



Discrete optimization

A tight formulation for the dial-a-ride problem

Daniela Gaul^a, Kathrin Klamroth^a, Christian Pfeiffer^b, Michael Stiglmayr^a, Arne Schulz^{b,c,*}^a School of Mathematics and Natural Sciences, University of Wuppertal, Gaußstraße 20, Wuppertal 42119, Germany^b Institute of Operations Management, University of Hamburg, Moorweidenstraße 18, 20148 Hamburg, Germany^c Institute of Quantitative Logistics, Helmut Schmidt University, Holstenhofweg 85, 22043 Hamburg, Germany

ARTICLE INFO

Keywords:

Routing
Transportation
Dial-a-ride problem
Mixed-integer linear programming
Valid inequalities

ABSTRACT

Ridepooling services play an increasingly important role in modern transportation systems. With soaring demand and growing fleet sizes, the underlying route planning problems become increasingly challenging. In this context, we consider the dial-a-ride problem (DARP): Given a set of transportation requests with pick-up and delivery locations, passenger numbers, time windows, and maximum ride times, an optimal routing for a fleet of vehicles, including an optimized passenger assignment, needs to be determined. We present tight mixed-integer linear programming (MILP) formulations for the DARP by combining two state-of-the-art models into novel *location-augmented-event-based* formulations. Strong valid inequalities and lower and upper bounding techniques are derived to further improve the formulations. We then demonstrate the theoretical and computational superiority of the new models: First, the linear programming relaxations of the new formulations are stronger than existing location-based approaches. Second, extensive numerical experiments on benchmark instances show that computational times are on average reduced by 53.9% compared to state-of-the-art event-based approaches.

1. Introduction

The dial-a-ride problem (DARP) is a well-studied routing problem (e.g., Ho et al., 2018). It has topical applications in modern transportation systems. While ridepooling services are probably the most important area of application (e.g., Foljanty, 2020; Pfeiffer & Schulz, 2022), DARPs are also relevant for the transportation of patients (e.g., Molenbruch, Braekers, Caris, & Berghe, 2017) or, in a related branch of research, for the transportation of goods. Given a fixed number of vehicles, the goal is to identify optimal vehicle routes together with an optimized passenger assignment so that a set of customer requests is served in a best possible way. A customer request typically consists of a number of passengers that want to be transported from a pick-up to a delivery location within specified time windows and within a maximum acceptable ride time. While several requests may share the same vehicle for parts of their trips, the capacity of a vehicle may not be exceeded at any point in time.

This paper is motivated by the ridepooling service HolMichApp (<https://www.holmich-app.de>) which was launched in 2019 in the mid-sized city of Wuppertal in Germany with the purpose of improving mobility and reducing congestion and the use of private cars. Ridepooling services play an important role in the transformation to a sustainable future by improving mobility and reducing congestion

and emissions. To make ridepooling services competitive and efficient, optimized routing and customer assignments are of central importance. Hence, the exact and fast solution of realistically-sized DARP instances in a rolling horizon environment is pivotal for the success of ridepooling services in particular, and for a sustainable improvement of public transportation systems in general.

In this paper, we present new mixed-integer linear programming (MILP) formulations for the static DARP that build upon state-of-the-art event-based (EB) and location-based (LB) models, using the best of both worlds. Our results transfer to the dynamic DARP in two ways: First, Gaul et al. (2021) showed that EB formulations can be applied efficiently in a rolling-horizon framework for the dynamic DARP. This also holds for the new models developed in this paper. Second, we show in this paper that the new models are tighter than well-known LB models (see Theorem 2 in Section 3).

1.1. Related work

Since this paper considers mixed-integer linear programming formulations for the DARP, the following short literature review is focused on exact methods for DARP and for the closely related pick-up and delivery

* Corresponding author at: Institute of Quantitative Logistics, Helmut Schmidt University, Holstenhofweg 85, 22043 Hamburg, Germany.

E-mail addresses: gaul@math.uni-wuppertal.de (D. Gaul), klamroth@math.uni-wuppertal.de (K. Klamroth), christian.pfeiffer@uni-hamburg.de (C. Pfeiffer), stiglmayr@math.uni-wuppertal.de (M. Stiglmayr), arne.schulz@uni-hamburg.de, arne.schulz@hsu-hh.de (A. Schulz).

problem with time windows (PDPTW). For a broader introduction we refer to the surveys of Molenbruch, Braekers, and Caris (2017) and Ho et al. (2018).

Most exact methods are based on branch-and-cut or branch-and-price frameworks that are tailored for different MILP formulations. Cordeau (2006) presented a 3-index MILP formulation for the DARP where binary variables indicate whether a certain vehicle k drives directly from location i to location j . One year later, Ropke et al. (2007) reformulated the model by Cordeau (2006) by omitting the index for the vehicles. Now, a binary variable indicates whether any vehicle drives directly from location i to location j . Both models are typical examples of *location-based (LB) formulations* in which binary variables indicate whether a vehicle drives directly between two locations. Cordeau (2006) and Ropke et al. (2007) suggested branch-and-cut frameworks for the LB models, and Ropke et al. (2007) derived additional valid inequalities. Particularly the fork constraints and the reachability constraints that were adapted from Lysgaard (2006) proved to be very effective. LB models are discussed in more detail in Section 2.1.

In branch-and-price methods, the DARP is usually subdivided into two subproblems: The master problem where tours are selected, and the pricing problem where the tours are generated. Ropke and Cordeau (2009) used a branch-and-cut-and-price approach for the PDPTW. They compared two formulations for the pricing problem, an elementary and a non-elementary shortest path problem. The branch-and-cut-and-price algorithm suggested in Gschwind and Irnich (2015) instead uses a pricing problem that includes ride time constraints explicitly.

Comparing branch-and-cut and branch-and-price, we observe that the LB formulation has a quadratic number of binary variables (in the number of locations). In the worst case, all feasible tours need to be constructed in the branch-and-price approach. However, the LB formulation requires a potentially exponential number of constraints to ensure that corresponding pick-up and delivery nodes are visited in the same tour. This is not the case in the pricing problem, as it can be formulated as a resource constrained shortest path problem. In conclusion, different formulations have a trade-off between the number of binary variables and the number of required constraints.

In recent years, new approaches found different compromises with respect to this trade-off. Rist and Forbes (2021) introduced route fragments which are segments of a vehicle tour such that the vehicle is empty at the beginning and at the end of a fragment. Thus, vehicle tours can be compounded by fragments. Rist and Forbes (2021) used the fact that time windows typically strongly reduce the number of possible fragments. In their approach, binary decision variables indicate whether a fragment is used in the solution or not and whether fragments are scheduled in direct succession within a tour. They used a branch-and-cut framework to solve their model. Gaul et al. (2022) introduced an *event-based* (EB) formulation where an event represents the current occupancy of the vehicle. In this formulation, binary variables indicate whether two events occur in direct succession in a vehicle's tour. Gaul et al. (2022) used the fact that the vehicle capacity is typically rather small in practice, as the number of events strongly depends on the vehicle capacity. We review their approach in detail in Section 2.2. Both aforementioned approaches by Rist and Forbes (2021) and Gaul et al. (2022) have the advantage that the vehicle capacity constraints are enforced by the choice of fragments and events, respectively. Moreover, precedence constraints are either ensured by the choice of fragments or can be implemented implicitly by the selection of arcs connecting events. This considerably reduces the number of constraints in EB models as compared to LB formulations. A survey on formulations for the related vehicle routing problem and a systematic study of how these formulations relate to each other is given in Letchford and Salazar-González (2005).

1.2. Our contribution

The above literature review implies a classification of MILP formulations with regard to the information contained in the binary variables, see Fig. 1 for an illustration. In this paper, we propose novel MILP formulations that combine the advantages from the LB formulation of Ropke et al. (2007) and the EB model of Gaul et al. (2022). The new models are further improved by adapting problem specific valid inequalities inspired by Schulz and Pfeiffer (2024).

Our main contributions are:

- We introduce a *location-augmented-event-based (LAEB) formulation* and an *aggregated location-augmented-event-based (ALAEB) formulation*. The ALAEB is a slightly adapted variant of the LAEB that has significantly fewer binary variables and thus potentially reduces the size of the branch-and-bound tree.
- We prove that both new formulations have a tighter LP relaxation than previous LB models (see Theorem 2 in Section 3). This tightness is also reflected by an average root node gap of only 1.6% in our computational study.
- Preprocessing the event-based graph reduces the size of both new MILP formulations by 22% on average.
- New valid inequalities are derived for the EB, LAEB, and ALAEB formulations.
- In a comprehensive computational study, the LAEB formulation leads to a 53.9% reduction of computational time on benchmark instances as compared to state-of-the-art EB formulations.
- Our formulation is hence suitable for a rolling-horizon framework and for practical applications of a ridepooling service like HolMichApp.

The paper is structured as follows: First, we recapitulate the LB formulation by Ropke et al. (2007) and the EB formulation by Gaul et al. (2022) in detail and present our new formulations in Section 2. Our theoretical investigation of the new formulations is presented in Section 3. In Section 4, we introduce methods to improve the performance of branch-and-bound search. The new formulations as well as the introduced methods are evaluated in the computational study in Section 5. Finally, the paper closes with a conclusion in Section 6.

2. MILP formulations

We consider a DARP with n customers. Each customer request $i \in R = \{1, \dots, n\}$ contains a number of persons $q_i \in \mathbb{N} \setminus \{0\}$ and a service time $s_i \geq 0$, which can be interpreted as the time the customer needs to enter or exit the vehicle. They have to be transported by one of K identical vehicles with capacity Q between their pick-up location i^+ and delivery location i^- . The set of pick-up locations is denoted by $P = \{1^+, \dots, n^+\}$ and the set of delivery locations by $D = \{1^-, \dots, n^-\}$. The vehicle depot is represented by 0. To simplify the notation, we set $s_{i^+} = s_{i^-} = s_i$, $s_0 = 0$, $q_{i^+} = q_{i^-} = q_i$ as well as $q_0 = 0$. A customer is not allowed to be in the vehicle for a longer time than a specified maximum ride time L_i . For the travel time \bar{t}_{ij} and corresponding routing costs \bar{c}_{ij} between each pair of locations $i, j \in J = P \cup D \cup \{0\}$ we assume that \bar{c}_{ij} and \bar{t}_{ij} are non-negative and fulfill the triangle inequality. Let \bar{t}_i denote the travel time for a direct ride from pick-up to delivery location of request i , i.e., $\bar{t}_i = \bar{t}_{i^+i^-}$. W.l.o.g. we assume $0 < \bar{t}_i + s_i < L_i$ for all $i \in R$. All locations $i \in J$ have to be visited within a given time window $[e_i, \ell_i]$. The beginning of service is given by the lower bound $e_0 = 0$ of the time window at the depot, and there is a fixed duration of service T , so that the end of service is given by $\ell_0 := e_0 + T$. We assume that $e_{i^+} \geq \bar{t}_{i^+i^-}$, i.e., each pick-up location can be reached at the beginning of the time window when starting at time 0 in the depot. The objective is to minimize the total routing costs. An overview of the used notation can be found in Appendix A. Note, that we use a bar over parameters and variables if they are indexed by locations and the same type of parameter and variable (without bar), respectively, is used for events.

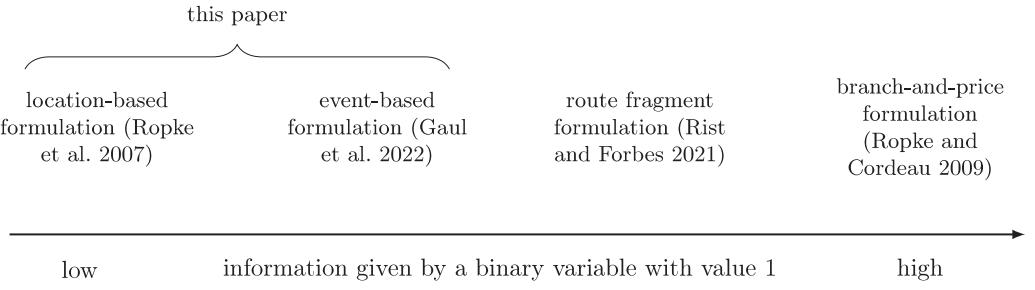


Fig. 1. Classification of formulations according to the information given by a binary variable with value 1.

The DARP can be represented by different graph-based formulations. We first present the LB formulation by Ropke et al. (2007) in the following section before we present the EB formulation by Gaul et al. (2022). Afterwards, we combine both formulations into two location-augmented-event-based formulations in Sections 2.3 and 2.4, respectively, by using the advantages of both the event- and the location-based formulation.

2.1. Location-based MILP formulation

In this section, we present the LB formulation by Ropke et al. (2007) which will be used as reference LB model in the following (the first one in their paper; additionally the number of vehicles is restricted). In this model, the vehicle depot is denoted by 0^+ and 0^- , raising the possibility of a different start and end depot, however, in this paper, 0^+ and 0^- represent the same vehicle depot 0 and are only used to stick to the notation from the original paper by Ropke et al. (2007). Thus, we set $s_{0^+} = s_{0^-} = s_0$, $q_{0^+} = q_{0^-} = q_0$, $e_{0^+} = e_{0^-} = e_0$, $\ell_{0^+} = \ell_{0^-} = \ell_0$ as well as $\bar{c}_{0+j} = \bar{c}_{0-j} = \bar{c}_0$, $\bar{c}_{i0^+} = \bar{c}_{i0^-} = \bar{c}_i$, $\bar{t}_{0+j} = \bar{t}_{0-j} = \bar{t}_0$, and $\bar{t}_{i0^+} = \bar{t}_{i0^-} = \bar{t}_i$. In the LB formulation, each node in the underlying graph is associated with a location $j \in \bar{J} = P \cup D \cup \{0^+, 0^-\}$. The graph is complete and directed, i.e., traveling the arc (i, j) means that the vehicle drives from location i to j , where $i, j \in \bar{J}$.

A binary variable \bar{x}_{ij} is associated with each arc and is 1 if a vehicle drives directly from location i to location j , i.e., the arc (i, j) is used, and 0 otherwise. Moreover, Q_j represents the number of customers in the vehicle after leaving location j and \bar{B}_j is the time when service at location j begins. In the model, S is the set of all sets $S \subseteq \bar{J}$ with $0^+ \in S$, $0^- \notin S$, and there is at least one request i for which the delivery node is in S but not the pick-up node. This means $S = \{S : 0^+ \in S \wedge 0^- \notin S \wedge \exists i : (i^+ \notin S \wedge i^- \in S)\}$. Moreover, \bar{M}_{ij} with $i, j \in \bar{J}$ is a sufficiently large constant. Then, the model is formulated as:

$$Z_{LB} = \min \sum_{i,j \in \bar{J}} \bar{c}_{ij} \cdot \bar{x}_{ij} \quad (1a)$$

$$\text{s.t. } \sum_{i \in \bar{J}} \bar{x}_{ij} = 1 \quad \forall j \in P \cup D \quad (1b)$$

$$\sum_{j \in \bar{J}} \bar{x}_{ij} = 1 \quad \forall i \in P \cup D \quad (1c)$$

$$\sum_{j \in P} \bar{x}_{0^+j} \leq K \quad (1d)$$

$$\sum_{i \in S} \bar{x}_{ij} \leq |S| - 2 \quad \forall S \in S \quad (1e)$$

$$\bar{B}_i + s_i + \bar{t}_{ij} - \bar{M}_{ij} (1 - \bar{x}_{ij}) \leq \bar{B}_j \quad \forall i, j \in \bar{J} \quad (1f)$$

$$Q_j + \mathcal{I}_{\{j \in P\}} q_j - Q (1 - \bar{x}_{ij}) \leq Q_j \quad \forall i, j \in \bar{J} \quad (1g)$$

$$e_j \leq \bar{B}_j \leq \ell_j \quad \forall j \in \bar{J} \quad (1h)$$

$$\max\{0, \mathcal{I}_{\{j \in P\}} q_j\} \leq Q_j \leq \min\{Q, Q + \mathcal{I}_{\{j \in P\}} q_j\} \quad \forall j \in \bar{J} \quad (1i)$$

$$\bar{B}_{i^-} - \bar{B}_{i^+} - s_i \leq L_i \quad \forall i \in P \quad (1j)$$

$$\bar{x}_{ij} \in \{0, 1\} \quad \forall i, j \in \bar{J} \quad (1k)$$

Objective function (1a) minimizes the total routing costs. Constraints (1b) and (1c) ensure that each customer location is visited and left exactly once. Due to constraint (1d) the depot is left at most K times, i.e., up to K vehicles are used. Because of constraints (1e) we are sure that the pick-up and the delivery location of each customer are visited in the same tour and in the correct order. Constraints (1f) and (1g) take care that each location is visited within its time window. Constraints (1h) model the vehicle load which is not allowed to exceed the vehicle's capacity because of constraints (1i), where

$$\mathcal{I}_{\{j \in P\}} = \begin{cases} 1 & j \in P, \\ -1 & \text{otherwise.} \end{cases}$$

No customer is allowed to be longer in the vehicle than their maximum ride time L_i , which is ensured by constraints (1j). Finally, Constraints (1k) are the binary constraints for the sequence variables.

