS. In these papers it is shown how to apply the single-source shortest path algorithm for interval graphs presented in [1] to test the existence of negative weight cycles in linear time. In case of feasible instances, a solution can be extracted in linear time as well. We point out that this approach will lead to a schedule visiting every stop as late as possible: the scheduled time of each stop is chosen by the length of a shortest path from the start vertex. This path corresponds to a chain of difference equations and can be seen as the tightest upper bound on the timing value. Since the shortest path lengths are upper bounds this implies that no feasible schedule can visit any of the stops later.

Theorem 2 (*). MIN PICKUP-DELIVERY SCHEDULING WITH HARD RIDE TIME CONSTRAINTS *can be solved in linear time.*

4 Bounded Ride Time Constraints

The MIN PICKUP-DELIVERY SCHEDULING problem is NP-hard as it follows from Section 3.1. In this section we show that some restrictions on the parameters of the problem improve the complexity of the problem.

Given an instance of MIN PDS, let \mathcal{W} be the set of all time window bounds for all stops, *i.e.* $\mathcal{W} = \bigcup_{s \in \mathcal{S}} \{a_s, b_s\}$, and $\mathcal{J}_s := \{(p, d) \in \mathcal{P} : d > s \text{ and } p \leq s\}$ be the set of the *loaded requests* after the stop $s \in \mathcal{S}$, and its size $\text{load}(s) := |\mathcal{J}_s|$.

We suppose that $\mu \in \mathbb{N}$ is a fixed constant. Let μ -MIN PICKUP-DELIVERY SCHEDULING (μ -MIN PDS) be the restriction of the MIN PICKUP-DELIVERY SCHEDULING problem to the instances with the ride time constraints bounded by μ , *i.e.* $r_{p,d} \leq \mu$ for all $(p, d) \in \mathcal{P}$. In the following we propose a polynomial-time algorithm for μ -MIN PDS.

Firstly we observe that the visit times of an optimal solution can be chosen from a restricted set of time values. We define the set

$$\widetilde{\mathcal{W}} := \left(\bigcup_{w \in \mathcal{W}} [w - n\mu, w + n\mu] \right)_{\geq a_1, \leq a_{2n}}.$$

Note that $\widetilde{\mathcal{W}} = \mathcal{W}$ in case of 0-MIN PDS.

We say that a schedule \mathbf{t} is *defined in* $\widetilde{\mathcal{W}}$ if and only if $t_s \in \widetilde{\mathcal{W}}$ for all $s \in \mathcal{S}$. The following lemma states that in fact there is an optimal schedule defined in $\widetilde{\mathcal{W}}$ (see the details in Subsection 7.3).

Lemma 1 (*). *For a given instance of μ -MIN PDS there is an optimal schedule t defined in $\widetilde{\mathcal{W}}$.*

Definition 1. *For a given schedule t of an instance of μ -MIN PDS and a stop $\ell \in \mathcal{S}$ we define the partial cost $\tilde{c}(t, \ell)$ of t up to the stop $\ell \in \mathcal{S}$ as*

$$\tilde{c}(t, \ell) := \sum_{s \in \mathcal{S}_{\leq \ell}} \sigma_s^L(t_s) + \sum_{(p,d) \in \mathcal{J}_\ell} \sigma_{p,d}^{RT}(t_\ell - t_p) + \sum_{\substack{(p,d) \in \mathcal{P} \\ d \leq \ell}} \sigma_{p,d}^{RT}(t_d - t_p).$$

In the following lemma we prove some observations regarding the partial cost function.

Lemma 2 (*). *For a given schedule \mathbf{t} of the instance μ -MIN PDS and the stop $\ell \in \mathcal{S}$ the following hold*

- (i) $\tilde{c}(\mathbf{t}, 1) = \sigma_1^L(\mathbf{t}_1)$;
- (ii) $\tilde{c}(\mathbf{t}, \ell + 1) = \tilde{c}(\mathbf{t}, \ell) + \sigma_{\ell+1}^L(\mathbf{t}_{\ell+1}) + \sum_{(p,d) \in \mathcal{J}_\ell} f_{p,d}^\ell(\mathbf{t})$
with $f_{p,d}^\ell(\mathbf{t}) := \begin{cases} \alpha_{p,d}(\mathbf{t}_{\ell+1} - \mathbf{t}_\ell), & \text{if } \mathbf{t}_\ell - \mathbf{t}_p > r_{p,d} \\ \sigma_{p,d}^{RT}(\mathbf{t}_{\ell+1} - \mathbf{t}_p), & \text{if } \mathbf{t}_\ell - \mathbf{t}_p \leq r_{p,d} \end{cases}$,
- (iii) $\tilde{c}(\mathbf{t}, 2n) = c(\mathbf{t})$,

Let I be an instance of μ -MIN PDS. For each stop l , $l = 1, 2, \dots, 2n$ we define so called l -labels to capture the structure of ‘similar’ schedules for I . The labels enable to restrict the number of schedules for I in each step and therefore to use the ideas of dynamic programming.

As it follows from Lemma 1, we can focus on schedules defined in $\widetilde{\mathcal{W}}$ only. For a schedule \mathbf{t} defined in $\widetilde{\mathcal{W}}$ and $\ell \in \mathcal{S}$, the ℓ -label of \mathbf{t} is defined as

$$\text{Label}_\ell(\mathbf{t}) = (\mathbf{t}_\ell, s^0, s^1, \dots, s^\mu, \tilde{c}(\mathbf{t}, \ell)),$$

where $s^m := \min\{s \in \mathcal{S} \text{ such that } \mathbf{t}_s \geq \mathbf{t}_\ell - m\}$, i.e. s^m is the first stop of the schedule \mathbf{t} visited at or after time $(\mathbf{t}_\ell - m)$ for any m , $0 \leq m \leq \mu$.

Note that every schedule has one such label for each $\ell \in \mathcal{S}$, but a label may describe more (different) schedules. We say that a label $L \in \widetilde{\mathcal{W}} \times \mathcal{S}^{\mu+1} \times \mathbb{Q}^+$ is a feasible ℓ -label if there exists a feasible schedule \mathbf{t} for I with $\text{Label}_\ell(\mathbf{t}) = L$.

In the following lemma we prove that in each stop there is a restriction on the number of labels to consider to find an optimal schedule.

Lemma 3 (Domination rule(*)). *Let I be an instance of μ -MIN PDS and the stop $\ell \in \mathcal{S}$ be fixed. Let $L^1 = (\tau, s^0, s^1, \dots, s^\mu, \tilde{c}_1)$ and $L^2 = (\tau, s^0, s^1, \dots, s^\mu, \tilde{c}_2)$ be feasible ℓ -labels with $\tilde{c}_1 \leq \tilde{c}_2$. Then there is a feasible schedule \mathbf{t} with $\text{Label}_\ell(\mathbf{t}) = L^1$ such that $c(\mathbf{t}) \leq c(\mathbf{t}')$ for any feasible schedule \mathbf{t}' with $\text{Label}_\ell(\mathbf{t}') = L^2$. We say that the label L_1 dominates the label L_2 .*

Now, according to Lemma 3, we can give an upper bound on the number of non-dominated labels for any fixed stop $l \in \mathcal{S}$. There are at most $|\widetilde{\mathcal{W}}|$ possibilities for the first item of the label, hence $\mathcal{O}(\mu \cdot n^2)$ if $\mu > 0$ (in case $\mu = 0$ only $\mathcal{O}(n)$), and $\mathcal{O}(n)$ choices for each of the next $(\mu + 1)$ items of the label.