As can be seen the size of the set S grows exponentially in the number of customers n . Moreover, constraints (1f) and (1g) are numerically unfavorable big- M -constraints (the Q in (1g) is also a large constant). The larger M_{ij} is, the closer can a fractional value of \bar{x}_{ij} be to 1 without being binding in constraints (1f) (Camm et al., 1990; Codato & Fischetti, 2006).

2.2. Event-based MILP formulation

The EB formulation introduced by Gaul et al. (2022) overcomes two of the aforementioned drawbacks of LB formulations, as it does not contain an exponentially growing number of constraints as constraints (1e), since it relies on a graph structure which enforces pairing and precedence constraints implicitly. Also vehicle capacity constraints (1g) are incorporated implicitly. In the event-based graph, nodes represent states or events, which are not only associated with the current location of the vehicle but also with the entire state of the vehicle, including the information on all customers sitting in the vehicle.

As at most Q customer requests are transported at the same time by a vehicle (if $q_i = 1$ for all of them), the allocation of customers to a vehicle (and hence the nodes of the event-based graph) can be represented by a Q -tuple. If fewer than Q requests are allocated to the vehicle, the remaining places are marked by 0. This holds also if a single request already contains Q persons. Thus, a 0 component in Q does not mean that the vehicle has empty seats. Since all customers present in the vehicle are part of the event description, we can easily ensure that capacity constraints (1g) are fulfilled by deleting all nodes where the sum of the q_i values of all customers present in the vehicle exceeds

Q . Thus, we can never reach a situation where a capacity constraint is violated.

Reordering the customers in the vehicle leads to different representations of the same state. To avoid ambiguities, we assume that the customers are noted in decreasing order of their index and the event is filled up with 0s on the remaining places. An exception is the first position of the Q -tuple, which indicates the current position of the vehicle, i.e., the last location visited. By this, there is a direct mapping of events to locations. On the other hand, each location may be associated with different events with different customers present in the vehicle at the time the location is reached.

As an example, event $(3^+, 2, 1)$ represents for a vehicle capacity of $Q = 3$ the situation that the vehicle is at the pick-up location of customer 3 (3^+) and customers 1 and 2 are sitting in the vehicle. Another example is the event $(1^-, 2, 0)$. Here the vehicle delivered customer 1 and customer 2 is still sitting in the vehicle.

Cordeau (2006) pointed out that time window and ride time constraints can be used to identify incompatible pairs of users. Given users $i, j \in R$, the following reductions are possible (cf. Gaul et al., 2022):

- if it is infeasible to visit locations i^+, i^-, j^+, j^- both in the order $j^+ \rightarrow i^+ \rightarrow j^- \rightarrow i^-$ and $j^+ \rightarrow i^+ \rightarrow i^- \rightarrow j^-$, the set $\{(v_1, v_2, \dots, v_Q) \in V : v_1 = i^+, v_k = j \text{ for any } k \in \{2, \dots, Q\}\}$ can be removed from the node set.
- if it is infeasible to visit locations i^+, i^-, j^+, j^- both in the order $j^+ \rightarrow i^+ \rightarrow i^- \rightarrow j^-$ and $i^+ \rightarrow j^+ \rightarrow i^- \rightarrow j^-$, the set $\{(v_1, v_2, \dots, v_Q) \in V : v_1 = i^-, v_k = j \text{ for any } k \in \{2, \dots, Q\}\}$ can be removed from the node set.

Now, let f_{ij}^1 and $f_{ij}^2 \in \{0, 1\}$ encode the feasibility of the paths $j^+ \rightarrow i^+ \rightarrow j^- \rightarrow i^-$ and $j^+ \rightarrow i^+ \rightarrow i^- \rightarrow j^-$, respectively, w.r.t. ride time and time window constraints. To simplify the notation, let $f_{i0}^1 = f_{0i}^1 = f_{i0}^2 = f_{0i}^2 = 1$. By this, the number of Q -tuples representing feasible user allocations can be reduced significantly by computing f_{ij}^1, f_{ij}^2 for all pairs of users $i, j \in R$. Then, the set of pick-up events of user $i \in R$ can be expressed as

$$V_{i^+} := \left\{ (v_1, v_2, \dots, v_Q) : v_1 = i^+, v_l \in R \cup \{0\} \setminus \{i\}, \right. \\ \left. f_{i,v_l}^1 + f_{i,v_l}^2 \geq 1 \forall l \in \{2, \dots, Q\}, \right. \\ \left. (v_l > v_{l+1} \vee v_{l+1} = 0) \forall l \in \{2, \dots, Q-1\}, \right. \\ \left. \sum_{l=1}^Q q_{v_l} \leq Q \right\}.$$

Analogously, the set of delivery nodes corresponding to $i \in R$ can be expressed as

$$V_{i^-} := \left\{ (v_1, v_2, \dots, v_Q) : v_1 = i^-, v_l \in R \cup \{0\} \setminus \{i\}, \right. \\ \left. f_{v_l,i}^1 + f_{v_l,i}^2 \geq 1 \forall l \in \{2, \dots, Q\}, \right. \\ \left. (v_l > v_{l+1} \vee v_{l+1} = 0) \forall l \in \{2, \dots, Q-1\}, \right. \\ \left. \sum_{l=1}^Q q_{v_l} \leq Q \right\}.$$

The depot is represented by the single event $(0, \dots, 0)$. We set $V_0 = \{(0, \dots, 0)\}$. In total, the nodes of the event-based graph are given by

$$V = V_0 \cup \bigcup_{i=1}^n V_{i^+} \cup \bigcup_{i=1}^n V_{i^-}.$$

The arcs correspond to feasible transitions between events. In the aforementioned example, there is no arc between events $(3^+, 2, 1)$ and $(1^-, 2, 0)$, as customer 3 is present in the vehicle in the first event but not in the second. However,

$$(3^+, 2, 1) \rightarrow (3^-, 2, 1) \rightarrow (1^-, 2, 0)$$

may be feasible transitions. In the left event, customer 3 enters the vehicle with customers 1 and 2 sitting in it at customer 3's pick-up location. In the second, the vehicle is at customers 3's delivery location, i.e., customer 3 leaves the vehicle while customers 1 and 2 are still in it. Then, the vehicle drives to customer 1's delivery location while customer 2 is still sitting in the vehicle. In comparison to the LB approach, arcs still correspond to a transition between two locations. However, while an arc in the location-based graph represents only that the vehicle drives from one location to another, the information on the other customers sitting in the vehicle is also encoded in the event-based representation. In other words, the LB arc between locations i and j aggregates all event-based arcs between nodes v and w representing these two locations, i.e., with $v_1 = i$ and $w_1 = j$ where v_1 and w_1 represent the first entries of v and w , respectively.

In line with this connection between both approaches, the set of arcs is

$$A = \bigcup_{m=1}^6 A_m,$$

where the subsets A_i are defined as follows:

- A_1 defines the transition between a customer's pick-up location and a customer's delivery location, i.e.

$$A_1 = \{((i^+, v_2, \dots, v_Q), (j^-, w_2, \dots, w_Q)) \in V \times V : \{i, v_2, \dots, v_Q\} = \{j, w_2, \dots, w_Q\}\}.$$

- A_2 represents transitions between a customer's pick-up location and another customer's pick-up location, i.e.

$$A_2 = \{((i^+, v_2, \dots, v_{Q-1}, 0), (j^+, w_2, \dots, w_Q)) \in V \times V : \{i, v_2, \dots, v_{Q-1}\} = \{w_2, \dots, w_Q\}\}.$$

- A_3 is the set of all transitions between a customer's delivery location and another customer's pick-up location, i.e.

$$A_3 = \{((i^-, v_2, \dots, v_Q), (j^+, v_2, \dots, v_Q)) \in V \times V : i \neq j\}.$$

- A_4 defines the transition between the delivery locations of two customers, i.e.

$$A_4 = \{((i^-, v_2, \dots, v_Q), (j^-, w_2, \dots, w_{Q-1}, 0)) \in V \times V : \{v_2, \dots, v_Q\} = \{j, w_2, \dots, w_{Q-1}\}\}.$$

- A_5 is the set of all transitions between a delivery location and the depot, i.e.

$$A_5 = \{((i^-, 0, \dots, 0), (0, \dots, 0)) \in V \times V\}.$$

- A_6 , finally, describes the transition between the depot and a pick-up location, i.e.

$$A_6 = \{((0, \dots, 0), (i^+, 0, \dots, 0)) \in V \times V\}.$$

In the following, we formulate the EB model by Gaul et al. (2022). To simplify the notation, let $a = (v, w) \in A$ be the arc between the nodes representing the events v and w . Then, routing costs and travel times between locations v_1 and w_1 can be represented as c_a and t_a , respectively. Moreover, let $\delta^{in}(v) = \{(w, v) \in A\}$ and $\delta^{out}(v) = \{(v, w) \in A\}$ be the sets of incoming and outgoing arcs for node v . Analogous to the LB model, variable x_a is 1 if arc a is used in the solution and B_v reports the beginning of service at location v_1 .

Gaul et al. (2022) present two model formulations. The following one is the second which led to significantly shorter computation times in their computational evaluation. In some applications of the DARP, we can distinguish between inbound requests, i.e. requests that submit a time for pick-up, and outbound requests, i.e. requests that submit a time for delivery, see Cordeau (2006). In the EB formulation, Gaul et al. (2022) assume that all pick-up time windows of inbound requests and

all delivery time-windows of outbound requests have a fixed length of TW .

$$Z_{EB} = \min \sum_{a \in A} c_a \cdot x_a \quad (2a)$$

$$\text{s.t. } \sum_{a \in \delta^{in}(v)} x_a - \sum_{a \in \delta^{out}(v)} x_a = 0 \quad \forall v \in V \quad (2b)$$

$$\sum_{a \in \delta^{in}(v), v \in V_i^+} x_a = 1 \quad \forall i \in R \quad (2c)$$

$$\sum_{a \in \delta^{out}(0, \dots, 0)} x_a \leq K \quad (2d)$$

$$B_w \geq B_v + s_{v_1} + t_{(v,w)} - M_{vw}(1 - x_{(v,w)}) \quad \forall (v, w) \in A, v \neq (0, \dots, 0) \quad (2e)$$

$$B_w \geq e_0 + t_{(v,w)}x_{(v,w)} \quad \forall ((0, \dots, 0), w) \in A \quad (2f)$$

$$e_0 \leq B_{(0, \dots, 0)} \leq \ell_0 \quad (2g)$$

$$e_{i^+} + TW \left(1 - \sum_{a \in \delta^{in}(v)} x_a \right) \leq B_v \leq \ell_{i^+} \quad \forall i \in R, v \in V_{i^+} \quad (2h)$$

$$e_{i^-} \leq B_v \leq e_{i^+} + L_i + s_{i^+} + TW \sum_{a \in \delta^{in}(v)} x_a \quad \forall i \in R, v \in V_{i^-} \quad (2i)$$

$$B_w - B_v - s_{i^+} \leq L_i \quad \forall i \in R, v \in V_{i^+}, w \in V_{i^-} \quad (2j)$$

$$x_a \in \{0, 1\} \quad \forall a \in A \quad (2k)$$

The objective function corresponds to that of the LB formulation; in both cases, the total travel costs are minimized (2a). Constraints (2b) are the flow conservation constraints. They ensure that each visited event is left again. Constraints (2c) take care that all pick-up locations are visited by a vehicle. Constraints (2b)–(2c) correspond to constraints (1b)–(1c) in the LB formulation. Due to constraint (2d) at most K vehicles leave the depot (corresponds to constraint (1d)). Thus, at most K tours are constructed. Constraints (2e) focus on the time consistency. If a vehicle drives from location v_1 to location w_1 using the events v and w , then the service at w_1 cannot start before the service at v_1 is finished ($B_v + s_{v_1}$) and the vehicle has driven to location w_1 ($t_{(v,w)}$; corresponds to constraints (1f)). Time consistency is considered separately for the first customer in a vehicle route (constraints (2f)). Constraints (2g)–(2i) focus on the time windows. While constraint (2g) is the special case for the depot, events are separated according to their current location into constraints (2h) (pick-up locations) and (2i) (delivery locations). This is due to the fact that each location is represented by different events in the EB model. However, only one of the events corresponding to a location is used in a solution. See Gaul et al. (2022) for a detailed derivation of the time window constraints. Maximum ride time constraints are implemented analogously to the LB formulation, cf. constraints (1j) and (2j). Note that $B_v \geq 0$ for all $v \in V$ because of constraints (2g)–(2i), as $e_{v_1} \geq 0$ for all $v \in V$.

Note further that the model assumes all vehicles to be homogeneous with a capacity of Q . If we have a heterogeneous fleet with vehicles of different size, we can copy the event-based graph for each vehicle type (each with a different capacity) connected by a common depot. The flow from the depot into the different parts of the graph is then restricted by the number of vehicles of that type (compare constraint (2d)). In contrast, it is much easier to consider heterogeneous vehicles in the LB formulation, as we need only to consider variables \bar{x}_{ij}^k instead of variables \bar{x}_{ij} , and adapt the constraints of type (1g) such that they depend on the vehicle type (and thereby add some more of them).

Comparing the EB formulation with the LB formulation we find advantages for both of them. Gaul et al. (2022) show that the number of nodes and arcs of the event-based graph can be bounded for $n \geq Q+1$ by $\mathcal{O}(n^Q)$ and $\mathcal{O}(n^{Q+1})$, respectively. In comparison, the LB approach requires one node per location, i.e., $\mathcal{O}(n)$, and one arc for each pair

of locations, i.e., $\mathcal{O}(n^2)$. Thus, the advantage of the LB formulation is the smaller size of the underlying graph and thereby of the number of variables. However, in practical applications Q is usually rather small (typically $Q = 3$ or $Q = 6$), and we can reduce the size of the event-based graph due to time windows, travel and ride times (which is presented in Section 4). On the other hand, the event-based graph has the already mentioned advantage that capacity, pairing, and precedence constraints are modeled implicitly in the graph.

We reformulate the EB model in the next sections to integrate the best of both models and show the advantages of the resulting formulations over the LB formulation theoretically afterwards.

2.3. Location-augmented-event-based MILP formulation

In this section, we integrate the implementation of time consistency in the LB formulation into the EB formulation to obtain a more efficient formulation. Note that this is also a generalization of the EB formulation because time window lengths need not longer be identical.

$$Z_{LAE} = \min \sum_{a \in A} c_a \cdot x_a \quad (3a)$$

s.t. (1h), (1j), (2b)–(2d), and (2k)

$$\bar{B}_j \geq \bar{B}_i + s_i + \bar{t}_{ij} - \bar{M}_{ij} \left(1 - \sum_{v: v_1=i, w: w_1=j} x_{(v,w)} \right) \quad \forall i, j \in J \quad (3b)$$

Due to the construction of the event-based graph and constraints (2c) and (2k) there is exactly one node in the graph representing an event w with $w_1 = i$ and $\sum_{a \in \delta^{in}(w)} x_a = 1$ while $\sum_{a \in \delta^{in}(w')} x_a = 0$ for all other nodes w' with $w'_1 = i$. Therefore, only for the first one $M_{vw}(1 - x_{(v,w)})$ is 0 in constraints (2e). This constraint sets $B_w = \bar{B}_j$ while the others only ensure that feasible values are used for $B_{w'}$. This leads to the simplification of time consistency constraints (3b). By this, time windows constraints (1h) and maximum ride time constraints (1j) can be ensured as in the LB formulation. As can be seen, the number of constraints in (1h), (1j), and (3b) is clearly smaller than in (2e)–(2j). Moreover, the numerically unfavorable big- M -constraints are stronger in (3b) than in (2e), as

$$\left(1 - \sum_{v: v_1=v'_1, w: w_1=w'_1} x_{(v,w)} \right) \leq (1 - x_{(v',w')}) \quad \forall (v', w') \in A$$

while $\bar{M}_{ij} = M_{vw}$ with $v_1 = i$ and $w_1 = j$ with the classical choice

$$\bar{M}_{ij} = \ell_i + s_i + \bar{t}_{ij} - e_j \quad \forall i, j \in J$$

and

$$M_{vw} = \ell_{v_1} + s_{v_1} + t_{(v,w)} - e_{w_1} \quad \forall (v, w) \in A.$$

Thus, the model formulation (1h), (1j), (2b)–(2d), (2k), and (3a)–(3b) is an improved version of (2a)–(2k). In the next section, we present another location-augmented-event-based formulation with fewer binary variables.

2.4. Aggregated location-augmented-event-based MILP formulation

The LB formulation has the advantage that only $\mathcal{O}(n^2)$ binary variables are required while the EB formulation has $\mathcal{O}(n^{Q+1})$ binary variables (for $n \geq Q+1$). We use the equality

$$\bar{x}_{ij} = \sum_{(v,w) \in A: v_1=i \wedge w_1=j} x_{(v,w)}, \quad (4)$$

which is already used in (3b). Thereby, we get the following ALAEB formulation:

$$Z_{ALAEB} = \min \sum_{i,j \in J} \bar{c}_{ij} \cdot \bar{x}_{ij} \quad (5a)$$

s.t. (1h), (1j), (1k), (2b)–(2d), and (4)

$$\begin{aligned} \bar{B}_j &\geq \bar{B}_i + s_i + \bar{t}_{ij} - \bar{M}_{ij} (1 - \bar{x}_{ij}) & \forall i, j \in P \cup D \\ 0 \leq x_a &\leq 1 & \forall a \in A \end{aligned} \quad (5b) \quad (5c)$$

While (5a) corresponds to (1a), constraints (5b) are equivalent to (3b) using (4). Thus, the difference between both location-augmented-event-based models is the fact that \bar{x} variables are added as binary variables while x variables are relaxed in constraints (5c). For this reason, the model contains fewer binary variables than the LAEB formulation. Note, that given an integer solution \bar{x} the tours are completely described. Thus, there are at most K flows of size one from depot to depot through the event-based graph, i.e., x_a variables are also integer. In comparison to the LB formulation, the ALAEB formulation replaces an exponential number of constraints (1e) by an additional number of $\mathcal{O}(n^{Q+1})$ continuous variables and $\mathcal{O}(V)$ constraints (2b)–(2d), and (4).

3. Theoretical analysis

In this section, we investigate the properties of the LP relaxations of the LAEB and the ALAEB formulation in comparison to the LP relaxation of the LB formulation. We introduced the ALAEB formulation mainly to branch on a different set of variables. The following theorem shows, however, that the LP relaxations of the LAEB and the ALAEB formulation are equivalent.

Theorem 1. *The LP relaxations of the LAEB formulation (1h), (1j), (2b)–(2d), (2k), and (3a)–(3b) and of the ALAEB formulation (1h), (1j), (1k), (2b)–(2d), (4), and (5a)–(5c) are equivalent.*

The proof of **Theorem 1** can be found in [Appendix B](#). We formulate and prove the following theorem only for the LAEB formulation, but due to **Theorem 1** the results hold also for the ALAEB formulation.

Theorem 2. *Let Z_{LB}^{rel} be the objective value of an optimal solution of the LP relaxation of the LB model (1a)–(1k), i.e., with $0 \leq \bar{x}_{ij} \leq 1$ for all $i, j \in \bar{J}$ instead of (1k). Let further Z_{LAEB}^{rel} be the objective value of an optimal solution of the LP relaxation of the LAEB model (1h), (1j), (2b)–(2d), (2k), and (3a)–(3b), i.e., with $0 \leq x_a \leq 1$ for all $a \in A$ instead of (2k). Then, $Z_{LAEB}^{rel} \geq Z_{LB}^{rel}$ holds. Moreover, there are instances in which $Z_{LAEB}^{rel} > Z_{LB}^{rel}$ holds.*

The proof of **Theorem 2** can be found in [Appendix C](#). The proof highlights again that the LAEB and the ALAEB formulation implicitly ensure that pairing, precedence, and capacity constraints are fulfilled. The implicit implementation of capacity constraints leads even to a tighter polyhedron of the LP relaxation. Due to their exponential size, pairing and precedence constraints (1e) are typically not added upfront but only if they are violated. Therefore, the LAEB and the ALAEB LP relaxations are also tighter in this aspect. However, we still face a significant number of nodes and arcs in the event-based graph, which we compactify in [Section 4](#).

4. Preprocessing and branch-and-cut methods

As mentioned at the end of [Section 2.2](#) the EB model formulation suffers from a very significant number of variables ($\mathcal{O}(n^{Q+1})$). Therefore, we present methods to reduce the size of the event-based graph in preprocessing and in branch-and-cut nodes as well as new types of additional valid inequalities in this section. [Fig. 2](#) gives an overview of the developed methods.

First, we introduce preprocessing methods to eliminate impossible events and arcs in the event-based graph based on bounds for B_v variables ([Section 4.1](#)). Moreover, we propose new valid inequalities to avoid subtours, incompatible events, and infeasible paths ([Section 4.2](#)). Second, methods are presented to improve the search in branch-and-cut nodes based on previous branching decisions, see [Sections 4.3–4.5](#).

4.1. Elimination of impossible event nodes and arcs in preprocessing

We already explained in [Section 2.2](#) that the number of events can be reduced by considering pairwise incompatibilities with respect to time windows and ride time constraints. We present a method to systematically compute earliest and latest starting times of service at all events. The earliest starting time B_v^{LB} at node v can be interpreted as the shortest path between depot event $(0, \dots, 0)$ and the considered event (v_1, \dots, v_Q) , i.e., the fastest way to pick up first customers v_2, \dots, v_Q (for all $v_j \neq 0$) and v_1 afterwards if $v_1 \in P$ or deliver v_1 afterwards if $v_1 \in D$ while respecting the corresponding time windows. Analogously, the latest starting time of service B_v^{UB} in (v_1, \dots, v_Q) is the latest time for start of service such that customers v_1, \dots, v_Q , $v_j \neq 0$, can all be delivered within the time windows. Naturally, an event cannot be feasible if $B_v^{UB} < B_v^{LB}$. An event v is also infeasible if there is a customer i in the vehicle who cannot be delivered within the maximum ride time, i.e., if $\bar{B}_i^{UB} + s_i + L_i - s_{v_1} - t_{v_1, i^-} < B_v^{LB}$, whereat $\bar{B}_i^{UB} = \max_{v \in V_i^+} \{B_v^{UB}\}$. Moreover, an arc (v, w) is infeasible if $B_v^{LB} + s_{v_1} + t_{(v, w)} > B_w^{UB}$. Then, we can eliminate (v, w) from the graph.

Due to construction of the event-based graph,

$$B_v^{LB} = \max \left\{ e_{v_1}, \min_{w: (w, v) \in \delta^{in}(v) \wedge w \in \bigcup_{i=1}^n V_i^+} \left\{ B_w^{LB} + s_{w_1} + t_{(w, v)} \right\} \right\} \quad (6)$$

and

$$\begin{aligned} B_v^{UB} = \min \left\{ \ell_{v_1} - \sum_{i \in R} \mathbb{1}_{\{v_1=i^-\}} \cdot \max \{0, \ell_{v_1} - (\bar{B}_i^{UB} + s_i + L_i)\}, \right. \\ \left. \max_{w: (v, w) \in \delta^{out}(v) \wedge w \in \bigcup_{i=1}^n V_i^-} \left\{ B_w^{UB} - t_{(v, w)} - s_{v_1} \right\}, \right. \\ \left. \min_{i \in R | \exists l \geq 2: v_l=i} \{ \bar{B}_i^{UB} + s_i + L_i - s_{v_1} - t_{v_1, i^-} \} \right\}. \end{aligned} \quad (7)$$

We can compute B_v^{LB} and B_v^{UB} in $\mathcal{O}(|V|)$ by evaluating the events v systematically in the correct sequence to ensure that all predecessors (with $(w, v) \in A$) and successors (with $(v, w) \in A$), respectively, are evaluated before. B_v^{LB} is computed in the following sequence:

1. v_1 is a pick-up location and ...

- (a) $v_2 = \dots = v_Q = 0$, i.e., all events where a customer is entering an empty vehicle.
- (b) $v_2 \neq 0$, and $v_3 = \dots = v_Q = 0$.
- (c) $v_2, v_3 \neq 0$, and $v_4 = \dots = v_Q = 0$.
- (d) ...
- (e) $v_2, \dots, v_Q \neq 0$.

2. v_1 is a delivery location.

Note that the vehicle could also drive from a delivery location to a pick-up location. However, because of the triangle inequality this path cannot be shorter than not visiting either the pick-up nor the delivery location of the corresponding customer. Therefore, it is reasonable to require $w \in \bigcup_{i=1}^n V_i^+$ in (6). For the same reason we only need B_w^{LB} of events where w_1 is a pick-up location to compute B_v^{LB} for events where v_1 is a delivery location.

Because of the triangle inequality the shortest path between an event v and the end depot $(0, \dots, 0)$ is to deliver customers v_1, \dots, v_Q without picking up a new customer. When we first iterate over $v \in V$ to compute B_v^{UB} , maximum ride times cannot be considered yet, because we need the upper bounds B_v^{UB} of all pick-up nodes to compute \bar{B}_i^{UB} . However, for the computation of B_v^{UB} , where v is a pick-up node, we need the upper bounds of all delivery nodes. Hence, in the first loop, we omit maximum ride times from the computation of B_v^{UB} . The following sequence is most efficient to determine B_v^{UB} :

1. v_1 is a delivery location and ...

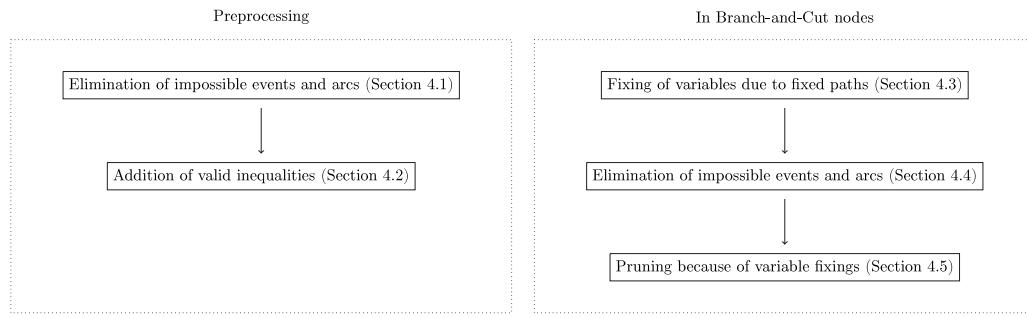


Fig. 2. Overview of developed methods.

- (a) $v_2 = \dots = v_Q = 0$, i.e., all events where v_1 is the last customer leaving the vehicle.
- (b) $v_2 \neq 0$, and $v_3 = \dots = v_Q = 0$.
- (c) $v_2, v_3 \neq 0$, and $v_4 = \dots = v_Q = 0$.
- (d) ...
- (e) $v_2, \dots, v_Q \neq 0$.

2. v_1 is a pick-up location.

In total, all events in V have to be considered once to determine B_v^{LB} and once to determine B_v^{UB} without consideration of maximum ride times. After all upper bounds B_v^{UB} have been computed, we are able to determine \bar{B}_i^{UB} , $i \in R$. We repeat the procedure to compute B_v^{UB} , this time including ride times. So, one iteration of the procedure requires an effort of $\mathcal{O}(|V|)$.

After computing B_v^{UB} for an event v , we can directly check whether $B_v^{UB} < B_v^{LB}$ and delete the event if so. Due to our sequence to compute B_v^{UB} values, the event is not considered in (7) any more to compute upper bounds for its predecessors.

However, if an event v with $v_1 \in P$ is deleted, lower bounds B_w^{LB} of its successor events with $(v, w) \in A$ might also change. Therefore, we store all of these successors in a list and update their lower bounds afterwards. When going through the list, deleting an event w can also lead to updated upper bounds for predecessors if $w_1 \in D$. Moreover, \bar{B}_i^{UB} can change if B_v^{UB} changes for a $v \in V_{i^+}$, $i \in R$. Thus, we need three lists, one for predecessors, one for successors, and one to store all i for which \bar{B}_i^{UB} has to be updated. Then, we run alternately through the lists in the sequence described above until the lists are empty. In the worst case, only one bound changes in each iteration.

Whenever updating a lower or upper bound of event v , we also check feasibility of all in- and outgoing arcs $((w, v) \in \delta^{in}(v)$ and $(v, w) \in \delta^{out}(v)$, respectively). In our tests, the computation of lower and upper bounds for all nodes and the detection of infeasible arcs was $< 0.5s$ for instances with $n = 100$ and $Q = 6$.

We can use the bounds to add the following valid inequalities to the EB model formulation

$$B_v^{LB} + (\ell_{i^-} - L_i - s_i - B_v^{LB}) \left(1 - \sum_{a \in \delta^{in}(v)} x_a\right) \leq B_v \leq B_v^{UB} \quad \forall i \in R, v \in V_{i^+} \quad (8)$$

$$B_v^{LB} \leq B_v \leq e_{i^+} + s_i + L_i + (B_v^{UB} - (e_{i^+} + s_i + L_i)) \sum_{a \in \delta^{in}(v)} x_a \quad \forall i \in R, v \in V_{i^-} \quad (9)$$

and the following valid inequalities to the location-augmented-event-based formulations:

$$\sum_{v: v_1=j} B_v^{LB} \cdot \sum_{a \in \delta^{in}(v)} x_a \leq \bar{B}_j \leq \sum_{v: v_1=j} B_v^{UB} \cdot \sum_{a \in \delta^{in}(v)} x_a \quad \forall j \in P \cup D \quad (10)$$

4.2. Addition of valid inequalities in preprocessing

We can add several further valid inequalities to the model formulation identifying events which cannot occur simultaneously.

4.2.1. Infeasible paths

In the literature, several authors introduced infeasible paths constraints (e.g. Ascheuer et al., 2000; Cordeau, 2006; Ropke et al., 2007) based on sequences of request locations which cannot occur due to time windows. Certainly, if a path of locations $i \rightarrow j \rightarrow k$ would lead to a violated time window, this is also true for all event paths $v \rightarrow w \rightarrow u$ with $v_1 = i$, $w_1 = j$, and $u_1 = k$. However, it might be that the event path $v \rightarrow w \rightarrow u$ is infeasible although the location path $i \rightarrow j \rightarrow k$ is feasible (e.g. because in event v another customer is in the vehicle who leads to a later departure time at i). Thus, it is possible to add a larger amount of infeasible path constraints based on events.

Let v be an event. Compute for all events w with $(w, v) \in A$ the earliest starting time $B_{wv}^{LB} = B_w^{LB} + s_{w_1} + t_{(w,v)}$ at v coming from w and for all events u with $(v, u) \in A$ the latest possible starting time of service $B_{vu}^{UB} = B_u^{UB} - s_{v_1} - t_{(v,u)}$ in v to reach u on time. If $B_{vu}^{UB} < B_{wv}^{LB}$, $x_{(w,v)} + x_{(v,u)} \leq 1$ is a valid inequality which can be lifted to

$$\sum_{w' \in V : (w', v) \in A \wedge B_{w'v}^{LB} \geq B_{wv}^{LB}} x_{(w', v)} + \sum_{u' \in V : (v, u') \in A \wedge B_{vu'}^{UB} \leq B_{vu}^{UB}} x_{(v, u')} \leq 1 \quad \forall v \in V \setminus V_0 \quad (11)$$

by including other predecessor (successor) events leading to a later earliest (earlier latest) starting time of service in v .

In general, let $\bar{S} = \{(u^1, u^2), \dots, (u^{\bar{m}-1}, u^{\bar{m}})\}$ be a path $u^1 \rightarrow u^2 \rightarrow \dots \rightarrow u^{\bar{m}}$, which is infeasible if $B_{u^1}^{LB} + \sum_{m=1}^{\bar{m}-1} s_{u^m} + t_{(u^m, u^{m+1})} > B_{u^{\bar{m}}}^{UB}$, i.e., if location $u^{\bar{m}}$ is not reached before its latest departure time. Then, $\sum_{a \in \bar{S}} x_a \leq \bar{m} - 2$ is a valid inequality.

This valid inequality can be lifted in two ways as Fig. 3 shows. In line with the argumentation above, $\sum_{a \in \bar{S}} x_a \leq \bar{m} - 2$ can be lifted by adding $x_{(v, u^2)}$ if $B_{vu^2}^{LB} \geq B_{u^1 u^2}^{LB}$ and $x_{(u^{\bar{m}-1}, w)} \leq B_{u^{\bar{m}-1} w}^{UB} \leq B_{u^{\bar{m}-1} u^{\bar{m}}}^{UB}$ on the left side. Moreover, further paths between u^2 and $u^{\bar{m}-1}$ can be added if $(i^-, 0, \dots, 0) \notin \{u^2, \dots, u^{\bar{m}-1}\}$ for all $i \in R$, i.e., the vehicle is not empty in between. In Fig. 3, $u_{i^+}^m$ is the event where customer i enters the vehicle and all customers who are in the vehicle after event u^m are still there. Analogously, $u_{i^-}^m$ is the event where customer i leaves the vehicle and all customers who are present in the vehicle directly before event u^m are in the vehicle. After each of the path's locations the vehicle can leave it to visit the pick-up location of a further customer i and come back to the path when customer i is delivered. This might be directly, i.e., the path $u^m \rightarrow u_{i^+}^m \rightarrow u_{i^-}^{m+1} \rightarrow u^{m+1}$ replaces the path $u^m \rightarrow u^{m+1}$, or indirectly, i.e., the vehicle also visits further locations between pick-up and delivery of customer i or drives from i 's delivery location to a later location of the original path. In the direct case, arcs $(u^m, u_{i^+}^m)$ and $(u_{i^-}^{m+1}, u^{m+1})$ replace arc (u^m, u^{m+1}) such that we have to weight both added arc variables with a factor $1/2$ to ensure feasibility of the lifted inequality. Note that we only "replace" the original arc conceptually in the solution, but we do not actually remove it from the lifted inequality. With respect to feasibility of the lifting, there are three cases:

- In generalization of the already discussed situation, $u^m \rightarrow u_{i^+}^m \rightarrow \dots \rightarrow u_{i^-}^{m+1} \rightarrow u^{m+1}$ replaces $u^m \rightarrow u^{m+1}$, which is a feasible lifting, as one arc with a factor of 1 is replaced by two arcs with a factor

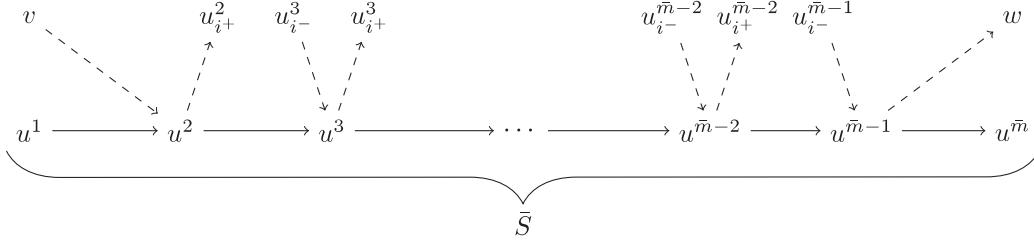


Fig. 3. Lifting for infeasible paths.

of $1/2$. Note that we do not need to consider the path between customer i 's pick-up and delivery. Due to the triangle inequality, the arrival time at u^{m+1} cannot be earlier than in the original path independent of the path between customer i 's pick-up and delivery.