Remark 1. For each instance of μ -MIN PDS and a stop $\ell \in \mathcal{S}$ the number of non-dominated ℓ -labels is bounded by $\mathcal{O}(\mu \cdot n^{\mu+3})$ if $\mu > 0$ and by $\mathcal{O}(n^2)$ if $\mu = 0$.

This leads to the following results:

Lemma 4. *An instance of μ -MIN PDS with $\mu > 0$ can be solved in time $\mathcal{O}(\mu^2 \cdot n^\mu \cdot \text{poly}(n))$.*

Proof. Starting with the initial labels $(\tau, 1, \dots, 1, \sigma_1^I(\tau))$ for each $\tau \in \widetilde{\mathcal{W}}_{\geq a_1}$ we have all the labels for the first stop for any feasible schedule defined on $\widetilde{\mathcal{W}}$, by Lemma 2 (i). The labels for the stop $\ell \in \mathcal{S}_{>1}$ can be calculated from the labels of the stop $(\ell - 1)$. For a non-dominated label $(\tau, s^0, \dots, s^\mu, \bar{c})$ of the stop $(\ell - 1)$ (there are $\mathcal{O}(\mu \cdot n^{\mu+3})$ such labels) do the following: for every possible visit time $\tau' \in \widetilde{\mathcal{W}}_{\geq \tau}$ at the stop ℓ (there are $\mathcal{O}(\mu \cdot n^2)$ such possible time visits), generate a new label:

- the first item of the label is τ' ;
- the s^* items are defined in the following way: $(\tau' - \tau)$ items have the value ℓ (truncate to at most $\mu + 1$ items), if $(\tau' - \tau) < \mu + 1$, then start to add the items s^0, \dots, s^μ from the previous $(\ell - 1)$ -label until there are $(\mu + 1)$ s^* -items,
- the new cost can be calculated in a linear time from the given label using Lemma 2 (ii).

Overall each new label is generated in time $\mathcal{O}(n)$. Any label at the stop $2n$ minimising the last item of the label (cost) represents only optimal schedules by Lemma 2 (iii).

Lemma 5. *An instance of 0-MIN PDS can be solved in time $\mathcal{O}(n^5)$.*

Proof. The result follows from the proof of Lemma 4 considering $\widetilde{\mathcal{W}} = \mathcal{W}$ and the fact that the number of non-dominated labels per stop is bounded by $\mathcal{O}(n^2)$.

We consider another specific case, when the goal is to minimise the sum of the lateness penalties and the sum of the ride times. This implies that $\mu = 0$. Since driving times are excluded from instances of our model (see Subsection 2.2), all ride times can be seen as excess ride times (excess ride times are defined as the actual ride time minus the driving time). The problem can be solved with the algorithm of Dumas et al. [5] in linear time (*). The ride times can also be minimised in a weighted manner, using $\alpha_{p,d} \geq 0$ for $(p, d) \in \mathcal{P}$. The waiting time before a stop $s \in \mathcal{S}_{>1}$ is then simply weighted by $\sum_{(p,d) \in \mathcal{J}_{s-1}} \alpha_{p,d}$.

5 Special Patterns in the Sequence of Stops

In this section we study a class of polynomial time solvable instance of MIN PICKUP-DELIVERY SCHEDULING. We introduce the *First Pickup Then Deliveries (FPTD)* instances in which all the stops $1, \dots, n$ are pickup stops, and the stops $n + 1, \dots, 2n$ are delivery stops. We show that MIN PICKUP-DELIVERY SCHEDULING can be solved in polynomial time in the class of *FPTD* instances despite the NP-hardness of the problem (Section 3.1).

Firstly, we prove that for each stop we can reduce the set of potential scheduling times to a subset polynomial in size. Each time in this subset is calculated from the time windows and maximal ride time values of the instance.

Lemma 6 (*). *Let I be an FPTD instance with $2n$ stops. Then there exists an optimal schedule \mathfrak{t} of I such that for each $s \in \mathcal{S}$:*

- if $t_s < t_n$, then $t_s \in \mathcal{B}^s(t_n) := \left(\{b_{s'} : s' \in \mathcal{S}, s \leq s' \leq n\}_{<t_n} \cup \bigcup_{(p,d) \in \mathcal{P}, p \leq s} \{\max\{t_n, a_d\} - r_{p,d}\} \right)_{\geq a_s}$;
- if $t_s = t_n$, then

$$t_s \in \mathcal{C} := \{a_n\} \cup \bigcup_{(p,d) \in \mathcal{P}} \{b_p, a_d - r_{p,d}\} \cup \bigcup_{(p',d') \in \mathcal{P}} \{b_p + r_{p',d'}, a_d - r_{p,d} + r_{p',d'}\};$$
- if $t_s > t_n$, then $t_s = a_s$.

According to Lemma 6, there is an optimal schedule \mathbf{t} with $t_n \in \mathcal{C}$, and $|\mathcal{C}|$ is quadratic in the instance size. Moreover, when t_n is fixed, each stop $s \in \mathcal{S}_{<n}$ with $t_s < t_n$ belongs to $\mathcal{B}^s(t_n)$, linear in size. Obviously, when t_n is fixed, one can schedule all deliveries $d \in \mathcal{S}_{>n}$ at $t_d := \max\{t_n, a_d\}$. In the following, we show how one can efficiently calculate an optimal schedule for a fixed t_n .

Let $p \in \mathcal{S}_{\leq n}$ be a pickup stop and $(p, d) \in \mathcal{P}$ its corresponding request. We define the function σ_p^k as the partial cost of scheduling p when $t_n = k$ such that if p is scheduled at l then $\sigma_p^k(l) = \sigma_p^L(l) + \sigma_{p,d}^{RT}(\max\{k, a_d\} - l)$. In order to define a recursive equation for calculating the cost of the optimal schedule, we define for each pickup stop $s \in \mathcal{S}_{\leq n}$ the function $T_s^k : \mathbb{N} \rightarrow \mathbb{N}$:

$$T_s^k(j) := \begin{cases} \max(\mathcal{B}^s(k) \cup \{k\})_{\leq j} & \text{if } (\mathcal{B}^s(k) \cup \{k\})_{\leq j} \neq \emptyset, \\ -1 & \text{otherwise.} \end{cases}$$

The call of $T_s^k(j)$ yields the largest time of the set $\mathcal{B}^s(k) \cup \{k\}$ which is smaller than j , or returns -1 if there is no such time. Thus, given $t_n = k$ and a bound j on the visit time for the pickup s , we are able to iterate over the candidate times for s in $\mathcal{B}^s(k) \cup \{k\}$.

Thereby, we can define a recursion table calculating the minimum cost of a schedule t when the value of t_n is fixed.

Lemma 7 (*). *Let I be an FPTD instance, $k \in \mathcal{C}$, and \mathbf{t} a schedule of I such that $t_n = k$ and \mathbf{t} has minimum cost. Then, $c(\mathbf{t}) = C[n, k] + \sum_{d \in \mathcal{S}_{>n}} \sigma_d^L(\max\{k, a_d\})$*

$$C[i, j] = \begin{cases} \min \left\{ C[i-1, T_{i-1}^k(j)] + \sigma_i^k(j), \right. & \text{if } i \geq 1, j \geq a_i, \\ C[i, T_i^k(j-1)] & \left. \right\} \\ 0 & \text{if } i = 0, \\ \infty & \text{otherwise.} \end{cases} \quad (2)$$

Finally, to find the cost of an optimal schedule, we have to compute the value $C[n, k]$ for each $k \in \mathcal{C}$. A dynamic programming algorithm (*) can solve it in $\mathcal{O}(n^4)$ time.