- $u^m \rightarrow u_{i+}^m \rightarrow \dots \rightarrow u_{i-}^{m'} \rightarrow u^{m'}$ with $m' > m + 1$ replaces the path $u^m \rightarrow \dots \rightarrow u^{m'}$, which is a feasible lifting, as at least two arcs with a factor of 1 ((u^m, u^{m+1}) and $(u^{m'-1}, u^{m'})$) are replaced by two arcs with a factor of $1/2$. Again, we do not need to consider the path between customer i 's pick-up and delivery.
- If any path including arcs (u_{i+}^m, u^m) and $(u^{m'}, u_{i+}^{m'})$ with $m' \geq m$ is part of the lifted path, the path is already infeasible due to precedence relations.

Note that the event u_{i+}^m (u_{i-}^m) or arc (u^m, u_{i+}^m) ((u_{i-}^m, u^m)) might not exist for some $i \in R$ and $m \in \{2, \dots, \bar{m} - 2\}$. Then, the corresponding arcs are simply not added to the valid inequality. As all other corresponding arcs are added, the lifting includes also paths where the delivery location visited right before the vehicle returns to the original path needs not to belong to the same customer as the pick-up location visited directly after leaving the original path. Together,

$$\sum_{(v,u^2) \in A : v \neq u_1 : B_{vu^2}^{LB} \geq B_{u^1u^2}^{LB}} x_{(v,u^2)} + \sum_{a \in \bar{S}} x_a + \sum_{m=2}^{\bar{m}-2} \sum_{(u^m,w) \in A : w_1 \in P} \frac{1}{2} \cdot x_{(u^m,w)} \\ + \sum_{m=3}^{\bar{m}-1} \sum_{(u^m) \in A : w_1 \in D} \frac{1}{2} \cdot x_{(u^m,w)} + \sum_{(u^{\bar{m}-1},w) \in A, w \neq u^{\bar{m}}, B_{u^{\bar{m}-1},w}^{UB} \leq B_{u^{\bar{m}-1},u^{\bar{m}}}^{UB}} x_{(u^{\bar{m}-1},w)} \\ \leq \bar{m} - 2 \quad (12)$$

is a valid inequality. We implemented inequalities (12) for the case of $\bar{m} = 4$. Details on the implementation of inequalities (11) and (12) can be found in Appendix D. In total, for each arc $(v,u) \in A$ we construct one inequality of type (11), hence there are $\mathcal{O}(|A|)$ valid inequalities. In the case of inequalities (12), we construct each inequality from an arc $(u^2, u^3) \in A$ and u^3 's successors, hence there are $\mathcal{O}(|A| \cdot (|R| + |Q|))$ valid inequalities.

4.2.2. Vehicle sharing

Let i and j be two customers. If they use the same vehicle, only one of them can enter the vehicle while the other is already sitting in it. Both of these events are incompatible to an event where i or j leaves the vehicle before the other one entered it. Thus,

$$\sum_{(v,w) \in A : w_1 = j^+ \wedge \exists l : w_l = i} x_{(v,w)} + \sum_{(v,w) \in A : w_1 = i^+ \wedge \exists l : w_l = j} x_{(v,w)} \\ + \sum_{(v,w) \in A : v_1 = i^- \wedge w_1 = j^+} x_{(v,w)} + \sum_{(v,w) \in A : v_1 = j^- \wedge w_1 = i^+} x_{(v,w)} \leq 1 \quad \forall i, j \in R : i < j \quad (13)$$

are valid inequalities. We have $\mathcal{O}(|R|^2)$ valid inequalities of type (13) and add all of them in the preprocessing to the model formulation.

Analogously, only one of them can leave the vehicle while the other is still sitting in it or one of them can leave the vehicle before the other

Table 1
Example travel time matrix for inequalities (13)–(16).

	0	1 ⁺	2 ⁺	3 ⁺	1 ⁻	2 ⁻	3 ⁻
0	0	5	5	5	5	5	5
1 ⁺	5	0	1	1	5	5	5
2 ⁺	5	1	0	1	5	5	5
3 ⁺	5	1	1	0	5	5	5
1 ⁻	5	1	1	1	0	1	1
2 ⁻	5	1	1	1	1	0	1
3 ⁻	5	1	1	1	1	1	0

enters it. Hence,

$$\sum_{(v,w) \in A : w_1 = i^- \wedge \exists l : w_l = j} x_{(v,w)} + \sum_{(v,w) \in A : w_1 = j^- \wedge \exists l : w_l = i} x_{(v,w)} \\ + \sum_{(v,w) \in A : v_1 = i^- \wedge w_1 = j^+} x_{(v,w)} + \sum_{(v,w) \in A : v_1 = j^- \wedge w_1 = i^+} x_{(v,w)} \leq 1 \quad \forall i, j \in R : i < j \quad (14)$$

are valid inequalities. We have $\mathcal{O}(|R|^2)$ valid inequalities of type (14) and add all of them in the preprocessing to the model formulation.

Moreover, an event where i or j leaves the vehicle directly after the other one, is incompatible to an event where one of them left the vehicle directly before the other one entered it. Thus,

$$\sum_{(v,w) \in A : v_1 = i^- \wedge w_1 = j^-} x_{(v,w)} + \sum_{(v,w) \in A : v_1 = j^- \wedge w_1 = i^-} x_{(v,w)} \\ + \sum_{(v,w) \in A : v_1 = i^- \wedge w_1 = j^+} x_{(v,w)} + \sum_{(v,w) \in A : v_1 = j^- \wedge w_1 = i^+} x_{(v,w)} \leq 1 \quad \forall i, j \in R : i < j \quad (15)$$

are valid inequalities. We have $\mathcal{O}(|R|^2)$ valid inequalities of type (15) and add all of them in the preprocessing to the model formulation.

Furthermore, if we select an arc (v,w) with $v_1 \in D$ and $w_1 \in P$, customers v_1 and w_1 cannot share a vehicle simultaneously. Thus,

$$\sum_{(v,w) \in A : v_1 = j^- \wedge w_1 = i^+} x_{(v,w)} + \sum_{(v,w) \in A : v_1 = i^- \wedge w_1 = j^+} x_{(v,w)} \\ + \sum_{(u',u) \in A : u_1 = k \wedge \exists l_1 : u_{l_1} = i \wedge \exists l_2 : u_{l_2} = j} x_{(u',u)} \leq 1 \quad (16)$$

is a valid inequality for all pairs of customers $i, j \in R$, $i < j$, being pairwise compatible and a further location $k \in J \setminus \{0\}$. We have $\mathcal{O}(|J| \cdot |R|^2)$ valid inequalities of type (16) and add all of them in the preprocessing to the model formulation.

An example illustrating that inequalities (13)–(16) in fact cut off fractional solutions which are optimal for the LP relaxation of the LAEB formulation is given below.

Example 1. We consider an instance with $R = \{1, 2, 3\}$, $Q = 3$, $K = 1$, $L_i = 100$ for all $i \in R$, and $q_j = 1$, $s_j = 0$, and $[e_j, \ell_j] = [5, 60]$ for all $j \in P \cup D$ as well as $q_0 = 0$, $s_0 = 0$, and $[e_0, \ell_0] = [0, 100]$. The travel time matrix is given in Table 1.

The LP relaxation of the LAEB formulation returns the optimal solution pictured in Fig. 4. For the sake of clarity only arcs with positive flow are included and the values of the variables \bar{B}_j , $j \in J$, are omitted.

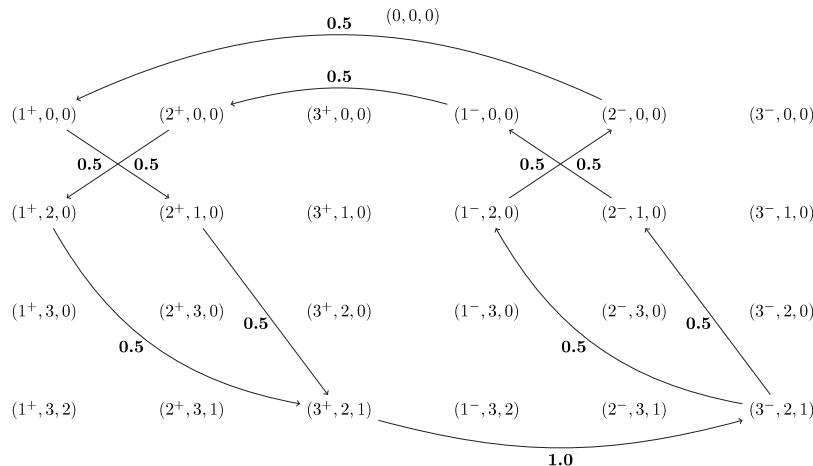


Fig. 4. Solution returned by the LP relaxation of the LAEB formulation

This solution violates each of the inequalities (13)–(16) for $i = 1$ and $j = 2$. For example, inequality (13) results in

$$x_{((1+,0,0),(2+,1,0))} + x_{((3-,1,0),(2+,1,0))} + x_{((1+,3,0),(2+,3,1))} + x_{((3+,1,0),(2+,3,1))} \\ + x_{((2+,0,0),(1+,2,0))} + x_{((3-,2,0),(1+,2,0))} + x_{((2+,3,0),(1+,3,2))} + x_{((3+,2,0),(1+,3,2))} \\ + x_{((1-,0,0),(2+,0,0))} + x_{((1-,3,0),(2+,3,0))} \\ + x_{((2-,0,0),(1+,0,0))} + x_{((2-,3,0),(1+,3,0))} \leq 1.$$

With the values given by the solution depicted in Fig. 4, the left hand side of the inequality is equal to 2 because the first variable in each row above has a value of 0.5, so the inequality is clearly violated. Note that due to the fact that time consistency constraints (3b) need not be binding in a fractional solution, the depot is not part of the tour in this example.

4.2.3. Customer incompatibility

With the pairwise incompatibilities presented in Section 2.2 we can identify in advance customer pairs i and j which cannot be served by the same vehicle due to time windows. Our procedure in Section 4.1 eliminates all events where i and j are simultaneously in the vehicle. If the paths $i^+ \rightarrow i^- \rightarrow j^+ \rightarrow j^-$ and $j^+ \rightarrow j^- \rightarrow i^+ \rightarrow i^-$ are infeasible due to time windows, we can conclude that at most one of them can share the vehicle with another customer k . Otherwise customers i and j would be served by the same vehicle (not necessarily at the same time) due to transitivity. Thus,

$$\sum_{(u,v) \in A : v_l = k^+ \wedge \exists l : v_l = i} x_{(u,v)} + \sum_{(u,v) \in A : v_l = i^+ \wedge \exists l : v_l = k} x_{(u,v)} \\ + \sum_{(u,v) \in A : v_l = k^+ \wedge \exists l : v_l = j} x_{(u,v)} + \sum_{(u,v) \in A : v_l = j^+ \wedge \exists l : v_l = k} x_{(u,v)} \leq 1 \quad (17)$$

is a valid inequality. We have $\mathcal{O}(|R|^3)$ valid inequalities of type (17). However, the valid inequality is only relevant if i and j are incompatible, but both of them are compatible with k . We add all of them in the preprocessing.

4.3. Fixing of variables due to fixed paths

Knowing which customers are together in a fixed path, i.e., a sequence of events connected by arcs whose variables are fixed to 1, also leads to further incompatible events if the depot is part of the path (Schulz & Pfeiffer, 2024). We update the set of fixed paths when an upward branch, i.e., a branch where an x or \bar{x} variable is fixed to 1, is created. Let two fixed paths be given which start in the depot. As both have to be served by different vehicles, customers in one of the paths cannot share a vehicle with customers of the other path. Let customer i be in the first and customer j be in the second path. Then,

we can eliminate all events v with $v_{l_1} \in \{i, i^+, i^-\}$ and $v_{l_2} \in \{j, j^+, j^-\}$ with $l_1, l_2 \in \{1, \dots, Q\}$ and their incident arcs. The same is true if we consider two fixed paths ending in the depot. Thus, we check such incompatibilities whenever a fixed path is merged with another fixed path containing the depot.

4.4. Elimination of impossible events and arcs in branch-and-cut nodes

Variable fixings in branch-and-cut nodes influence the lower and upper bounds computed in the preprocessing, see Section 4.1. While for the EB formulation and the LAEB formulation of the DARP, we branch on variables x_a , $a \in A$, using the ALAEB formulation branching takes place on variables \bar{x}_{ij} , $i, j \in \bar{J}$. If a variable $x_{(v,w)}$ is fixed to 0 or to 1, either directly because a new branch for the EB or the LAEB model is created, or as an indirect result of another fixed variable $x_{(v',w')}$ or \bar{x}_{ij} , the effects on upper and lower bounds are as follows: If $x_{(v,w)}$ is fixed to 0, w cannot be the successor of v any more. Thus, w can be excluded in the maximum in (7). Analogously, w can be excluded in the minimum in (6) if $x_{(w,v)}$ is fixed to 0. If we fix $x_{(v,w)}$ to 1, w is determined as the successor of v such that we can replace (6) by

$$B_w^{LB} = \max\{e_{w_1}, B_v^{LB} + s_{v_1} + t_{(v,w)}\} \quad (6')$$

and (7) by

$$B_v^{UB} = \min \left\{ \ell_{v_1} - \sum_{i \in R} \mathbb{1}_{\{v_1=i^-\}} \cdot \max \{0, \ell_{v_1} - (\bar{B}_i^{UB} + s_i + L_i)\}, B_w^{UB} - t_{(v,w)} - s_{v_1}, \right. \\ \left. \min_{i \in R} \exists i \geq 2 : v_i = i \left\{ \bar{B}_i^{UB} + s_i + L_i - s_{v_1} - t_{v_1, i^-} \right\} \right\}. \quad (7)$$

In other words, the events v and w are merged to a single event if $x_{(v,w)} = 1$ (cf. Schulz & Pfeiffer, 2024). Moreover, we eliminate events and arcs from the graph which have become infeasible due to variable fixings in a branch-and-cut node. For example, if arc $x_{(v,w)}$ has been fixed to 1 and w is a pick-up node, we know that all other events $u \in V$, $u_1 = w_1$, $u \neq w$, are infeasible for the subtree rooted in the current branch-and-cut node.

After we ran the procedures described in this and the previous section, we store all events v for which either B_v^{LB} or B_v^{UB} changed and update lower and upper bounds for the beginning of service of all possible predecessor and successor events as described at the end of Section 4.1. Moreover, we check feasibility of events v and arcs (v, w) , i.e., check if $B_v^{LB} \leq B_w^{UB}$ and $B_v^{LB} + s_{v,w} + t_{(v,w)} \leq B_w^{UB}$, respectively.

If fixing of variables $x_{(v,w)}$ or the update of lower and upper bound results in an infeasible event, the event is removed from the graph

Table 2
Abbreviations used in Section 5.

Abbreviation	Explanation/ Reference
General abbreviations	
Inst.	Name of test instance
Obj.	Objective Value
EB	Event-based MILP
LAEB	Location-augmented-event-based MILP
ALAEB	Aggregated location-augmented-event-based MILP
N/A	Not applicable (no integer solution found within the time limit)
# Imp.	Number of instances with improved computational time
CPU	CPU time in seconds
CPU Graph	CPU time in seconds needed for graph preprocessing, only mentioned if >0.5 s
Avg. Deviation	Average deviation of CPU time compared to the benchmark (first row of the corresponding table)
Avg.* Deviation	Avg. Deviation only on instances with improved computational time
rGap	root node gap
Avg. rGap	Average root node gap
Preprocessing	
GP	Graph preprocessing, see Section 4.1
VS1–VS4	Vehicle Sharing, see (13)–(16)
CI1	Customer Incompatibility, see (17)
IP1–IP2	Infeasible Paths, see (11)–(12)
Branch-and-Cut Algorithm	
LBsUBs	add new inequalities of type (8)–(10) if bounds changed, see Section 4.4
FA	add inequalities $x_a \leq 0$ for infeasible arcs a , see Section 4.4
FP	update the set of fixed paths and eliminate events (and incident arcs) which represent users of two different paths connected to the start or end depot together in the vehicle, see Section 4.3

together with its incident arcs. Finally, we add new inequalities of types (8)–(10) if bounds changed, and inequalities $x_a \leq 0$ if the arc a has been declared infeasible.