Theorem 3 (*). *An FPTD instance I with $2n$ stops can be solved in $\mathcal{O}(n^4)$.*

6 Conclusion

We study a new model of the Dial-A-Ride Problem for the scheduling of fixed sequences with several time constraints typical in the Pickup-and-Delivery scenario. We highlight the key role of soft maximal ride time constraints in the combinatorial complexity of the problem, as they induce the NP-hardness of the problem. Then, we prove that if the maximal ride times are bounded by a constant, we can obtain a polynomial-time algorithm. Finally, we show that instances of the problem with a special structure can be solved efficiently, independently of the timing constraints. We believe that this result can be generalised whenever the number of times a pickup is followed by a delivery in the sequence is bounded.

To get a better understanding of the overall complexity of the problem further research may consider other constraints, *e.g.* allow earliness at stops, and more complex penalty functions.

References

1. Atallah, M.J., Chen, D.Z., Lee, D.T.: An optimal algorithm for shortest paths on weighted interval and circular-arc graphs, with applications. *Algorithmica* **14**(5), 429–441 (1995). <https://doi.org/10.1007/BF01192049>
2. Cordeau, J.F., Laporte, G.: The dial-a-ride problem (DARP): Variants, modeling issues and algorithms. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies* **1**(2), 89–101 (2003)
3. Crescenzi, P.: A short guide to approximation preserving reductions. In: *Proceedings of the 12th Annual IEEE Conference on Computational Complexity*. pp. 262–273. CCC '97, IEEE Computer Society, Washington, DC, USA (1997)
4. Desrosiers, J., Dumas, Y., Solomon, M.M., Soumis, F.: Time constrained routing and scheduling. *Handbooks in operations research and management science* **8**, 35–139 (1995)
5. Dumas, Y., Soumis, F., Desrosiers, J.: Optimizing the schedule for a fixed vehicle path with convex inconvenience costs. *Transportation Science* **24**(2), 145–152 (1990)
6. Firat, M., Woeginger, G.J.: Analysis of the dial-a-ride problem of Hunsaker and Savelsbergh. *Operations Research Letters* **39**(1), 32–35 (2011)
7. Gatto, M., Jacob, R., Peeters, L., Schöbel, A.: The computational complexity of delay management. In: *International Workshop on Graph-Theoretic Concepts in Computer Science*. pp. 227–238. Springer (2005)
8. Gschwind, T.: Route feasibility testing and forward time slack for the synchronized pickup and delivery problem. *Tech. rep.* (2015)
9. Hunsaker, B., Savelsbergh, M.: Efficient feasibility testing for dial-a-ride problems. *Operations research letters* **30**(3), 169–173 (2002)
10. Lampis, M., Kaouri, G., Mitsou, V.: On the algorithmic effectiveness of digraph decompositions and complexity measures. *Discrete Optimization* **8**(1), 129–138 (2011). <https://doi.org/10.1016/j.disopt.2010.03.010>
11. Tang, J., Kong, Y., Lau, H., Ip, A.W.: A note on efficient feasibility testing for dial-a-ride problems. *Operations Research Letters* **38**(5), 405–407 (2010)
12. Tarjan, R.E.: Edge-disjoint spanning trees and depth-first search. *Acta Informatica* **6**(2), 171–185 (1976)

7 Additional proofs

7.1 Driving and (Un)Loading Times

In favour of simplicity, our model neglects times needed to travel between stops as well as loading or unloading times. We show in the following that incorporating these into an extended problem definition is not more expressive. Suppose X is an instance of the *extended* MIN PDS, where for each stop $s \in \mathbb{S}$ we are additionally given a driving time d_s to reach stop $s + 1$ (assume $d_{2n} = 0$ for the last stop) and (un)loading time v_s at stop s . Further a schedule for X is only feasible if for $s \in \mathbb{S}_{<2n}$ it holds the strengthened inequality $\mathbf{t}_s + v_s + d_s \leq \mathbf{t}_{s+1}$. Lastly, lateness penalties and ride time penalties factor in the visiting times, such that the cost $\widehat{c}(\mathbf{t})$ of a schedule \mathbf{t} of X is defined as

$$\widehat{c}(\mathbf{t}) := \sum_{s \in \mathbb{S}} \sigma_s^L(\mathbf{t}_s + v_s) + \sum_{(p,d) \in \mathcal{P}} \sigma_{p,d}^{RT}(\mathbf{t}_d + v_d - \mathbf{t}_p). \quad (3)$$

Lemma 8. *Let X be an instance of the extended MIN PDS and I_X its corresponding instance of MIN PDS. Then there exists a one-to-one cost-preserving mapping between the schedules of X and I_X . I_X can be constructed in linear time.*

Proof. For a stop $\ell \in \mathbb{S}$ denote $\Gamma_\ell := \sum_{s < \ell} (d_s + v_s)$. Define an instance I_X of MIN PDS with the same sequence \mathbb{S} of the stops, the same set of requests, and the same penalty function coefficients as X . For all $s \in \mathbb{S}$ set time windows as $a'_s := a_s - \Gamma_s$ and $b'_s := b_s - \Gamma_s - v_s$. The maximal ride times are defined as $r'_{p,d} := r_{p,d} - \Gamma_d + \Gamma_p - v_d$ for all $(p,d) \in \mathcal{P}$. Note that all adjustments of this kind can be performed in a single preprocessing pass over the extended instance X in linear time.

Now we show there is a one-to-one correspondence between schedules of X and I_X . For \mathbf{t} a schedule of X , let $f(\mathbf{t})$ map to a schedule $\mathbf{t}' = f(\mathbf{t})$ of I_X such that for $s \in \mathbb{S}$, $\mathbf{t}'_s = \mathbf{t}_s - \Gamma_s$. Note that f is bijective.

First, we show that \mathbf{t} is feasible for X exactly when \mathbf{t}' is feasible for I_X . For $s \in \mathbb{S}$ we have $\mathbf{t}_s \geq a_s \Leftrightarrow \mathbf{t}_s - \Gamma_s \geq a_s - \Gamma_s \Leftrightarrow \mathbf{t}'_s \geq a'_s$; for $s \in \mathbb{S}_{<2n}$ we have

$$\mathbf{t}_s + v_s + d_s \leq \mathbf{t}_{s+1} \Leftrightarrow \mathbf{t}_s - \Gamma_s = \mathbf{t}_s + v_s + d_s - \Gamma_{s+1} \leq \mathbf{t}_{s+1} - \Gamma_{s+1} \Leftrightarrow \mathbf{t}'_s \leq \mathbf{t}'_{s+1}.$$

Now, we show that the penalty cost of \mathbf{t} for X equals the penalty cost of \mathbf{t}' for I_X . We have

$$\begin{aligned} \widehat{c}(\mathbf{t}) &= \sum_{s \in \mathbb{S}} \sigma_s^L(\mathbf{t}_s + v_s) + \sum_{(p,d) \in \mathcal{P}} \sigma_{p,d}^{RT}(\mathbf{t}_d + v_d - \mathbf{t}_p) \\ &= \sum_{\substack{s \in \mathbb{S} \\ \mathbf{t}_s + v_s > b_s}} \alpha_s(\mathbf{t}_s + v_s - b_s) + \beta_s + \sum_{\substack{(p,d) \in \mathcal{P} \\ \mathbf{t}_d + v_d - \mathbf{t}_p > r_{p,d}}} \alpha_{p,d}(\mathbf{t}_d + v_d - \mathbf{t}_p - r_{p,d}) + \beta_{p,d} \\ &= \sum_{\substack{s \in \mathbb{S} \\ \mathbf{t}'_s > b'_s}} \alpha_s(\mathbf{t}'_s - b'_s) + \beta_s + \sum_{\substack{(p,d) \in \mathcal{P} \\ \mathbf{t}'_d - \mathbf{t}'_p > r'_{p,d}}} \alpha_{p,d}(\mathbf{t}'_d - \mathbf{t}'_p - r'_{p,d}) + \beta_{p,d} \quad (4) \\ &= \sum_{s \in \mathbb{S}} \sigma_s^L(\mathbf{t}'_s) + \sum_{(p,d) \in \mathcal{P}} \sigma_{p,d}^{RT}(\mathbf{t}'_d - \mathbf{t}'_p) = c(\mathbf{t}) \end{aligned}$$

To see Eq. (4), note that $t_s + v_s - b_s = t'_s - b'_s$ and $t_d + v_d - t_p - r_{p,d} = t'_d - t'_p - r'_{p,d}$ hold. These can be applied to the penalty function arguments as well as the sum ranges.

7.2 Proof of Theorem 2

Proof (Theorem 2). For a given instance I of MIN PDS-HRT consider the bijection $f : x \mapsto 2n - x + 1$ on S . Define the *backward* instance I' with the same sequence of stops and with requests $\mathcal{P}' := \{(f(d), f(p)) : (p, d) \in \mathcal{P}\}$. The time windows are defined by $a'_s := 0$ and $b'_s := a_{2n} - a_{f(s)}$ for all $s \in S$. The ride time constraints are $r'_{f(d),f(p)} := r_{p,d}$. Notice that a feasible schedule of I' always exists, *e.g.* $(0, \dots, 0)$ is a feasible solution. However, solving the instance I' with Firat and Woeginger's algorithm [6] yields a feasible schedule t' of I' such that every stop is scheduled as late as possible to be still feasible. We can translate t' to a schedule t of I by setting $t_s := a_{2n} - t'_{f(s)}$ for all $s \in S$. Then t may violate lateness constraints of I , but we will show that it is feasible for I . Firstly, for $s \in S$, we get $t_s = a_{2n} - t'_{f(s)} \geq a_{2n} - b'_{f(s)} = a_{2n} - a_{2n} + a_s = a_s$. When $s \in S_{<2n}$, we get $t_s = a_{2n} - t'_{f(s)} = a_{2n} - t'_{2n-s+1} \leq a_{2n} - t'_{2n-s} = a_{2n} - t'_{f(s+1)} = t_{s+1}$. Lastly, for $(p, d) \in \mathcal{P}$, we have $t_d - t_p = t'_{f(p)} - t'_{f(d)} \leq r'_{f(p),f(d)} = r_{p,d}$. Therefore t is a feasible schedule for I . As it was noted earlier, t' visits every stop 'as late as possible' to be still feasible. Since all lateness penalties are non-decreasing, t is an optimal schedule for I .

7.3 Proof of Lemma 1

The essential idea of the proof is shifting of the visit times in a schedule without increasing its cost until the schedule is defined in $\widetilde{\mathcal{W}}$. For this purpose we define a so called *closure* of each stop. The closure is a subset of stops for which the timings will be adjusted together.

Definition 2. Let I be an instance of MIN PDS and t be a feasible schedule of I . For a stop $s \in S$ the closure \mathcal{G}_s^t is the minimal subset of S fulfilling the following:

- (a) $s \in \mathcal{G}_s^t$,
- (b) for $x \in \mathcal{G}_s^t$ and $y \in S$ with $t_x = t_y$ also $y \in \mathcal{G}_s^t$,
- (c) for each $(p, d) \in \mathcal{P}$ such that $t_d - t_p \leq r_{p,d}$ and $\{p, d\} \cap \mathcal{G}_s^t \neq \emptyset$ then all stops $x \in S$ with $p \leq x \leq d$ must be in \mathcal{G}_s^t .

Now we show that we can modify the visit times of all stops included in a closure without violating any new ride time constraints.

Lemma 9. Let t be a feasible schedule of an instance I of MIN PDS and $\ell \in S$ be a stop with closure \mathcal{G}_ℓ^t . For a fixed $\delta \in \{-1, 1\}$ let t' be a schedule such that $t'_s = t_s + \delta$ for all $s \in \mathcal{G}_\ell^t$ and $t'_s = t_s$ for all $s \in S \setminus \mathcal{G}_\ell^t$. Then:

- (i) t' satisfies all ride time constraints satisfied by t and
- (ii) for all stops $s \in S_{<2n}$ it holds $t'_s \leq t'_{s+1}$.

Proof. (i) As it follows from Definition 2(c), for all $(p, d) \in \mathcal{P}$ with $\mathbf{t}_d - \mathbf{t}_p \leq r_{p,d}$, either both p and d are contained in \mathcal{G}_ℓ^t or neither of them. If $p, d \notin \mathcal{G}_\ell^t$ then $\mathbf{t}'_d - \mathbf{t}'_p = \mathbf{t}_d - \mathbf{t}_p \leq r_{p,d}$. If $p, d \in \mathcal{G}_\ell^t$ then $\mathbf{t}'_d - \mathbf{t}'_p = \mathbf{t}_d + \delta - \mathbf{t}_p - \delta = \mathbf{t}_d - \mathbf{t}_p \leq r_{p,d}$. Hence in both cases \mathbf{t}' respects the ride time constraints.

(ii) Fix $s \in \mathcal{S}_{<2n}$. Since \mathbf{t} is feasible it holds $\mathbf{t}_s \leq \mathbf{t}_{s+1}$. By $\delta \in \{-1, 1\}$, $\mathbf{t}'_s > \mathbf{t}'_{s+1}$ would imply $\mathbf{t}_s = \mathbf{t}_{s+1}$ and $\{s, s+1\} \cap \mathcal{G}_\ell^t = 1$, which conflicts with the Definition 2(b).

Using Lemma 9 we can shift all visit times of stops in a closure until one of the stops reaches a time window border. As it follows from the following lemma this implies that all stops in the closures are *close* to this border and thus in \mathcal{W} .

Lemma 10. *Let \mathbf{t} be a feasible schedule of an instance I of μ -MIN PDS and let $s \in \mathcal{S}$. If \mathcal{G}_s^t contains at least two stops, say x, y , and the stop x precedes y , i.e. $x < y$, then:*

- (i) *if the stop y immediately follows the stop x , i.e. $y = x+1$, then $t_y \leq t_x + \mu$, and*
- (ii) *$t_y \leq t_x + n\mu$.*

Proof. (i) Let $y = x + 1$. By the definition of \mathcal{G}_s^t , either $\mathbf{t}_x = \mathbf{t}_y$, in which case (i) follows, or there exists $p, d \in \mathcal{G}_s^t$ such that $(p, d) \in \mathcal{P}$, $\mathbf{t}_d - \mathbf{t}_p \leq r_{p,d}$, and $p \leq x < y \leq d$. By the feasibility of \mathbf{t} we have $\mathbf{t}_p \leq \mathbf{t}_x$ and $\mathbf{t}_y \leq \mathbf{t}_d$ and $\mathbf{t}_d - \mathbf{t}_p \leq r_{p,d} \leq \mu$. Therefore, $\mathbf{t}_y \leq \mathbf{t}_d \leq \mathbf{t}_p + \mu \leq \mathbf{t}_x + \mu$.