4.5. Pruning because of variable fixing in branch-and-cut nodes

If one of the eliminated events v is already merged with other events, i.e., is incident to an arc for which the corresponding arc variable is fixed to 1, the branch-and-cut node can be pruned.

5. Computational study

In this section, we evaluate the efficiency of the presented MILP models, lower and upper bounding strategies, and valid inequalities. Comparing the EB and the LAEB model, the same branching decisions can be applied, as in both models branching takes place on variables x_a , $a \in A$. The LAEB model has fewer variables, as we replace variables B_v , $v \in V$, by variables \tilde{B}_j , $j \in \bar{J}$. Hence, we expect the LAEB model to be more efficient. Comparing our two location-augmented-event-based formulations, the LAEB model has fewer variables but more binary variables. Moreover, variable fixing in Section 4.5 can more likely be applied, as we directly branch on the arcs of the event-based graph. Therefore, a branch where an x_a , $a \in A$, variable is fixed to 1 directly implies that both adjacent events occur. This branching step contains significantly more information than just fixing a single relation between two locations. On the other hand, a sequence of variables \bar{x}_{ij} has to be fixed to 1 to ensure that an event occurs if we branch on \bar{x}_{ij} variables, but we have significant fewer \bar{x}_{ij} variables. Taking everything together, it is unclear which advantage predominates. We compare all three formulations in a computational study.

Our computations were performed on an Intel Core i7-8700 CPU, 3.20 GHz, 32 GB memory and implemented in C++ using CPLEX 12.10. Note, that the execution of CPLEX was limited to one thread only and the search method was limited to traditional branch-and-cut. The time limit for the solution in all tests was set to 7200 seconds. Throughout this section, we use some abbreviations which are summarized in Table 2. We first analyze the performance of the presented MILP formulations (Section 5.1). Then, our preprocessing components are validated in Section 5.2. Afterwards, we test the components for the branch-and-cut algorithm (Section 5.3). We present our final results on benchmark instances in Section 5.4.

5.1. Comparing the MILP formulations

The basic versions (without any of the methods presented in Section 4) of EB, LAEB, and ALAEB formulations are compared against each other using the a- and b-benchmark instances introduced in Cordeau (2006) and Ropke et al. (2007). Furthermore, we use an extended version of these instances denoted by a-X* and b-X* (adapted from Gschwind and Irnich (2015)), where the time window length is doubled by postponing the upper bound of the pick-up or delivery time window by 15 minutes, i.e., $\ell_{i+} = 15$. Due to the larger time windows these instances are harder to solve. Note, that we do not increase the vehicle capacity as done in Gschwind and Irnich (2015). A vehicle capacity of $Q = 12$ is a rather unrealistic assumption for ridepooling services (compare, e.g., the ridepooling cabs of MOIA¹ or HolMichApp,² both based in Germany, which have a capacity of $Q = 6$). Previous studies have shown that even six seats are rarely fully occupied in practical relevant settings (Pfeiffer & Schulz, 2022). Thus, extended time windows are more relevant for the practical application than larger vehicles. A second reason is that our approach is tailored to the case of ridepooling applications, and the complexity of our models strongly increases for larger vehicle capacities such as $Q = 12$. For the sake of completeness, we include the results of the LAEB model combined with preprocessing techniques for the original a-X and b-X instances with doubled time window length and doubled vehicle capacity in the appendix in Tables E.12 and E.13. A large majority of the instances can still be solved in a moderate amount of time, but other approaches, see e.g., Gschwind and Irnich (2015), perform better here.

The instances are denoted as aK-n, bK-n or aK-n-X* and bK-n-X*, respectively, where K indicates the number of vehicles and n denotes the number of requests. In the a- and a-X*-instances, $Q = 3$ and $L_i = 30$ and $q_i = 1$ for all $i \in R$, whereas in the b- and b-X*-instances $Q = 6$ and $L_i = 45$ and $q_i \in \{1, \dots, 6\}$ for all $i \in R$.

The results are presented in Tables 3 and 4. Note, that the computational times for the EB formulation reported here are not comparable to the results reported by Gaul et al. (2022), as in their paper the execution of CPLEX was limited to 12 threads whereas it is limited to one thread in this paper. On the smaller as well as on the larger

¹ <https://www.moia.io>

² <https://www.holmich-app.de/>

Table 3
Comparing the plain MILP formulations using the a- and b- instances.

Inst.	Obj.	EB			LAEB			ALAEB		
		Gap	CPU	rGap	Gap	CPU	rGap	Gap	CPU	rGap
a2-16	294.3		0.03	0.8%		0.01	0.8%		0.03	0.8%
a2-20	344.9		0.03	0.0%		0.01	0.0%		0.02	0.0%
a2-24	431.1		0.09	1.9%		0.03	2.1%		0.06	2.1%
a3-18	300.5		0.04	1.8%		0.06	1.8%		0.04	1.8%
a3-24	344.8		0.14	0.5%		0.08	2.8%		0.3	3.0%
a3-30	494.8		0.05	0.0%		0.02	0.0%		0.08	0.6%
a3-36	583.2		0.21	2.5%		0.13	2.5%		0.16	2.5%
a4-16	282.7		0.35	0.7%		0.04	0.7%		0.05	0.7%
a4-24	375.0		0.05	0.0%		0.02	0.0%		0.03	0.1%
a4-32	485.5		0.43	1.8%		0.12	2.0%		0.24	2.1%
a4-40	557.7		0.83	1.7%		1.11	1.8%		2.04	2.8%
a4-48	668.8		0.39	2.0%		0.47	2.2%		2.66	4.1%
a5-40	498.4		0.24	1.1%		0.11	1.2%		0.32	1.2%
a5-50	686.6		1.85	2.6%		1.8	2.6%		77.42	4.0%
a5-60	808.3		0.72	2.0%		0.93	2.1%		16.71	2.2%
a6-48	604.1		0.68	0.7%		0.42	0.6%		3.85	2.2%
a6-60	819.3		9.61	2.4%		9.59	2.4%		85.33	3.2%
a6-72	916.1		16.75	2.5%		28.49	2.6%		656.72	3.5%
a7-56	724.0		4.85	1.3%		1.76	1.6%		54.59	3.1%
a7-70	875.7		14.99	1.2%		4.47	1.3%		199.76	2.2%
a7-84	1033.3		20.72	1.9%		16.59	2.1%		1535.79	4.6%
a8-64	747.5		8.35	2.0%		5.59	1.9%		420.34	2.9%
a8-80	945.8		23.06	2.7%		15.84	2.8%	1.0%	2h	4.5%
a8-96	1229.7	1.0%	2h	4.1%	1.1%	2h	4.1%	2.9%	2h	5.2%
Total/Avg.			7304	1.6%		7287	1.8%		17456	2.5%
b2-16	309.4		0.03	0.6%		0.07	0.6%		0.04	0.6%
b2-20	332.7		0.01	0.0%		0.04	0.0%		0.01	0.0%
b2-24	444.7		0.05	1.7%		0.05	1.8%		0.05	1.8%
b3-18	301.6		0.04	2.0%		0.05	2.2%		0.04	2.2%
b3-24	394.5		0.26	2.2%		0.19	2.2%		0.16	2.2%
b3-30	531.4		0.03	0.5%		0.03	0.5%		0.03	0.5%
b3-36	603.8		0.05	0.3%		0.04	0.3%		0.04	0.3%
b4-16	296.9		0.01	1.6%		0.03	1.7%		0.01	1.7%
b4-24	371.4		0.04	0.7%		0.04	0.8%		0.05	0.8%
b4-32	494.9		0.03	0.0%		0.03	0.0%		0.03	0.0%
b4-40	656.6		0.09	0.3%		0.07	0.3%		0.13	0.3%
b4-48	673.8		0.76	0.9%		0.42	1.0%		2.42	1.1%
b5-40	613.7		0.17	0.4%		0.09	0.5%		0.1	0.4%
b5-50	761.4		0.42	1.4%		0.2	1.5%		0.55	1.6%
b5-60	902.0		0.99	1.8%		2.16	1.9%		7.74	1.9%
b6-48	714.8		0.22	0.5%		0.12	0.5%		0.13	0.5%
b6-60	860.0		0.28	0.6%		0.17	0.6%		0.25	0.6%
b6-72	978.5		8.85	1.0%		5.64	1.0%		26.95	1.0%
b7-56	824.0		6.79	0.9%		4.01	0.9%		22.72	0.9%
b7-70	912.6		2.51	2.0%		2.77	2.3%		9.12	2.5%
b7-84	1203.3		2.87	1.2%		3.07	1.2%		6.3	1.3%
b8-64	839.9		2.27	2.3%		1.1	2.3%		5.56	2.3%
b8-80	1036.4		3.49	1.3%		1.46	1.9%		16.89	1.9%
b8-96	1185.6		78.05	1.4%		34.43	1.4%		287.43	1.7%
Total/Avg.			108	1.1%		56	1.1%		386	1.2%

benchmark instances, the LAEB model outperforms the other two formulations regarding the required computation time, but its superiority becomes most evident when comparing its performance to the other formulations on the harder a-X* and b-X* instances. Regarding the root node gap the EB formulation is slightly superior. However, note that the EB formulation requires equal sized time windows. Although they are given in the benchmark instances, this is not necessarily true in general.

The average root node gap for all three formulations is very small and ranges from 1.1% to 5.4%, which demonstrates that, although some instances still take a long time to solve, the MILP formulations are already very tight. To sum it up, it seems that the smaller number of integer variables in the ALAEB formulation does not make up for the loss of information compared to the EB or LAEB formulation when branching on \bar{x}_{ij} variables instead of $x_{(v,w)}$. Comparing the EB and the LAEB model, the replacement of variables B_v , $v \in V$, (of which there may be $\mathcal{O}(n^Q)$ (see Gaul et al., 2022)) by variables B_j , $j \in J$, (of which there are $2n$ variables) as well as the tighter big-M formulation explain

the presented speedup. Since the LAEB formulation improves on the EB formulation and turns out to be the best formulation, we restrict ourselves to this formulation in the remaining parts of the numerical tests.

5.2. Validating preprocessing components

We proposed a preprocessing procedure to eliminate infeasible nodes and arcs (Section 4.1) and three types of new valid inequalities (Section 4.2).

To evaluate the effect of the aforementioned methods, we use the following test set:

- instances a6-72, a8-80, b6-72, and b8-96 introduced above
- instances Q3n80_2, Q3n80_3, Q3n80_4, Q6n80_5, Q6n100_1, and Q6n100_5 from the artificial instance test set used in Gaul et al. (2022)

Table 4
Comparing the plain MILP formulations using the a-X* and b-X* instances.

Inst.	EB				LAEB				ALAEB			
	Obj.	Gap	CPU	rGap	Obj.	Gap	CPU	rGap	Obj.	Gap	CPU	rGap
a2-16-X*	278.2		0.14	1.3%	278.2		0.08	1.7%	278.2		0.09	1.7%
a2-20-X*	330.7		0.03	2.0%	330.7		0.02	2.0%	330.7		0.05	2.4%
a2-24-X*	389.1		0.26	2.6%	389.1		0.20	2.6%	389.1		0.62	2.8%
a3-18-X*	272.7		0.07	2.5%	272.7		0.11	3.0%	272.7		0.39	7.0%
a3-24-X*	289.6		0.73	2.2%	289.6		0.75	3.5%	289.6		1.99	3.5%
a3-30-X*	452.8		0.34	5.0%	452.8		0.28	5.4%	452.8		1.32	5.6%
a3-36-X*	501.0		0.37	1.7%	501.0		0.29	1.9%	501.0		1.08	1.9%
a4-16-X*	235.2		0.43	5.9%	235.2		0.34	5.9%	235.2		1.54	5.9%
a4-24-X*	359.4		0.37	2.3%	359.4		0.12	2.5%	359.4		0.71	4.2%
a4-32-X*	447.3		6.39	3.8%	447.3		2.54	3.9%	447.3		19.78	4.5%
a4-40-X*	509.0		38.56	2.9%	509.0		21.93	3.0%	509.0		85.36	3.2%
a4-48-X*	620.3		1194.49	6.4%	620.3		660.95	7.8%	622.5	2.5%	2h	10.6%
a5-40-X*	464.0		16.77	3.1%	464.0		8.79	3.3%	464.0		33.64	4.5%
a5-50-X*	621.9		285.24	5.8%	621.9		213.63	5.5%	621.9	0.5%	2h	6.6%
a5-60-X*	745.4		281.67	3.6%	745.4		254.23	4.0%	745.4	0.5%	2h	6.2%
a6-48-X*	572.5		7082.45	5.5%	572.5		2675.00	5.6%	572.5	1.1%	2h	6.4%
a6-60-X*	757.9		1993.91	3.7%	757.9		855.54	4.6%	757.9	1.8%	2h	5.5%
a6-72-X*	N/A	N/A	2h	N/A	868.4	3.4%	2h	6.8%	869.6	4.5%	2h	8.5%
a7-56-X*	663.5		4411.85	5.0%	663.5		5131.52	5.7%	668.9	1.9%	2h	6.7%
a7-70-X*	815.3	0.6%	2h	6.2%	815.3		1592.16	6.4%	815.3	3.0%	2h	8.0%
a7-84-X*	N/A	N/A	2h	N/A	N/A		2h	N/A	N/A	N/A	2h	N/A
a8-64-X*	702.6	1.9%	2h	5.6%	701.2	0.9%	2h	5.5%	701.4	2.8%	2h	7.1%
a8-80-X*	N/A	N/A	2h	N/A	N/A		2h	N/A	N/A	N/A	2h	N/A
a8-96-X*	N/A	N/A	2h	N/A	N/A		2h	N/A	N/A	N/A	2h	N/A
Total/Avg.			58514	3.9%			47418	4.3%		86546		5.4%
b2-16-X*	282.5		0.31	2.2%	282.5		0.40	2.4%	282.5		0.08	2.4%
b2-20-X*	323.6		0.02	3.0%	323.6		0.02	3.0%	323.6		0.04	3.0%
b2-24-X*	412.3		0.03	0.0%	412.3		0.02	0.0%	412.3		0.02	0.0%
b3-18-X*	290.4		0.06	3.0%	290.4		0.04	3.0%	290.4		0.07	3.0%
b3-24-X*	363.7		0.12	0.4%	363.7		0.06	0.4%	363.7		0.28	0.5%
b3-30-X*	504.3		0.17	2.3%	504.3		0.20	2.3%	504.3		0.26	2.3%
b3-36-X*	565.9		0.10	0.9%	565.9		0.05	0.9%	565.9		0.06	0.9%
b4-16-X*	289.9		0.03	2.2%	289.9		0.02	2.3%	289.9		0.02	2.3%
b4-24-X*	347.0		3.55	4.4%	347.0		0.95	5.0%	347.0		2.17	5.0%
b4-32-X*	491.0		0.07	1.0%	491.0		0.05	1.0%	491.0		0.06	1.0%
b4-40-X*	628.3		0.21	1.0%	628.3		0.25	1.1%	628.3		0.37	1.1%
b4-48-X*	627.4		4.89	1.4%	627.4		1.61	1.4%	627.4		6.68	2.0%
b5-40-X*	585.1		5.53	2.7%	585.1		1.72	2.8%	585.1		24.13	2.7%
b5-50-X*	708.8		2.31	0.8%	708.8		1.12	0.8%	708.8		5.33	0.8%
b5-60-X*	851.9		10.72	1.8%	851.9		5.03	1.9%	851.9		25.92	2.6%
b6-48-X*	691.6		2.58	1.7%	691.6		0.79	1.7%	691.6		1.94	2.2%
b6-60-X*	841.6		2.32	1.7%	841.6		0.63	1.8%	841.6		5.14	1.8%
b6-72-X*	930.3		52.73	1.7%	930.3		49.56	1.5%	930.3		31.17	1.9%
b7-56-X*	787.9		142.77	1.4%	787.9		4.32	1.5%	787.9		285.5	1.5%
b7-70-X*	865.3		19.34	2.6%	865.3		14.68	2.5%	865.3		132.21	2.8%
b7-84-X*	1141.2		3676.19	3.2%	1141.2		2428.40	3.2%	1141.9	1.0%	2h	3.6%
b8-64-X*	818.3		12.07	1.9%	818.3		9.16	2.0%	818.3		24.68	2.4%
b8-80-X*	998.3		16.10	2.2%	998.3		14.04	2.5%	998.3		27.32	2.5%
b8-96-X*	1137.7	0.3%	2h	2.0%	1137.7		3425.66	2.1%	1139.5	1.1%	2h	2.7%
Total/Avg.			11152	1.9%			5958	2.0%		14973		2.1%

- five real data instances from HolMichApp containing one full day of completed trips each

The characteristics of the instances are summarized in Table 5. The instances have been selected to provide adequate diversity in terms of share of requests per vehicle, share of number of requests per service duration, and length of time windows.