(ii) If all stops in \mathcal{G}_s^t are scheduled at the same time, the claim obviously holds. Otherwise, we consider the partition of \mathcal{G}_s^t into nonempty sets G_1, \dots, G_k , where for all $i \leq k$ and all $x, y \in G_i$ it holds $\mathbf{t}_x = \mathbf{t}_y =: \mathbf{t}(G_i)$ and for $i < j \leq k$ it holds $\mathbf{t}(G_i) < \mathbf{t}(G_j)$. Due to the properties of \mathcal{G}_s^t , for each $i < k$ there exists $i < j \leq k$ such that there is a $(p, d) \in \mathcal{P}$ with $p \in G_i$, $d \in G_j$, and $\mathbf{t}(G_j) \leq \mathbf{t}(G_i) + r_{p,d} \leq \mathbf{t}(G_i) + \mu$. This leads to the observation, that for all $i < k$ we have $\mathbf{t}(G_{i+1}) \leq \mathbf{t}(G_i) + \mu$. Thus, we get $\mathbf{t}(G_k) \leq \mathbf{t}(G_1) + k\mu$. We note that since for each $i < k$ there is a pickup stop in G_i , it is $k \leq n + 1$ and thus $\mathbf{t}(G_k) \leq \mathbf{t}(G_1) + n\mu$. Obviously $\mathbf{t}_x \geq \mathbf{t}(G_1)$ and $\mathbf{t}_y \leq \mathbf{t}(G_k)$ which completes the proof.

Definition 3. *Let I be an instance of μ -MIN PICKUP-DELIVERY SCHEDULING and \mathbf{t} be a schedule of I . Let $\mathcal{L}_t := \{s \in \mathcal{S} : \mathbf{t}_s > b_s\}$ be the set containing all stops of \mathcal{S} in which \mathbf{t} violates the lateness constraint. Similarly, let $\mathcal{R}_t := \{(p, d) \in \mathcal{P} : \mathbf{t}_d - \mathbf{t}_p > r_{p,d}\}$ be the set of all requests from \mathcal{P} for which \mathbf{t} violates the ride time constraints.*

Proof (Lemma 1). We prove by contradiction. Assume the lemma does not hold, then for every optimal schedule \mathbf{t} of I there is a stop $\ell \in \mathcal{S}$ such that $\mathbf{t}_\ell \notin \widetilde{\mathcal{W}}$. Let \mathbf{t}^* be an optimal schedule maximising $\ell \in \mathcal{S}$ such that $\mathbf{t}_s^* \in \widetilde{\mathcal{W}}$ for all $s < \ell$ and among all such schedulings \mathbf{t}_s^* is chosen to maximise \mathbf{t}_ℓ^* .

Let $\mathcal{G} := \mathcal{G}_\ell^{t^*}$ be the closure of the stop ℓ . If there is $g \in \mathcal{G}$ with $\mathbf{t}_g^* \in \mathcal{W}$, then by Lemma 10 (ii), $|\mathbf{t}_\ell^* - \mathbf{t}_g^*| \leq n\mu$ and therefore also $\mathbf{t}_\ell^* \in \widetilde{\mathcal{W}}$ which contradicts the choice of ℓ . Therefore for all $g \in \mathcal{G}$ necessarily $\mathbf{t}_g^* \notin \mathcal{W}$ and since \mathbf{t}^* is feasible, also $\mathbf{t}_g^* > a_g$.

Now define a schedule \mathbf{t}^- such that we set $\mathbf{t}_s^- := \mathbf{t}_s^* - 1$ for $s \in \mathcal{G}$ and $\mathbf{t}_s^- := \mathbf{t}_s^*$ otherwise. Note that \mathbf{t}^- is feasible since for all $g \in \mathcal{G}$ it holds $\mathbf{t}_g^* > a_g$ and for each stop s , $\mathbf{t}_s^- \leq \mathbf{t}_{s+1}^-$ due to Lemma 9 (ii). Due to Lemma 9 (i) $\mathcal{R}_{\mathbf{t}^-} \subseteq \mathcal{R}_{\mathbf{t}^*}$. Denote $L := \mathcal{L}_{\mathbf{t}^*} \cap \mathcal{G}$, $R_1 := \{(p, d) \in \mathcal{R}_{\mathbf{t}^*} : p \notin \mathcal{G}, d \in \mathcal{G}\}$, and $R_2 := \{(p, d) \in \mathcal{R}_{\mathbf{t}^*} : p \in \mathcal{G}, d \notin \mathcal{G}\}$, then

$$c(\mathbf{t}^-) = c(\mathbf{t}^*) + \sum_{x \in R_2} \alpha_x - \sum_{x \in R_1} \alpha_x - \sum_{x \in \mathcal{R}_{\mathbf{t}^*} \setminus \mathcal{R}_{\mathbf{t}^-}} \beta_x - \sum_{s \in L} \alpha_s - \sum_{s \in \mathcal{L}_{\mathbf{t}^*} \setminus \mathcal{L}_{\mathbf{t}^-}} \beta_s$$

By the optimality of \mathbf{t}^* and feasibility of \mathbf{t}^- we have $c(\mathbf{t}^-) \geq c(\mathbf{t}^*)$, yielding

$$\sum_{x \in R_2} \alpha_x \geq \sum_{s \in L} \alpha_s + \sum_{x \in R_1} \alpha_x \quad (5)$$

Now define the schedule \mathbf{t}^+ as $\mathbf{t}_s^+ := \mathbf{t}_s^* + 1$ for $s \in \mathcal{G}$ and $\mathbf{t}_s^+ := \mathbf{t}_s^*$ otherwise. According to Lemma 9 (ii), \mathbf{t}^+ is feasible. Since $\mathbf{t}_g^* \neq b_g \in \mathcal{W}$ for all $g \in \mathcal{G}$, no new late stop visit is introduced in \mathbf{t}^+ . Also no new ride time violations are introduced as it follows from Lemma 9 (i). Therefore we have

$$c(\mathbf{t}^+) \leq c(\mathbf{t}^*) + \sum_{x \in R_1} \alpha_x + \sum_{s \in L} \alpha_s - \sum_{x \in R_2} \alpha_x \stackrel{\text{by Eq. (5)}}{\leq} c(\mathbf{t}^*).$$

This means \mathbf{t}^+ is also optimal. Furthermore for all stops $s \in \mathcal{S}$ for which $s < \min \mathcal{G}$ it holds $\mathbf{t}_s^+ = \mathbf{t}_s^* \in \widetilde{\mathcal{W}}$. Now due to the choice of \mathbf{t}^* and ℓ , $\mathbf{t}_\ell^+ \notin \widetilde{\mathcal{W}}$. But $\mathbf{t}_\ell^+ > \mathbf{t}_\ell^*$ which contradicts the choice of \mathbf{t}^* regarding the maximum value \mathbf{t}_ℓ^* .