First, we test the efficiency of graph preprocessing (GP). On average, the number of nodes is reduced by 32%, and the number of arcs by 12%. As the number of nodes and arcs translates directly to the number of variables in the MILP, GP leads to a reduction of 22% of the variables in total. Table 6 shows the results for a combination of GP with different subsets of valid inequalities. Note, that the computational times in Table 6 refer to the solution times reported by CPLEX. The time needed to execute GP and add additional valid inequalities to the LAEB formulation is only a fraction of a second. The first row shows the total solution time for all instances when no preprocessing components

are switched on. In the following rows, we test all components and all but one component to be switched on and compare computation times to the first row, where no preprocessing takes place. We achieve an average deviation in computation time ranging from -17 to -52% over all test instances and an average deviation ranging from -64% to -82% including only those test instances where the required computation time was smaller in comparison to the basic setting in the first row. The significant difference between the average deviation over all instances and the average deviation only over the instances with an improvement underlines the heterogeneity of the test set. In the next part of the table, we compare the influence of only one of the valid inequalities switched on (and GP switched on, since otherwise the comparison would be unfair, as IP1 and IP2 cannot be switched on without GP). The improvements in the second part of the table are not as strong as the improvement in the first part, indicating that a combination of multiple valid inequalities and GP is most promising. Taking a look at the average root node gap, we are able to reduce it from 2.9% to

Table 5
Characteristics of the test instances.

Inst.	n	K	T	TW	L_i	Q
a6-72	72	6	720	15	30	3
a8-80	80	8	600	15	30	3
b6-72	72	6	720	15	45	6
b8-96	96	8	720	15	45	6
Q3n80_2	80	11	240	15	$1.5 \bar{t}_i$	3
Q3n80_3	80	11	240	15	$1.5 \bar{t}_i$	3
Q3n80_4	80	10	240	15	$1.5 \bar{t}_i$	3
Q6n80_5	80	12	240	15	$1.5 \bar{t}_i$	6
Q6n100_1	100	17	240	15	$1.5 \bar{t}_i$	6
Q6n100_5	100	14	240	15	$1.5 \bar{t}_i$	6
2021-10-05	85	8	960	25	$\bar{t}_i + \max(10, 0.75\bar{t}_i)$	6
2021-10-07	89	8	960	25	$\bar{t}_i + \max(10, 0.75\bar{t}_i)$	6
2021-10-09	97	12	1080	25	$\bar{t}_i + \max(10, 0.75\bar{t}_i)$	6
2021-10-16	120	12	1080	25	$\bar{t}_i + \max(10, 0.75\bar{t}_i)$	6
2021-10-18	68	2	960	25	$\bar{t}_i + \max(10, 0.75\bar{t}_i)$	6

1.9% with multiple combinations of preprocessing steps. This again highlights the tightness of the LAEB formulation, but also underlines the efficiency of the preprocessing methods. It turns out that for this set of test instances, the best combination is the one given in the bottom row of the table, leading to an average reduction of 58% in CPU time. Although the test instances are very diverse, we cannot exclude that for other sets of test instances another combination prevails, so that we suggest to use a combination of all of the presented preprocessing methods.

After verifying the new presented MILP formulation and preprocessing methods, we next examine the influence of the presented strategies in branch-and-cut nodes.

5.3. Testing branch-and-cut algorithm

The lower part of Table 2 gives an overview of the methods introduced in Sections 4.3–4.5 to reduce the size of the graph in branch-and-cut nodes. Following our strategy to evaluate individual components, we solved the LAEB model together with all of the presented preprocessing steps once for each subset of branch-and-cut components on the set of test instances introduced in the previous section. The results are listed in Table 7.

The first row of Table 7 shows the total computational time for the LAEB model and preprocessing steps. In the next rows, all three, two or only one of the presented branching techniques are switched on. Note, that we did not include the case where neither LBsUBs nor FA are switched on, since in this case no valid inequalities would be added. We were able to achieve further improvements of about 25% compared to the total computational time after applying preprocessing techniques in about a third of the instances. However, taking the whole set of test instances into account, there is an overall increase ranging from 54% to 66% in CPU. This may be explained by the fact that the overhead caused by the implementation of our cuts in CPLEX is not balanced by the profit from using additional cuts. Also the average root node gap was very small already such that we might not get low enough in the branch-and-cut tree to fully use the benefit of the presented methods. Nevertheless, the results underline again the heterogeneity of the instances and show that the presented methods are very effective on certain instances.

5.4. Results on benchmark instances

In this section, we show results on the benchmark instances using the LAEB formulation and the preprocessing methods. The results are shown in Table 9. On the a-instances, the total computational time was reduced by 94% compared to using the plain LAEB or the plain EB formulation. On the b-instances, the computational time was reduced by

59% and 79%, respectively (compare Table 3). On the a-X* instances, the computational time was reduced by 27% compared to using the plain LAEB formulation and by 41% compared to the EB formulation. On the b-X* instances, the computational time was reduced by 92% and 96%, respectively. These results prove the efficiency of the new LAEB formulation and the high impact of the proposed preprocessing techniques.

Although computation times on different computers are only comparable to a limited extent, our results are competitive with the branch-and-cut algorithm by Gschwind and Irnich (2015): While some of the a-instances were solved faster by our approach and some of them were solved faster by their branch-and-cut algorithm, the average solution times over all b-instances are smaller by a factor of about 50 using our methods. However, our solution times cannot compete with the speed of the branch-and-cut algorithm developed by Rist and Forbes (2021) but could still prove useful in practice, as the implementation of a MILP formulation and preprocessing methods is very fast and easy. Moreover, the results show that our methods are able to solve instances of substantial size within seconds. This is especially true, as our results show that the model performs even better for instances where groups are transported (b and b-X* instances) which is an important case for ridepooling providers. The results in this section are still for one thread only. Since we do not use user-defined cuts if we omit the methods presented in Sections 4.3–4.5, we do not need to limit the number of threads used by CPLEX or to restrict the search method to traditional branch-and-cut. Using 12 threads the whole set of a- and b-instances was solved in 62 s. The corresponding results for the a-X* and b-X* instances show that medium-sized and some of the large instances of the harder benchmark set can be solved within a few seconds (compare Table 8). For a complete view, results on 12 threads using the EB formulation are illustrated in the appendix in Table F.14 as well.

6. Conclusion

The new presented MILP formulations combine the existing formulations, the location- and the event-based formulation, of the DARP. As shown in Theorem 2, the LAEB formulation is tighter than the LB formulation. Computational tests on large benchmark instances show that, using the LAEB formulation, computational times can be reduced by 19% (a-X* instances) and 47% (b-X* instances), respectively, compared to the EB formulation. Additionally, graph preprocessing and the introduction of new valid inequalities which eliminate subtours, infeasible events, and infeasible paths strongly further improve computational times by 27% (a-X*) instances and 92% (b-X*), respectively. An average root node gap of 1.6% proves that our formulation and preprocessing methods generate a very tight MILP model. Our methods are useful for application in praxis, as the implementation is fast and easy and even medium-sized instances of the harder benchmark set can be solved within a few seconds (compare Table 8).

While the literature mostly focuses on the static DARP, on-demand ridepooling services need to include customer requests on time when they arise. Gaul et al. (2021) showed that the dynamic DARP can be solved using a rolling-horizon algorithm in which the EB formulation is updated and solved whenever new requests arrive. Since the LAEB formulation improves on the EB formulation, this strategy can directly be adapted by using the LAEB formulation instead.

CRediT authorship contribution statement

Daniela Gaul: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Kathrin Klamroth:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Investigation, Funding acquisition, Conceptualization. **Christian Pfeiffer:** Writing – review & editing, Writing

Table 6

Comparing individual and joint effect of preprocessing components on the test instances. A 0 represents a component to be switched off and a 1 represents it to be switched on.

GP	VS1	VS2	VS3	VS4	CI1	IP1	IP2	# Imp.	Total CPU	R. gap	Avg. Deviation	Avg.* Deviation
0	0	0	0	0	0	0	0	-	407.89	2.9%	-	-
1	1	1	1	1	1	1	1	12	340.57	1.9%	-17%	-64%
1	0	1	1	1	1	1	1	12	195.70	1.9%	-52%	-82%
1	1	0	1	1	1	1	1	12	196.69	1.9%	-52%	-79%
1	1	1	0	1	1	1	1	13	278.88	1.8%	-32%	-68%
1	1	1	1	0	1	1	1	13	289.63	1.9%	-29%	-75%
1	1	1	1	1	0	1	1	12	306.77	1.9%	-25%	-65%
1	1	1	1	1	1	0	1	13	322.33	2.0%	-21%	-70%
1	1	0	0	0	0	0	0	11	299.61	2.6%	-27%	-66%
1	0	1	0	0	0	0	0	10	345.21	2.5%	-15%	-46%
1	0	0	1	0	0	0	0	9	425.08	2.6%	4%	-59%
1	0	0	0	1	0	0	0	10	390.51	2.7%	-4%	-71%
1	0	0	0	0	1	0	0	11	302.21	2.6%	-26%	-60%
1	0	0	0	0	0	1	0	12	168.40	2.4%	-58%	-74%
1	0	0	0	0	0	0	1	13	307.55	2.1%	-25%	-69%
1	0	0	0	0	1	1	1	14	171.31	1.9%	-58%	-75%

Table 7

Evaluating the performance of the branch-and-cut algorithm. A 0 represents a component to be switched off and a 1 represents it to be switched on.

LBsUBs	FA	FP	Avg. rGap	# Imp.	Total CPU	Avg. Deviation	Avg.* Deviation
0	0	0	1.9%	-	340.57	-	-
1	1	1	1.9%	5	583.29	66%	-25%
0	1	1	1.9%	6	557.65	54%	-20%
1	0	1	1.9%	6	512.79	56%	-25%
1	1	0	1.9%	5	580.06	66%	-25%
1	0	0	1.9%	6	508.42	56%	-25%
0	1	0	1.9%	6	555.10	54%	-20%

Table 8

LAEB model + preprocessing on 12 threads.

Inst.	Obj.	CPU	Inst.	Obj.	CPU	Inst.	Obj.	Gap	CPU	Inst.	Obj.	CPU
a2-16	294.3	0.23	b2-16	309.4	0.11	a2-16-X*	278.2	0.13	b2-16-X*	282.5	0.12	
a2-20	344.9	0.03	b2-20	332.7	0.05	a2-20-X*	330.7	0.03	b2-20-X*	323.6	0.06	
a2-24	431.1	0.06	b2-24	444.7	0.05	a2-24-X*	389.1	0.09	b2-24-X*	412.3	0.03	
a3-18	300.5	0.05	b3-18	301.6	0.07	a3-18-X*	272.7	0.09	b3-18-X*	290.4	0.08	
a3-24	344.8	0.07	b3-24	394.5	0.07	a3-24-X*	289.6	0.34	b3-24-X*	363.7	0.12	
a3-30	494.8	0.04	b3-30	531.4	0.03	a3-30-X*	452.8	0.14	b3-30-X*	504.3	0.14	
a3-36	583.2	0.08	b3-36	603.8	0.08	a3-36-X*	501	0.2	b3-36-X*	565.9	0.07	
a4-16	282.7	0.06	b4-16	296.9	0.05	a4-16-X*	235.2	0.2	b4-16-X*	289.9	0.03	
a4-24	375	0.04	b4-24	371.4	0.07	a4-24-X*	359.4	0.15	b4-24-X*	347	0.27	
a4-32	485.5	0.14	b4-32	494.9	0.04	a4-32-X*	447.3	0.76	b4-32-X*	491	0.1	
a4-40	557.7	0.17	b4-40	656.6	0.11	a4-40-X*	509	1.75	b4-40-X*	628.3	0.21	
a4-48	668.8	0.32	b4-48	673.8	0.41	a4-48-X*	620.3	15.88	b4-48-X*	627.4	1.18	
a5-40	498.4	0.13	b5-40	613.7	0.13	a5-40-X*	464	2.17	b5-40-X*	585.1	1.39	
a5-50	686.6	0.73	b5-50	761.4	0.23	a5-50-X*	621.9	15.94	b5-50-X*	708.8	0.48	
a5-60	808.3	0.49	b5-60	902	0.64	a5-60-X*	745.4	16.68	b5-60-X*	851.9	1.72	
a6-48	604.1	0.37	b6-48	714.8	0.18	a6-48-X*	572.5	101.85	b6-48-X*	691.6	0.54	
a6-60	819.3	1.01	b6-60	860	0.22	a6-60-X*	757.9	100.03	b6-60-X*	841.6	1.04	
a6-72	916.1	6.57	b6-72	978.5	1.47	a6-72-X*	868.4	4368.93	b6-72-X*	930.3	6.08	
a7-56	724	0.68	b7-56	824	1.1	a7-56-X*	663.5	102.02	b7-56-X*	787.9	6.5	
a7-70	875.7	1.02	b7-70	912.6	0.89	a7-70-X*	815.3	100.12	b7-70-X*	865.3	2.77	
a7-84	1033.3	2.52	b7-84	1203.3	1.35	a7-84-X*	950.6	6396.27	b7-84-X*	1141.2	50.73	
a8-64	747.5	1.49	b8-64	839.9	0.96	a8-64-X*	701.2	191.76	b8-64-X*	818.3	3.04	
a8-80	945.8	4.98	b8-80	1036.4	0.68	a8-80-X*	880.3	4579.24	b8-80-X*	998.3	2.73	
a8-96	1229.7	26.96	b8-96	1185.6	5.15	a8-96-X*	1118.5	2.3% 2h	b8-96-X*	1137.7	147.31	
Total		48		14				23196		226		

Table 9

Results for the benchmark instances solved using the LAEB formulation and preprocessing.

Inst.	Obj.	CPU	rGap	Inst.	Obj.	CPU	rGap	Inst.	Obj.	Gap	CPU	rGap	Inst.	Obj.	CPU	rGap
a2-16	294.3	0.10	0.0%	b2-16	309.4	0.17	0.6%	a2-16-X*	278.2	0.07	0.0%	b2-16-X*	282.5	0.57	1.3%	
a2-20	344.9	0.02	0.0%	b2-20	332.7	0.02	0.0%	a2-20-X*	330.7	0.03	0.0%	b2-20-X*	323.6	0.02	0.8%	
a2-24	431.1	0.05	0.8%	b2-24	444.7	0.07	1.0%	a2-24-X*	389.1	0.13	1.4%	b2-24-X*	412.3	0.03	0.0%	
a3-18	300.5	0.02	0.3%	b3-18	301.6	0.03	0.9%	a3-18-X*	272.7	0.11	2.8%	b3-18-X*	290.4	0.05	1.3%	
a3-24	344.8	0.07	1.6%	b3-24	394.5	0.12	1.9%	a3-24-X*	289.6	0.74	3.3%	b3-24-X*	363.7	0.08	0.4%	
a3-30	494.8	0.06	0.0%	b3-30	531.4	0.03	0.0%	a3-30-X*	452.8	0.27	2.2%	b3-30-X*	504.3	0.16	1.6%	
a3-36	583.2	0.17	2.5%	b3-36	603.8	0.05	0.1%	a3-36-X*	501.0	0.48	0.6%	b3-36-X*	565.9	0.07	0.0%	
a4-16	282.7	0.04	0.7%	b4-16	296.9	0.02	0.6%	a4-16-X*	235.2	0.50	5.4%	b4-16-X*	289.9	0.02	1.7%	
a4-24	375.0	0.04	0.0%	b4-24	371.4	0.04	0.1%	a4-24-X*	359.4	0.15	1.5%	b4-24-X*	347.0	0.47	2.1%	
a4-32	485.5	0.12	1.1%	b4-32	494.9	0.04	0.0%	a4-32-X*	447.3	1.78	3.4%	b4-32-X*	491.0	0.08	0.7%	
a4-40	557.7	0.28	1.2%	b4-40	656.6	0.09	0.3%	a4-40-X*	509.0	0.95	2.1%	b4-40-X*	628.3	0.28	0.1%	
a4-48	668.8	0.27	1.7%	b4-48	673.8	0.65	0.7%	a4-48-X*	620.3	56.36	5.5%	b4-48-X*	627.4	1.46	0.9%	
a5-40	498.4	0.16	0.6%	b5-40	613.7	0.12	0.4%	a5-40-X*	464.0	6.12	2.9%	b5-40-X*	585.1	2.65	1.9%	
a5-50	686.6	1.51	2.0%	b5-50	761.4	0.21	1.0%	a5-50-X*	621.9	73.02	4.0%	b5-50-X*	708.8	0.53	0.6%	
a5-60	808.3	0.45	1.4%	b5-60	902.0	0.86	1.6%	a5-60-X*	745.4	39.78	3.3%	b5-60-X*	851.9	1.66	1.3%	
a6-48	604.1	0.38	0.4%	b6-48	714.8	0.18	0.3%	a6-48-X*	572.5	935.75	4.7%	b6-48-X*	691.6	0.53	0.4%	
a6-60	819.3	2.14	1.6%	b6-60	860.0	0.24	0.1%	a6-60-X*	757.9	332.38	3.2%	b6-60-X*	841.6	1.13	1.3%	
a6-72	916.1	11.34	2.2%	b6-72	978.5	3.95	0.9%	a6-72-X*	869.6	2.1%	2h	b6-72-X*	930.3	7.16	1.4%	
a7-56	724.0	2.32	1.2%	b7-56	824.0	1.46	0.8%	a7-56-X*	663.5	551.48	4.1%	b7-56-X*	787.9	5.06	0.8%	
a7-70	875.7	2.11	0.9%	b7-70	912.6	0.65	1.5%	a7-70-X*	815.3	711.65	4.7%	b7-70-X*	865.3	6.11	1.6%	
a7-84	1033.3	11.13	1.6%	b7-84	1203.3	1.50	0.8%	a7-84-X*	950.6	1.4%	2h	b7-84-X*	1141.2	89.10	2.1%	
a8-64	747.5	2.29	1.1%	b8-64	839.9	0.70	1.6%	a8-64-X*	701.2	3042.70	4.1%	b8-64-X*	818.3	2.38	1.8%	
a8-80	945.8	7.58	1.9%	b8-80	1036.4	0.64	0.8%	a8-80-X*	880.3	2.7%	2h	b8-80-X*	998.3	6.00	1.7%	
a8-96	1229.7	425.06	2.8%	b8-96	1185.6	11.13	1.1%	a8-96-X*	N/A	N/A	2h	b8-96-X*	1137.7	340.51	2.3%	
Total		467	1.2%			22	0.7%				34554	3.4%			466	1.2%

– original draft, Software, Methodology, Investigation, Conceptualization. **Michael Stiglmayr:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization. **Arne Schulz:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Project administration, Methodology, Investigation, Formal analysis, Conceptualization.