7.4 Proof of Lemma 2

Proof (Lemma 2). (i) and (iii) follow directly from the definition of the partial cost \tilde{c} .

Regarding (ii), we note that for $\ell \in \mathcal{S}_{<2n}$ it is $\mathcal{J}_{\ell+1} = \mathcal{J}_\ell \cup \{(\ell+1, d)\}$ if $(\ell+1)$ is a pickup stop and $(\ell+1, d) \in \mathcal{P}$, or $\mathcal{J}_{\ell+1} = \mathcal{J}_\ell \setminus \{(p, \ell+1)\}$ if $(\ell+1)$ is a delivery stop, $(p, \ell+1) \in \mathcal{P}$. Therefore,

$$\begin{aligned} \sum_{(p,d) \in \mathcal{J}_{\ell+1}} \sigma_{p,d}^{RT}(\mathbf{t}_{\ell+1} - \mathbf{t}_p) &= \sum_{(p,d) \in \mathcal{J}_\ell} \sigma_{p,d}^{RT}(\mathbf{t}_{\ell+1} - \mathbf{t}_p) + \sigma_{\ell+1,d}^{RT}(\mathbf{t}_{\ell+1} - \mathbf{t}_{\ell+1}) \\ &= \sum_{(p,d) \in \mathcal{J}_\ell} \sigma_{p,d}^{RT}(\mathbf{t}_{\ell+1} - \mathbf{t}_p) \end{aligned}$$

or

$$\sum_{(p,d) \in \mathcal{J}_{\ell+1}} \sigma_{p,d}^{RT}(\mathbf{t}_{\ell+1} - \mathbf{t}_p) = \sum_{(p,d) \in \mathcal{J}_\ell} \sigma_{p,d}^{RT}(\mathbf{t}_{\ell+1} - \mathbf{t}_p) - \sigma_{p,\ell+1}^{RT}(\mathbf{t}_{\ell+1} - \mathbf{t}_p).$$

Thus

$$\begin{aligned}
\tilde{c}(\mathbf{t}, \ell + 1) &= \sum_{s \in \mathcal{S}_{\leq \ell+1}} \sigma_s^L(\mathbf{t}_s) + \sum_{(p,d) \in \mathcal{J}_{\ell+1}} \sigma_{p,d}^{RT}(\mathbf{t}_{\ell+1} - \mathbf{t}_p) + \sum_{\substack{(p,d) \in \mathcal{P} \\ d \leq \ell+1}} \sigma_{p,d}^{RT}(\mathbf{t}_d - \mathbf{t}_p) \\
&= \sum_{s \in \mathcal{S}_{\leq \ell+1}} \sigma_s^L(\mathbf{t}_s) + \sum_{(p,d) \in \mathcal{J}_{\ell}} \sigma_{p,d}^{RT}(\mathbf{t}_{\ell+1} - \mathbf{t}_p) + \sum_{\substack{(p,d) \in \mathcal{P} \\ d \leq \ell}} \sigma_{p,d}^{RT}(\mathbf{t}_d - \mathbf{t}_p) \\
&= \tilde{c}(\mathbf{t}, \ell) + \sigma_{\ell+1}^L(\mathbf{t}_{\ell+1}) + \sum_{(p,d) \in \mathcal{J}_{\ell}} (\sigma_{p,d}^{RT}(\mathbf{t}_{\ell+1} - \mathbf{t}_p) - \sigma_{p,d}^{RT}(\mathbf{t}_{\ell} - \mathbf{t}_p)).
\end{aligned}$$

For $(p, d) \in \mathcal{J}_{\ell}$ we have $\sigma_{p,d}^{RT}(\mathbf{t}_{\ell} - \mathbf{t}_p) = 0$ if $\mathbf{t}_{\ell} - \mathbf{t}_p \leq r_{p,d}$, otherwise $\sigma_{p,d}^{RT}(\mathbf{t}_{\ell+1} - \mathbf{t}_p) - \sigma_{p,d}^{RT}(\mathbf{t}_{\ell} - \mathbf{t}_p) = \alpha_{p,d}(\mathbf{t}_{\ell+1} - \mathbf{t}_p) + \beta_{p,d} - \alpha_{p,d}(\mathbf{t}_{\ell} - \mathbf{t}_p) - \beta_{p,d} = \alpha_{p,d}(\mathbf{t}_{\ell+1} - \mathbf{t}_{\ell})$. Thus

$$\sum_{(p,d) \in \mathcal{J}_{\ell}} (\sigma_{p,d}^{RT}(\mathbf{t}_{\ell+1} - \mathbf{t}_p) - \sigma_{p,d}^{RT}(\mathbf{t}_{\ell} - \mathbf{t}_p)) = \sum_{(p,d) \in \mathcal{J}_{\ell}} f_{p,d}^{\ell}(\mathbf{t}),$$

which concludes the proof.

7.5 Proof of Lemma 3

Proof (Lemma 3). Let \mathbf{t}^2 be a feasible schedule with $Label_{\ell}(\mathbf{t}^2) = L^2$ which minimises $c(\mathbf{t}^2)$. Let \mathbf{t}^1 be a feasible schedule with $Label_{\ell}(\mathbf{t}^1) = L^1$. Let \mathbf{t} be the schedule defined as $\mathbf{t}_s := \mathbf{t}_s^1$ for $s \in \mathcal{S}_{< \ell}$ and $\mathbf{t}_s := \mathbf{t}_s^2$ for $s \in \mathcal{S}_{\geq \ell}$. By this definition (and $\mathbf{t}_{\ell} = \mathbf{t}_{\ell}^2 = \tau = \mathbf{t}_{\ell}^1$), $Label_{\ell}(\mathbf{t}) = L^1$.

Now we show inductively that $\tilde{c}(\mathbf{t}, i) \leq \tilde{c}(\mathbf{t}^2, i)$ for all i , $\ell \leq i \leq 2n$. The case $i = \ell$ holds due to $\tilde{c}_1 \leq \tilde{c}_2$. For the induction step we assume $\tilde{c}(\mathbf{t}, i) \leq \tilde{c}(\mathbf{t}^2, i)$ holds for every i , $\ell \leq i < 2n$ and now we show also validity for $i + 1$. Using Lemma 2 (ii) we have

$$\tilde{c}(\mathbf{t}, i + 1) = \tilde{c}(\mathbf{t}, i) + \sigma_{i+1}^L(\mathbf{t}_{i+1}) + \sum_{(p,d) \in \mathcal{J}_i} f_{p,d}^i(\mathbf{t}).$$

By the definition of \mathbf{t} we have $\mathbf{t}_i = \mathbf{t}_i^2$ and $\mathbf{t}_{i+1} = \mathbf{t}_{i+1}^2$. Clearly $\sigma_{i+1}^L(\mathbf{t}_{i+1}) = \sigma_{i+1}^L(\mathbf{t}_{i+1}^2)$. Next, we show $f_{p,d}^i(\mathbf{t}) = f_{p,d}^i(\mathbf{t}^2)$ for all $(p, d) \in \mathcal{J}_i$. Let $(p, d) \in \mathcal{J}_i$.

- If $p \geq s^{\mu}$ we show that $\mathbf{t}_p = \mathbf{t}_p^2$. For contradiction we suppose $\mathbf{t}_p \neq \mathbf{t}_p^2$ and without loss of generality $\mathbf{t}_p > \mathbf{t}_p^2$. If $m := \mathbf{t}_{\ell} - \mathbf{t}_p$, then following the schedule \mathbf{t} it must hold $s^m \leq p$, but following the schedule \mathbf{t}^2 also $s^m > p$, a contradiction. Since \mathbf{t} and \mathbf{t}^2 schedule such stop p at the same time, $f_{p,d}^i(\mathbf{t}) = f_{p,d}^i(\mathbf{t}^2)$.
- If $p < s^{\mu}$, it means $\mathbf{t}_p < \mathbf{t}_{\ell} - \mu \leq \mathbf{t}_i - r_{p,d}$ and this leads to $f_{p,d}^i(\mathbf{t}) = \alpha_{p,d}(\mathbf{t}_{i+1} - \mathbf{t}_i)$. The same arguments ensure $f_{p,d}^i(\mathbf{t}^2) = \alpha_{p,d}(\mathbf{t}_{i+1}^2 - \mathbf{t}_i^2)$. Due to $\mathbf{t}_i = \mathbf{t}_i^2$ and $\mathbf{t}_{i+1} = \mathbf{t}_{i+1}^2$ we conclude $f_{p,d}^i(\mathbf{t}) = f_{p,d}^i(\mathbf{t}^2)$.

Together with the induction hypothesis we can see that $\tilde{c}(\mathbf{t}, i + 1) \leq \tilde{c}(\mathbf{t}^2, i + 1)$ and together with Lemma 2 (iii), the case $i = 2n$ concludes the proof.

7.6 Proof of the application of Dumas et al. algorithm [5] for minimising lateness penalties and sum of the ride times

Let I be an instance of MIN PDS. The cost of a schedule t of I minimising the sum of the lateness penalties and the sum of the ride times is

$$c(t) = \sum_{s \in S} \sigma_s^L(t_s) + \sum_{(p,d) \in \mathcal{P}} (t_d - t_p). \quad (6)$$

Since there is no dependence with the ride time constraint, we can set $\sigma_{p,d}^{RT}(l) = l$, for all $(p,d) \in \mathcal{P}$, and then Equations 1 and 6 are equivalent. Notice that this is a special case of μ -MIN PDS where $\mu = r_{p,d} = 0$, $\alpha_{p,d} = 1$ and $\beta_{p,d} = 0$ for all request $(p,d) \in \mathcal{P}$. We point out that, since the ride time constraints are zero, the cost of a waiting time is directly dependent on the number of requests affected by it, which is exactly the load of the vehicle. Then for a schedule t , we have

$$\sum_{(p,d) \in \mathcal{P}} t_d - t_p = \sum_{s \in S < 2n} \text{load}(s) \cdot (t_{s+1} - t_s). \quad (7)$$

Since this can be seen as penalised waiting times the problem can be solved with the algorithm of Dumas et al. [5] in linear time.

7.7 Proof of Lemma 6

Definition 4. Let t be a schedule of an instance I of MIN PDS and $X \subseteq S$. We say that X is extended if for all $s, s' \in S$, if $s \in X$ and $t_s = t_{s'}$ then $s' \in X$.

Definition 5. Let t be a schedule of an instance I of MIN PDS and $X \subseteq S$ be extended. Let $t(X) := \{t_s : s \in X\}$ be the set of different visit times for the stops in X by t . We define altered schedules t^{X+} and t^{X-} by setting $t_s^{X+} = t_s + 1$ and $t_s^{X-} = \max\{a_s, t_s - 1\}$ if $t_s \in t(X)$ and $t_s^{X+} = t_s^{X-} = t_s$ otherwise. Note that if t is a feasible schedule then both t^{X+} and t^{X-} are feasible schedules.

We also write t^{s+} and t^{s-} as shorthands for t^{X+} and t^{X-} with $X = \{x \in S : t_x = t_s\}$.

Lemma 11. Let I be an FPTD instance with $2n$ stops, t a feasible schedule for I , and $s \leq n$ a pickup stop with $t_s < t_{n+1}$. Then t^{s+} violates no ride time constraint respected by t .

The above lemma is straightforward to see since no pickups are scheduled earlier and no deliveries are scheduled later in t^{s+} compared to t .

Lemma 12. Let I be an instance of MIN PDS, t be a schedule of I , and $X \subseteq S$ be extended and such that $t_s > a_s$ for all $s \in X$. If $2c(t) < c(t^{X-}) + c(t^{X+})$ then t^{X-} or t^{X+} must violate a constraint satisfied by t .

Proof. To prove a contradiction, we suppose that t^{X-} and t^{X+} only violate a subset of the constraints violated in t and that $2c(t) < c(t^{X-}) + c(t^{X+})$. We define the following sets:

- $X^L := \{x \in X : t_x > b_x\}$,
- $X^P := \{(p, d) \in \mathcal{P} : p \in X, d \notin X, t_d - t_p > r_{p,d}\}$,
- $X^D := \{(p, d) \in \mathcal{P} : d \in X, p \notin X, t_d - t_p > r_{p,d}\}$.

Let

$$\Delta := \sum_{x \in X^L} \alpha_x^L - \sum_{(p,d) \in X^P} \alpha_{p,d}^{RT} + \sum_{(p,d) \in X^D} \alpha_{p,d}^{RT}.$$

Since we suppose that t^{X^-} and t^{X^+} only violate subsets of constraints compared to t , we have $c(t^{X^-}) \leq c(t) - \Delta$ (note that this makes use of $t_s > a_s$ for all $s \in X$ since otherwise $t_s^{X^-} = t_s$) and $c(t^{X^+}) \leq c(t) + \Delta$. By summing these inequalities, we obtain $2c(t) \geq c(t^{X^-}) + c(t^{X^+})$, a contradiction.

Proof (Lemma 6). We consider an optimal schedule t of an *FPTD* instance I . Among all optimal schedules of I , t is chosen

- (a) to minimise $\sum_{(p,d) \in \mathcal{P}} t_d$, the sum of the delivery times, and such that
- (b) for all $s \in S_{\leq n}$ with $t_s < t_{n+1}$, any feasible schedule t' with $t'_s > t_s$ has $c(t') > c(t)$.

In the following we show that the schedule t has the requested properties. Due to assumptions about our model discussed in Subsection 2.2, we obviously have $t_{2n} = a_{2n}$ and following (a) the scheduling must use ‘as soon as possible delivery’ strategy, *i.e.* $t_s = \max\{t_n, a_s\}$, for every $s \in S_{> n}$.

Let $s \in S$. Notice that both t^{s-} and t^{s+} are feasible schedules. Since t is optimal, obviously $c(t^{s-}) \geq c(t)$. We discuss separately the different options how t_s , t_n , and t_{n+1} can be related and make a conclusion in each case.

Firstly, we assume $t_s < t_n$, thus $t_s < t_n \leq t_{n+1}$. Consider the following:

- if t^{s+} violates a constraint satisfied in t , then according to Lemma 11, it must be a time window constraint and thus there is a stop $s' \in S_{\geq s}$ with $b_{s'} = t_s$, hence $t_s \in \{b_{s'} : s' \in S, s \leq s' \leq n\}_{< t_n, \geq a_s} \subseteq \mathcal{B}^s(t_n)$;
- if t^{s-} violates a constraint satisfied in t , then it must be a ride time constraint, and there is a stop $p \in S$, $(p, d) \in \mathcal{P}$, such that $t_d - r_{p,d} = t_p = t_s$, hence $t_s \in \bigcup_{(p,d) \in \mathcal{P}, p \leq s} \{\max\{t_n, a_d\} - r_{p,d}\}_{\geq a_s} \subseteq \mathcal{B}^s(t_n)$.