Acknowledgments

This work was partially supported by the state of North Rhine Westphalia (Germany) within the project “bergisch.smart_mobility”.

Appendix A. Notation

See Table A.10.

Appendix B. Proof of Theorem 1

Constraints (1h), (1j), and (2b)–(2d) are part of both model formulations. As we consider the LP relaxation, constraints (2k) and (5c) are identical. Thus, we have to show that (3a)–(3b) are equivalent to (1k), (4), and (5a)–(5b).

Starting with (1k), (4), and (5a)–(5b), we can replace \bar{x}_{ij} in objective (5a) and constraints (5b) by $\sum_{(v,w) \in A: v_1=i \wedge w_1=j} x_{(v,w)}$ (Eqs. (4)) and obtain objective (3a) and constraints (3b), respectively. Moreover, Eqs. (4) and constraints (2b) and (2c) imply $0 \leq \bar{x}_{ij} \leq 1$ for all $i, j \in J$. Thus, constraints (1k) are redundant in the LP relaxation of ALAEB. However, then \bar{x}_{ij} is only set in (4) but not used in the model any more such that (4) is also redundant and we obtain (3a)–(3b).

Analogously, we can start with (3a)–(3b), add the redundant Eqs. (4) and constraints (1k). Using Eqs. (4), we can replace

$\sum_{(v,w) \in A: v_1=i \wedge w_1=j} x_{(v,w)}$ by \bar{x}_{ij} in objective (3a) and constraints (3b) to obtain objective (5a) and constraints (5b), respectively. By this, we obtain (1k), (4), and (5a)–(5b).

Appendix C. Proof of Theorem 2

The proof consists of three steps. First, we prove that every feasible LP solution of the LAEB formulation is also a feasible LP solution of the

LB formulation, i.e., $Z_{LAEB}^{rel} \geq Z_{LB}^{rel}$. Second, we give an example that there is a feasible LP solution for the LB formulation which is not LP feasible for the LAEB formulation. Third, we give a concrete instance in which $Z_{LAEB}^{rel} > Z_{LB}^{rel}$ holds.

- Let $\mathbf{x} = (x_1, \dots, x_{|A|})$ be a feasible LP solution for the LAEB formulation. Set $\bar{x}_{ij} = \sum_{v: v_1=i, w: w_1=j} x_{(v,w)}$ for all $i, j \in J$. Furthermore, define $\bar{x}_{0+j} = \bar{x}_{0-j} = \bar{x}_{0j}$ and $\bar{x}_{i0+} = \bar{x}_{i0-} = \bar{x}_{i0}$. Due to (2c) the ingoing flow is 1 for each pick-up location i^+ . Because of (2b) this holds also for the outgoing flow. All outgoing arcs of a node v with $v_1 = i^+$ end in a node w with $w_j = i$ for one $j = 2, \dots, Q$ until a node w with $w_1 = i^-$ is reached. This holds also for all nodes connected by an arc with node w . With the same argument this holds for their connected nodes and so on. On the other hand there is no arc between a node w' with $w'_1 \neq i^+$ and $w'_l \neq i$ for all $l = 2, \dots, Q$ and a node w with $w_1 = i^-$. Thus, constraints (2b) lead to

$$\sum_{(v,w): w_1=i^+} x_{(v,w)} = 1 = \sum_{(v,w): w_1=i^-} x_{(v,w)}. \quad (\text{C.1})$$

Together, (1b) and (1c) are fulfilled. With our choice of \bar{x}_{ij} , (2d) leads to

$$\sum_{i \in P} \bar{x}_{0+j} = \sum_{(v,w): v_1=0} x_{(v,w)} \leq K,$$

which means that (1d) is fulfilled.

There is one constraint in (1e) for each $S \in S$. Each S consists of a path starting in the starting depot 0^+ and visiting the delivery location of a request i without having visited the respective pick-up location before. In the EB formulation, all flow starts in the depot (2d), as for all other nodes the flow constraint (2b) holds. Besides, a flow of 1 enters an event-node associated with the pick-up location of i due to (2c). Thus, and due to our definition of \bar{x}_{ij} , a flow of 1 has to leave set S . Moreover, due to construction of the event-based graph and our definition of \bar{x}_{ij} , all flow leaving a pick-up location of i has to go to a delivery location of i , i.e., enter set S again. In total, the flow amongst nodes in S cannot exceed $|S| - 2$. Thus, constraints (1e) are fulfilled.

Table A.10
Notation used in the paper.

B_v	variable in the EB formulation indicating start of service at location v_1
B_v^{LB}	if event v occurs
B_v^{UB}	lower bound for B_v
B_v^U	upper bound for B_v
B_{vw}^{LB}	earliest feasible beginning of service at node w , when $x_{(v,w)} = 1$
B_{vw}^{UB}	latest feasible beginning of service at node v , when $x_{(v,w)} = 1$
B_i^{UB}	$= \max_{w \in V_A} \{B_{vw}^{UB}\}$, i.e., upper bound on beginning of service at location i^+
\tilde{B}_j	variable indicating start of service at location $j \in J$ or $j \in \bar{J}$, respectively
\tilde{c}_{ij}	costs for traveling between locations $i, j \in J$ or $i, j \in \bar{J}$, respectively
c_a	costs for traveling arc $a = (v, w) \in A$, i.e., between locations v_1 and w_1
D	set of delivery locations, i.e., $D = \{1^-, \dots, n^-\}$
$\delta^{in}(v)$	$= \{(w, v) \in A\}$, i.e., ingoing arcs of node v
$\delta^{out}(v)$	$= \{(v, w) \in A\}$, i.e., outgoing arcs of node v
e_j	earliest possible start of service at location $j \in J$ or $j \in \bar{J}$, respectively
f_{ij}^1, f_{ij}^2	feasibility of the paths $j^+ \rightarrow i^+ \rightarrow j^- \rightarrow i^-$ and $j^+ \rightarrow i^+ \rightarrow i^- \rightarrow j^-$, respectively, w.r.t. ride time and time window constraints, $i, j \in R$
J	set of all locations in the event-based and location-augmented-event-based formulations, i.e., $J = P \cup D \cup \{0\}$
\bar{J}	set of locations in the LB formulation, i.e., $\bar{J} = J \setminus \{0\} \cup 0^+, 0^-$
i^+, i^-	pick-up and delivery location of customer i
K	number of vehicles
L_i	maximum ride time of customer i
ℓ_j	latest possible start of service at location $j \in J$ or $j \in \bar{J}$, respectively
n	number of customers
P	set of pick-up locations, i.e., $P = \{1^+, \dots, n^+\}$
Q	vehicle capacity
Q_j	variable in the LB model indicating the vehicle load after visiting location $j \in \bar{J}$
q_j	number of persons entering or leaving the vehicle at location $j \in J$ or $j \in \bar{J}$, respectively
R	set of all requests, i.e., $R = \{1, \dots, n\}$
s_j	service time at location $j \in J$ or $j \in \bar{J}$, respectively
T	service duration
TW	$= \ell_j - e_j$ length of pick-up (inbound requests) or delivery (outbound requests) time window
\tilde{t}_{ij}	travel time between locations $i, j \in J$ or $i, j \in \bar{J}$, respectively
\tilde{t}_i	travel time for a direct ride from pick-up to delivery location of request i
$t_{(v,w)}$	travel time between locations v_1 and w_1
V	set of nodes in event-based graph
V_0	event $(0, \dots, 0)$
V_{i^+}	set of nodes $v \in V$ with $v_1 = i^+$
V_{i^-}	set of nodes $v \in V$ with $v_1 = i^-$
x_a	binary variable in EB and LAEB model, continuous variable in ALAEB model
\tilde{x}_{ij}	binary variable in LB and ALAEB model

Furthermore, constraints (1g) and (1i) are fulfilled by $Q_{0^+} = Q_{0^-} = 0$ and

$$Q_j = \mathbb{1}_{\{j \in P\}} q_j + \sum_{(u,v) \in A: v_1=j} x_{(u,v)} \sum_{l=2}^Q q_{v_l} \quad \forall j \in P \cup D$$

whereat $\mathbb{1}_{\{j \in P\}}$ is the indicator function, i.e.,

$$\mathbb{1}_{\{j \in P\}} = \begin{cases} 1 & \text{if } j \in P \\ 0 & \text{otherwise.} \end{cases}$$

As all terms are non-negative, $Q_j \geq 0$ for all $j \in \bar{J}$. For pick-up locations $Q_j \geq q_j$; thus, $Q_j \geq \max\{0, \mathbb{1}_{\{j \in P\}} q_j\}$ for all $j \in \bar{J}$. Due to construction of event nodes,

$$\sum_{l=1}^Q q_{v_l} \leq Q \quad \forall v \in V$$

and

$$\sum_{l=2}^Q q_{v_l} \leq Q + \mathbb{1}_{\{j \in P\}} q_j \quad \forall v \in V$$

for delivery locations. Thus, $Q_j \leq \min\{Q, Q + \mathbb{1}_{\{j \in P\}} q_j\}$ for all $j \in \bar{J}$ and constraints (1i) are fulfilled.

Due to construction of the event-based graph, there are no arcs (v, w) with $v_1 = i$ and $w_1 = j$ and

$$\mathbb{1}_{\{i \in P\}} q_i + \sum_{l=2}^Q q_{v_l} + \mathbb{1}_{\{j \in P\}} q_j > \mathbb{1}_{\{j \in P\}} q_j + \sum_{l=2}^Q q_{w_l}$$

for any pair $i, j \in \bar{J}$. Thus,

$$\mathbb{1}_{\{i \in P\}} q_i + \sum_{l=2}^Q q_{v_l} + \mathbb{1}_{\{j \in P\}} q_j \leq \mathbb{1}_{\{j \in P\}} q_j + \sum_{l=2}^Q q_{w_l}$$

holds for every $i, j \in \bar{J}$ if $x_{(v,w)} > 0$ with $v_1 = i$ and $w_1 = j$ and thereby

$$\begin{aligned} & \sum_{(v,w) \in A: v_1=i \wedge w_1=j} x_{(v,w)} \left(\mathbb{1}_{\{i \in P\}} q_i + \sum_{l=2}^Q q_{v_l} + \mathbb{1}_{\{j \in P\}} q_j \right) \\ & \leq \sum_{(v,w) \in A: v_1=i \wedge w_1=j} x_{(v,w)} \left(\mathbb{1}_{\{j \in P\}} q_j + \sum_{l=2}^Q q_{w_l} \right) \end{aligned} \quad (\text{C.2})$$

for every $i, j \in \bar{J}$. Moreover,

$$\mathbb{1}_{\{i \in P\}} q_i + \sum_{l=2}^Q q_{v_l} + \mathbb{1}_{\{j \in P\}} q_j - Q \leq \mathbb{1}_{\{j \in P\}} q_j \quad \forall i, j \in \bar{J}, v \in V : v_1 = i$$

event-based

location-based

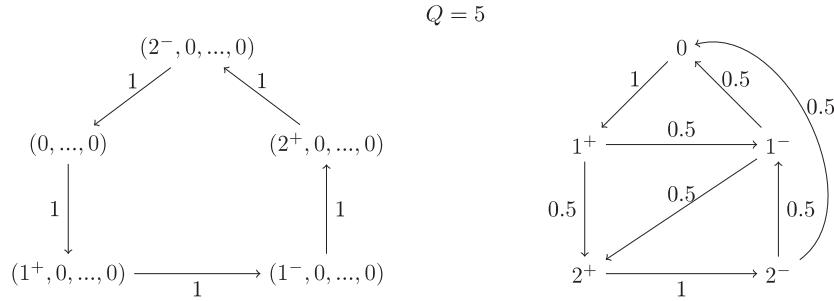
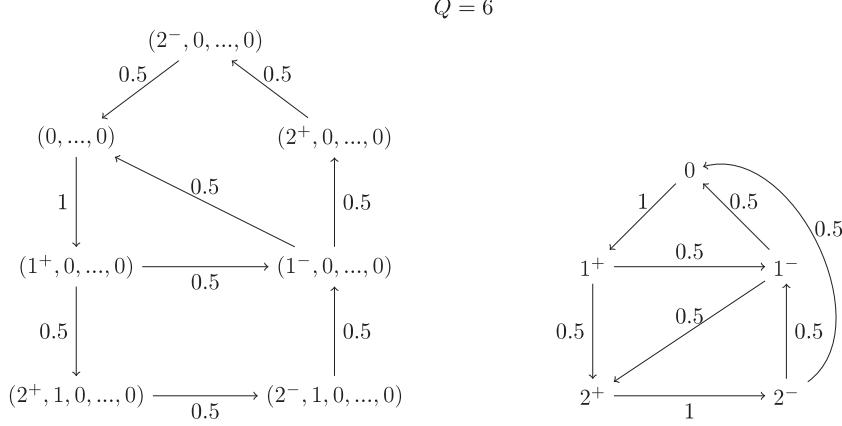


Fig. C.5. Example for step two in the proof of Theorem 2.

holds because of construction of the events. Thus,

$$\begin{aligned} & \left(1 - \sum_{(v,w) \in A: v_1 = i \wedge w_1 = j} x_{(v,w)}\right) \left(\mathbb{1}_{\{j \in P\}} q_j + \sum_{l=2}^Q q_{v_l} + I_{\{j \in P\}} q_j - Q \right) \\ & \leq \left(1 - \sum_{(v,w) \in A: v_1 = i \wedge w_1 = j} x_{(v,w)}\right) \mathbb{1}_{\{j \in P\}} q_j \quad \forall i, j \in \bar{J}. \end{aligned} \quad (\text{C.3})$$

Together, we get

$$\begin{aligned} & Q_i + I_{\{j \in P\}} q_j - Q \left(1 - \sum_{(v,w) \in A: v_1 = i \wedge w_1 = j} x_{(v,w)}\right) \\ & = \mathbb{1}_{\{i \in P\}} q_i + \underbrace{\sum_{(u,v) \in A: v_1 = i} x_{(u,v)} \sum_{l=2}^Q q_{v_l}}_{\leq 1} + I_{\{j \in P\}} q_j \\ & \quad - Q \left(1 - \sum_{(v,w) \in A: v_1 = i \wedge w_1 = j} x_{(v,w)}\right) \\ & \leq \mathbb{1}_{\{i \in P\}} q_i + \sum_{l=2}^Q q_{v_l} + I_{\{j \in P\}} q_j - Q \left(1 - \sum_{(v,w) \in A: v_1 = i \wedge w_1 = j} x_{(v,w)}\right) \\ & \stackrel{(\text{C.2})-(\text{C.3})}{\leq} \left(1 - \sum_{(v,w) \in A: v_1 = i \wedge w_1 = j} x_{(v,w)}\right) \mathbb{1}_{\{j \in P\}} q_j \\ & \quad + \sum_{(v,w) \in A: v_1 = i \wedge w_1 = j} x_{(v,w)} \left(\mathbb{1}_{\{j \in P\}} q_j + \sum_{l=2}^Q q_{w_l} \right) \end{aligned}$$

$$\begin{aligned} & \leq \mathbb{1}_{\{j \in P\}} q_j + \sum_{(u,w) \in A: w_1 = j} x_{(u,w)} \sum_{l=2}^Q q_{w_l} \\ & = Q_j \quad \forall i, j \in \bar{J}. \end{aligned}$$

With our choice of \bar{x}_{ij} constraints (1g) are fulfilled. Together, the first step follows.

2. Fig. C.5 shows an instance with two requests, $q_1 = q_2 = 3$, and one vehicle. Furthermore, we assume that time windows are not binding. The figure shows feasible LP solutions for the EB formulation on the left side and for the LB formulation on the right side. Note that only arcs with positive value for $x_{(v,w)}$ and \bar{x}_{ij} , respectively, are included in the figure. In the upper part of the figure, the vehicle capacity is 6, i.e., both customers can be transported simultaneously. The graphs of both approaches show the solutions

$$\begin{aligned} & 0 \rightarrow 1^+ \rightarrow 2^+ \rightarrow 2^- \rightarrow 1^- \rightarrow 0 \\ & 0 \rightarrow 1^+ \rightarrow 1^- \rightarrow 2^+ \rightarrow 2^- \rightarrow 0 \end{aligned}$$

both with a weight of 0.5. If we reduce the vehicle capacity to 5, the first solution is not integer feasible any more. While the event-based graph pictures this fact (lower left part of the figure), the solution in the LB formulation does not change. The reason is that capacity constraints are still fulfilled if the first solution is weighted with 0.5 and $Q_0 = 0, Q_{1^+} = 3, Q_{1^-} = 0$,

Table C.11Example for step three in the proof of [Theorem 2](#).