Secondly, assume that $t_s = t_n = t_{n+1}$. Let K such that $\{s\} \cup \{p : (p, d) \in \mathcal{P}, t_p = t_d - r_{p,d}, t_d = t_s\} \subseteq K$ and extend it if necessary. If $t_\ell = a_\ell = t_n$ for $\ell \in K$, we also have $t_n = a_n \in \mathcal{C}$. If on the other hand $t_p = a_p < t_n$ for some $p \in K$, then p is a pickup. By (b) $c(t^{p+}) > c(t)$, which means there is a stop $\ell \geq p$ with $t_\ell = t_p$ and $t_\ell \geq b_\ell$. Also $t_\ell = t_p = a_p \leq a_\ell \leq b_\ell$, concluding $t_\ell = b_\ell$. Then $t_n = b_\ell + r_{p,d} \in \mathcal{C}$. On the other hand, when all $\ell \in K$ have $t_\ell > a_\ell$, we can make use of Lemma 12. By (a) and $(n+1) \in K$, it must hold $c(t^{K-}) > c(t)$. Further, by optimality of t , $c(t) \leq c(t^{K+})$, and according to Lemma 12 a constraint satisfied in t is violated by t^{K-} or t^{K+} :

- if t^{K-} violates a constraint satisfied in t , then it must be a maximal ride time constraint of a request $(p, d) \in \mathcal{P}$ with $p \in K, d \notin K$. Then $t_d > t_n$ and by (a) it is $t_d = a_d$ and $t_p = a_d - r_{p,d}$. Further, by definition of K , either $t_n = t_p$ or $t_n = t_p + r_{p',d'}$ for some $(p', d') \in \mathcal{P}$. We conclude $t_s = t_n \in \mathcal{C}$.
- if t^{K+} violates a constraint satisfied in t we can see that it cannot be a ride time constraint, since for all $(p, d) \in \mathcal{P}$ with $d \in K$ and $t_d = t_p + r_{p,d}$, also

$p \in K$. Therefore, it is a time window constraint, and $t_s = b_\ell$ for an $\ell \in S$ or $t_s = b_p + r_{p,d}$ for a $(p, d) \in \mathcal{P}$, thus $t_s \in \mathcal{C}$.

Thirdly, assume $t_s = t_n < t_{n+1}$, then s is a pickup stop. Obviously, $t_n < a_{n+1}$ and due to (a) $t_d = a_d$ for every delivery $d \in S_{>n}$. If $t_s = a_\ell$ for some $\ell \leq n$, then by $a_n \geq a_\ell$ and feasibility of t it is $t_s = a_n \in \mathcal{C}$. Otherwise it follows from Lemma 12 that t^{s-} or t^{s+} violates a constraint satisfied in t :

- if t^{s+} violates a constraint satisfied in t , then according to Lemma 11, it must be a time window constraint, and $t_n = b_n$, hence $t_s \in \mathcal{C}$;
- if t^{s-} violates a constraint satisfied in t , then it must be a ride time constraint, and $t_s = t_n = a_d - r_{p,d}$, $(p, d) \in \mathcal{P}$, hence $t_s \in \mathcal{C}$.

Finally, assume that $t_s > t_n$. Then the stop s must be a delivery stop, *i.e.* $s \in S_{>n}$ and the deliveries being scheduled using ‘as soon as possible’ strategy according to (a) $t_s = a_s$.

7.8 Proof of Lemma 7

Proof (Lemma 7). We consider that t schedules the deliveries as soon as possible, *i.e.* $t_d = \max\{k, a_d\}$, for all $d \in S_{>n}$. We have $c(t) = \sum_{s \in S} \sigma_s^L(t_s) + \sum_{(p,d) \in \mathcal{P}} \sigma_{p,d}^{RT}(t_d - t_p)$. In our situation, we can write $c(t) = \sum_{s \in S} \sigma_s^L(t_s) + \sum_{(p,d) \in \mathcal{P}} \sigma_{(p,d)}^{RT}(\max\{k, a_d\} - t_p)$. Thus, by rewriting the sum, we obtain $c(t) = \sum_{(p,d) \in \mathcal{P}} [\sigma_p^L(t_p) + \sigma_{p,d}^{RT}(\max\{k, a_d\} - t_p)] + \sum_{d \in S_{>n}} \sigma_d^L(\max\{k, a_d\})$. Hence, we need to prove $C[n, m] = \sum_{(p,d) \in \mathcal{P}} [\sigma_p^L(t_p) + \sigma_{p,d}^{RT}(\max\{k, a_d\} - t_p)] = \sum_{p \in S_{\leq n}} \sigma_p^k(t_p)$.

It is easy to notice that $C[i, j] = 0 + \sigma_1^k(T_1^k(j_1)) + \dots + \sigma_i^k(T_i^k(j)) = \sum_{l=1}^i \sigma_l^k(T_l^k(j_l))$ with $0 \leq j_1 \leq j_2 \leq \dots \leq j$. Moreover, we can assume that $T_l^k(j_l) \geq a_l$ for $l \leq i$, otherwise $C[i, j_l] = \infty$, which cannot be a minimum. So the equation only considers times which respect the feasibility of a corresponding schedule. Hence, $C[n, k] = c(t) - \sum_{d \in S_{>n}} \sigma_d^L(\max\{k, a_d\})$.

7.9 FPTD algorithm

Algorithm 1: Dynamic programming algorithm for *FPTD* instances.

Input: I : an *FPTD* instance.

Output: an optimal schedule of I .

- 1 $best := \infty$;
 - 2 $C^* :=$ two-dimensional array;
 - 3 **for** $k \in \mathcal{C}$ **do** // $\mathcal{O}(n^2)$
 - 4 Compute $C[n, k]$ with Eq. (2) and save the values in C ; // $\mathcal{O}(n^2)$
 - 5 $score := C[n, k] + \sum_{d \in S_{>n}} \sigma_d^L(\max\{k, a_d\})$;
 - 6 **if** $score < best$ **then**
 - 7 $C^* \leftarrow C$;
 - 8 $best \leftarrow score$;
 - 9 Backtrack C^* to get the schedule t and set $t_d := \max\{t_n, a_d\}$, $\forall d \in S_{>n}$;
// $\mathcal{O}(n)$
 - 10 Return t ;
-

Proof (Theorem 3). According to Lemma 7, Eq. (2) calculates the cost of an optimal schedule. The proof shows that each time used by Eq. (2) to compute $C[n, k]$ can be used to create a feasible schedule. Then, by backtracking the computed table C^* , one can obtain an optimal schedule of I in linear time. Compute $C[n, k]$ for a fixed k takes $\mathcal{O}(nm)$ for $m = \max_{s \in \mathcal{S}} |\mathcal{B}^s(k)|$. Since $|\mathcal{B}^s(k)| = \mathcal{O}(n)$ for all $s \in \mathcal{S}$, independently of k , one can calculate the optimal schedule for a fixed $t_n = k$ in $\mathcal{O}(n^2)$. This step has to be done for each k in \mathcal{C} , thus $\mathcal{O}(n^2)$ times. Hence the $\mathcal{O}(n^4)$ time complexity.