	0	1 ⁺	2 ⁺	1 ⁻	2 ⁻
0	0	1	1	10	10
1 ⁺	1	0	2	9	9
2 ⁺	1	2	0	9	9
1 ⁻	10	9	9	0	1
2 ⁻	10	9	9	1	0

$Q_{2^+} = 5$, and $Q_{2^-} = 2$ as the following evaluation shows:

$$\begin{aligned}
 Q_i + I_{\{j \in P\}} q_j - Q(1 - x_{ij}) &\leq Q_j \\
 0 \rightarrow 1^+ &0 + 3 - 5 \cdot (1 - 0.5 - 0.5) = 3 & \leq 3 \\
 1^+ \rightarrow 1^- &3 - 3 - 5 \cdot 0.5 = -2.5 & \leq 0 \\
 1^+ \rightarrow 2^+ &3 + 3 - 5 \cdot 0.5 = 3.5 & \leq 5 \\
 1^- \rightarrow 0 &0 + 0 - 5 \cdot 0.5 = -2.5 & \leq 0 \\
 1^- \rightarrow 2^+ &0 + 3 - 5 \cdot 0.5 = 0.5 & \leq 5 \\
 2^+ \rightarrow 2^- &5 - 3 - 5 \cdot (1 - 0.5 - 0.5) = 2 & \leq 2 \\
 2^- \rightarrow 0 &2 + 0 - 5 \cdot 0.5 = -0.5 & \leq 0 \\
 2^- \rightarrow 1^- &2 - 3 - 5 \cdot 0.5 = -3.5 & \leq 0
 \end{aligned}$$

In all other cases, $\bar{x}_{ij} = 0$. Since $Q_j \geq \max\{0, I_{\{j \in P\}} q_j\}$, it holds that $Q_i - Q \leq 0 \leq Q_j - I_{\{j \in P\}} q_j$ and the inequality is fulfilled in these cases. Thus, there is an LP solution in the LB formulation which is not feasible for the EB formulation.

3. It remains to show that these LP solutions can be optimal.

With the travel costs in [Table C.11](#) the solution $0 \rightarrow 1^+ \rightarrow 2^+ \rightarrow 2^- \rightarrow 1^- \rightarrow 0$ has an objective value of 23 and the solution $0 \rightarrow 1^+ \rightarrow 1^- \rightarrow 2^+ \rightarrow 2^- \rightarrow 0$ an objective value of 38. Moreover, swapping 1^+ and 2^+ or 2^- and 1^- in the first solution or swapping the indices of customers 1 and 2 leads to the same objective values due to symmetry. Therefore, the LB model will use as much as possible of the first solution and its symmetric ones (such that capacity constraints stay fulfilled) while the EB model uses the second solution. Thus, $Z_{LAE}^{rel} > Z_{LB}^{rel}$ holds.

Table E.12

Results of LAEB model + preprocessing using a single thread for the original a-X and b-X instances.

Inst.	Obj.	Gap	CPU	CPU Graph	Inst.	Obj.	Gap	CPU	CPU Graph
a2-16-X	278.2		0.06		b2-16-X	244.3		0.39	
a2-20-X	330.7		0.02		b2-20-X	280.2		0.08	
a2-24-X	389.1		0.09		b2-24-X	372.6		0.56	
a3-18-X	272.7		0.07		b3-18-X	251.9		0.90	
a3-24-X	289.6		0.45		b3-24-X	321.2		0.85	
a3-30-X	452.8		0.18		b3-30-X	433.3		1.50	
a3-36-X	501.0		0.20		b3-36-X	486.0		0.85	
a4-16-X	235.2		0.40		b4-16-X	233.2		7.26	
a4-24-X	359.4		0.12		b4-24-X	293.0		8.11	
a4-32-X	447.3		3.55		b4-32-X	410.2		2.76	
a4-40-X	509.0		9.76		b4-40-X	478.7		2.59	
a4-48-X	619.0	49.86	0.63		b4-48-X	543.2		1052.93	1.91
a5-40-X	464.0		16.04		b5-40-X	494.8		442.79	
a5-50-X	621.9		204.39		b5-50-X	606.8	1.3%	2h	1.12
a5-60-X	745.4		39.10		b5-60-X	704.5		6428.60	2.54
a6-48-X	572.5		539.54	1.20	b6-48-X	577.8		6385.86	
a6-60-X	757.9		2205.76	4.93	b6-60-X	703.9	2.1%	2h	1.56
a6-72-X	868.4	1.6%	2h	1.36	b6-72-X	808.9	4.0%	2h	8.18
a7-56-X	663.5		354.86	0.95	b7-56-X	612.6	2.3%	2h	278.01
a7-70-X	815.3		2556.53	6.42	b7-70-X	741.4	2.0%	2h	5.26
a7-84-X	952.4	1.6%	2h	5.76	b7-84-X	N/A	N/A	2h	19.26
a8-64-X	701.2		3828.15	8.86	b8-64-X	N/A	N/A	2h	50.92
a8-80-X	882.7	2.2%	2h	8.98	b8-80-X	N/A	N/A	2h	6.24
a8-96-X	N/A	N/A	2h	28.56	b8-96-X	N/A	N/A	2h	225.09
Total			38609					79136	

Appendix D. Implementation of infeasible path constraints

For the implementation of inequalities [\(11\)](#) and [\(12\)](#) we first compute B_{vw}^{LB} and B_{vw}^{UB} for all $(v, w) \in A$. To implement inequalities [\(11\)](#), for all events v , we loop over all successors u (in decreasing order of B_{vu}^{UB}), determine the first predecessor w of v with $B_{vu}^{UB} < B_{vw}^{LB}$, and add all valid inequalities of type [\(11\)](#), i.e., for each v we add the inequality with variables $x_{(v,u')}$ such that $B_{vu'}^{UB} \leq B_{vu}^{UB}$ and variables $x_{(w',v)}$ such that $B_{w'v}^{LB} \geq B_{vw}^{LB}$.

Note that there is a dominance if for two successors \bar{u}^1 and \bar{u}^2 of the decreasing ordered list with $B_{v\bar{u}^1}^{UB} \geq B_{v\bar{u}^2}^{UB}$ the set of arcs (w, v) with $B_{v\bar{u}^1}^{UB} < B_{wv}^{LB}$ and $B_{v\bar{u}^2}^{UB} < B_{wv}^{LB}$, respectively, are identical. Then, the valid inequality [\(11\)](#) for \bar{u}^2 is equal or weaker to the one for \bar{u}^1 and therefore not added.

For the implementation of [\(12\)](#) we consider the case $\bar{m} = 4$. For all arcs $(u^2, u^3) \in A \setminus (\delta^{\text{in}}((0, \dots, 0)) \cup \delta^{\text{out}}((0, \dots, 0)))$, we loop over all successors u^4 of u^3 (in decreasing order of $B_{u^3u^4}^{UB}$) and determine the first predecessor u^1 (in increasing order of $B_{u^1u^2}^{LB}$) of u^2 for which the path $u^1 \rightarrow u^2 \rightarrow u^3 \rightarrow u^4$ is infeasible. We can apply the same dominance criterion like in the implementation of [\(11\)](#) here. Again, we add all variables $x_{(u^3,w)}$ with $B_{u^3w}^{UB} \leq B_{u^3u^4}^{UB}$, $w \neq u^4$, and all variables $x_{(v,u^2)}$ with $B_{vu^2}^{LB} \geq B_{u^1u^2}^{LB}$, $v \neq u^1$, to the left side of the inequality. We loop over all requests $i \in R$ and check whether events $u_{i^+}^2$ and $u_{i^-}^3$ and arcs $(u^2, u_{i^+}^2)$ and $(u_{i^-}^3, u^3)$ exist. If so, we add $\frac{1}{2} \cdot x_{(u^2, u_{i^+}^2)}$ and $\frac{1}{2} \cdot x_{(u_{i^-}^3, u^3)}$ to the inequality.

Appendix E. Computational results for original a-X and b-X instances

[Tables E.12](#) and [E.13](#).

Appendix F. Computational results for EB model on 12 threads

See [Table F.14](#).

Table E.13

Results of LAEB model + preprocessing using 12 threads for the original a-X and b-X instances.

Inst.	Obj.	Gap	CPU	CPU Graph	Inst.	Obj.	Gap	CPU	CPU Graph
a2-16-X	278.2	0.07			b2-16-X	244.3	0.23		
a2-20-X	330.7	0.03			b2-20-X	280.2	0.10		
a2-24-X	389.1	0.11			b2-24-X	372.6	0.22		
a3-18-X	272.7	0.07			b3-18-X	251.9	0.92		
a3-24-X	289.6	0.43			b3-24-X	321.2	0.81		
a3-30-X	452.8	0.13			b3-30-X	433.3	0.94		
a3-36-X	501.0	0.18			b3-36-X	486.0	0.80		
a4-16-X	235.2	0.22			b4-16-X	233.2	0.89		
a4-24-X	359.4	0.12			b4-24-X	293.0	3.59		
a4-32-X	447.3	0.94			b4-32-X	410.2	1.27		
a4-40-X	509.0	2.72			b4-40-X	478.7	1.90		
a4-48-X	619.0	18.75	0.62		b4-48-X	543.2	98.22	2.24	
a5-40-X	464.0	2.39			b5-40-X	494.8		111.12	
a5-50-X	621.9	29.85			b5-50-X	606.8	1.3%	2h	1.47
a5-60-X	745.4	21.05			b5-60-X	704.5		5214.38	3.12
a6-48-X	572.5	199.68	1.18		b6-48-X	578.2	1.1%	2h	
a6-60-X	757.9	388.60	4.82		b6-60-X	703.2	0.2%	2h	2.14
a6-72-X	868.4	2.0%	2h		b6-72-X	790.0	0.5%	2h	10.44
a7-56-X	663.5	454.40	0.93		b7-56-X	603.9		7117.10	283.17
a7-70-X	815.3	1505.02	6.31		b7-70-X	743.7	2.1%	2h	6.60
a7-84-X	950.9	0.8%	2h		b7-84-X	N/A	N/A	2h	15.54
a8-64-X	701.2	2271.22	8.72		b8-64-X	648.6	4.5%	2h	54.67
a8-80-X	881.3	1.4%	2h		b8-80-X	843.0	5.5%	2h	8.30
a8-96-X	1119.7	3.4%	2h		b8-96-X	N/A	N/A	2h	234.03
Total		33696						77352	

Table F.14

EB model on 12 threads.

Inst.	Obj.	CPU	Inst.	Obj.	CPU	Inst.	Obj.	Gap	CPU	Inst.	Obj.	CPU
a2-16	294.3	0.10	b2-16	309.4	0.03	a2-16-X	278.2	0.10	b2-16-X	282.5	0.18	
a2-20	344.9	0.02	b2-20	332.7	0.01	a2-20-X	330.7	0.06	b2-20-X	323.6	0.05	
a2-24	431.1	0.05	b2-24	444.7	0.05	a2-24-X	389.1	0.13	b2-24-X	412.3	0.04	
a3-18	300.5	0.04	b3-18	301.6	0.04	a3-18-X	272.7	0.10	b3-18-X	290.4	0.08	
a3-24	344.8	0.10	b3-24	394.5	0.11	a3-24-X	289.6	0.44	b3-24-X	363.7	0.17	
a3-30	494.8	0.04	b3-30	531.4	0.06	a3-30-X	452.8	0.32	b3-30-X	504.3	0.19	
a3-36	583.2	0.11	b3-36	603.8	0.09	a3-36-X	501.0	0.32	b3-36-X	565.9	0.13	
a4-16	282.7	0.06	b4-16	296.9	0.02	a4-16-X	235.2	0.28	b4-16-X	289.9	0.04	
a4-24	375.0	0.04	b4-24	371.4	0.06	a4-24-X	359.4	0.22	b4-24-X	347.0	0.69	
a4-32	485.5	0.15	b4-32	494.9	0.04	a4-32-X	447.3	1.02	b4-32-X	491.0	0.10	
a4-40	557.7	0.28	b4-40	656.6	0.14	a4-40-X	509.0	7.25	b4-40-X	628.3	0.27	
a4-48	668.8	0.50	b4-48	673.8	0.94	a4-48-X	620.3	48.19	b4-48-X	627.4	2.05	
a5-40	498.4	0.23	b5-40	613.7	0.21	a5-40-X	464.0	3.18	b5-40-X	585.1	2.45	
a5-50	686.6	2.47	b5-50	761.4	0.45	a5-50-X	621.9	86.59	b5-50-X	708.8	0.89	
a5-60	808.3	1.03	b5-60	902.0	0.92	a5-60-X	745.4	38.18	b5-60-X	851.9	5.00	
a6-48	604.1	0.89	b6-48	714.8	0.26	a6-48-X	572.5	1119.86	b6-48-X	691.6	1.25	
a6-60	819.3	8.83	b6-60	860.0	0.43	a6-60-X	757.9	637.21	b6-60-X	841.6	1.33	
a6-72	916.1	13.87	b6-72	978.5	5.11	a6-72-X	871.3	1.2%	b6-72-X	930.3	49.64	
a7-56	724.0	1.24	b7-56	824.0	3.62	a7-56-X	663.5	361.20	b7-56-X	787.9	36.08	
a7-70	875.7	6.93	b7-70	912.6	2.42	a7-70-X	815.3	1828.44	b7-70-X	865.3	27.07	
a7-84	1033.3	5.89	b7-84	1203.3	2.72	a7-84-X	953.7	2.3%	b7-84-X	1141.2	1300.92	
a8-64	747.5	7.19	b8-64	839.9	2.16	a8-64-X	701.2	0.4%	b8-64-X	818.3	7.71	
a8-80	945.8	25.11	b8-80	1036.4	1.54	a8-80-X	N/A	N/A	b8-80-X	998.3	12.32	
a8-96	1229.7	461.04	b8-96	1185.6	24.26	a8-96-X	N/A	N/A	b8-96-X	1137.7	1525.10	
Total		536		46				40133		2974		

References

- Ascheuer, N., Fischetti, M., & Grötschel, M. (2000). A polyhedral study of the asymmetric traveling salesman problem with time windows. *Networks*, 36(2), 69–79.
- Camm, J. D., Raturi, A. S., & Tsubakitani, S. (1990). Cutting big M down to size. *Interfaces*, 20(5), 61–66.
- Codato, G., & Fischetti, M. (2006). Combinatorial Benders' cuts for mixed-integer linear programming. *Operations Research*, 54(4), 756–766.
- Cordeau, J.-F. (2006). A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, 54(3), 573–586.
- Foljanty, L. (2020). Mapping the global on-demand ridepooling market. <https://lukas-foljanty.medium.com/mapping-the-global-on-demand-ridepooling-market-f8318de1c030>. (Access September 2024).
- Gaul, D., Klamroth, K., & Stiglmayr, M. (2021). Solving the dynamic dial-a-ride problem using a rolling-horizon event-based graph. In *21st symposium on algorithmic approaches for transportation modelling, optimization, and systems*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
- Gaul, D., Klamroth, K., & Stiglmayr, M. (2022). Event-based MILP models for ridepooling applications. *European Journal of Operational Research*, 301(3), 1048–1063.
- Gschwind, T., & Irnich, S. (2015). Effective handling of dynamic time windows and its applications to solving the dial-a-ride problem. *Transportation Science*, 49(2), 335–354.
- Ho, S. C., Szeto, W. Y., Kuo, Y.-H., Leung, J. M., Petering, M., & Tou, T. W. (2018). A survey of dial-a-ride problems: Literature review and recent developments. *Transportation Research, Part B (Methodological)*, 111, 395–421.
- Letchford, A. N., & Salazar-González, J.-J. (2005). Projection results for vehicle routing. *Mathematical Programming*, 105(2–3), 251–274.
- Lysgaard, J. (2006). Reachability cuts for the vehicle routing problem with time windows. *European Journal of Operational Research*, 175(1), 210–223.
- Molenbruch, Y., Braekers, K., & Caris, A. (2017). Typology and literature review for dial-a-ride problems. *Annals of Operations Research*, 259(1), 295–325.
- Molenbruch, Y., Braekers, K., Caris, A., & Berghe, G. V. (2017). Multi-directional local search for a bi-objective dial-a-ride problem in patient transportation. *Computers & Operations Research*, 77, 58–71.
- Pfeiffer, C., & Schulz, A. (2022). An ALNS algorithm for the static dial-a-ride problem with ride and waiting time minimization. *OR Spectrum*, 44, 87–119.
- Rist, Y., & Forbes, M. A. (2021). A new formulation for the dial-a-ride problem. *Transportation Science*, 55(5), 1113–1135.
- Ropke, S., & Cordeau, J.-F. (2009). Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science*, 43(3), 267–286.
- Ropke, S., Cordeau, J.-F., & Laporte, G. (2007). Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks*, 49(4), 258–272.
- Schulz, A., & Pfeiffer, C. (2024). Using fixed paths to improve branch-and-cut algorithms for precedence-constrained routing problems. *European Journal of Operational Research*, 312(2), 456–472.