Load Factor Optimization for the Auto-Carrier Loading Problem (ACLP)

Christian Jäck
BMW Group, christian.jaeck@bmw.de,
Chair of Service Operations, Mercator School of Management, University of Duisburg-Essen

Jochen Gönsch
Chair of Service Operations, Mercator School of Management, University of Duisburg-Essen, jochen.goensch@uni-due.de,

Hans Dörmann Osuna
BMW Group, hans.doermann-osuna@bmw.de

The distribution of passenger vehicles is a complex task and a high cost factor for automotive original equipment manufacturers (OEMs). On the way from the production plant to the customer, vehicles travel long distances on different carriers such as ships, trains, and trucks. To save costs, OEMs and logistics service providers aim to maximize their loading capacities. Modern auto carriers are extremely flexible. Individual platforms can be rotated, extended, or combined to accommodate vehicles of different shapes and weights and to nest them in a way that makes the best use of the available space. In practice, finding feasible combinations is done with the help of simple heuristics or based on personal experience. In research, most papers that deal with auto carrier loading focus on route or cost optimization. Only a rough approximation of the loading sub-problem is considered.

In this paper, we present two different methodologies to approximate realistic load factors considering the flexibility of modern auto carriers and their height, length, and weight constraints. Based on our industry partner's process, the vehicle distribution follows a FIFO principle. For the first approach, we formulate the problem as a mixed integer quadratically constrained assignment problem. The second approach considers the problem as a two-dimensional nesting problem with irregular shapes. We perform computational experiments using real-world data from a large German automaker to validate and compare both models with each other and with an approximate model adapted from literature. The simulation results for the first approach show that on average for 9.37% of all auto carriers it is possible to load an additional vehicle compared to the current industry solution. This translates to 1.36% less total costs. The performance of the nesting approach is slightly worse, but as it turns out it is well suited to check load combinations for feasibility.

Key words: auto carrier loading; mixed integer linear programming; nesting problems; no-fit-polygon; real world data

1. Introduction

According to the annual report of the German association of the automotive industry, VDA (2020), more than 80 million passenger vehicles were sold worldwide in the year 2020. Despite constantly evolving sales channels and retail models, the physical distribution of finished vehicles has not changed significantly over the last decades. On their way to the end users, new vehicles are transported across multiple continents via trucks, trains, and ships. Due to the complexity, weight, and value of new passenger vehicles, their distribution is a large, but essentially non-value-adding cost factor for automotive companies. When choosing the transport carrier, car manufacturers try to opt for train or ship transportation because of economies of scale. Despite the higher costs, transportation by trucks, or so-called auto carriers, is often required because of their flexibility.

In research, the problem of distributing finished vehicles via truck is known as the auto carrier transportation problem (ATP), and was first considered by Agbegha, Ballou, and Mathur (1998). The ATP consists of two sub-problems:

- (1) The vehicle routing problem (VRP) in the scope of the ATP is the problem of how to efficiently route the auto carrier along a road network to deliver vehicles to one or multiple dealers.
- (2) The auto carrier loading problem (ACLP) is a combinatorial problem that searches for a feasible set (and arrangement) of vehicles for a specific auto carrier.

Problem (1) is studied extensively in various research fields that involve traveling salesman or other VRP problems. Most of the literature in the domain of auto carrier transportation focuses on VRPs with route, scheduling, or cost optimization from the view of logistics service providers. Usually, the goal is to distribute a large number of vehicles from a single depot to multiple dealers at different locations with multiple auto carriers at minimum costs or along the shortest route. Regarding the loading sub-problem (2), the vehicles are to be allocated on the truck in such a way that the load is feasible according to the restrictions of the auto carrier. Ideally, the vehicles are sorted last-in-first-out according to the order in which the dealers are supplied. This minimizes reloading along the route.

The ACLP is not trivial and is dissimilar to loading and packing problems from other research fields. Many researchers simplify the ACLP or consider special cases such as auto carriers with static loading planes which makes them similar to Ro-Ro ships or auto trains. In reality, auto carriers are highly flexible in terms of loading capabilities. They usually have multiple loading planes and platforms that can be moved horizontally and vertically. Some platforms can even be rotated or physically extended. Car manufacturers aim to minimize costs by achieving high load factors. The load factor is the maximum number of vehicles of one or of multiple different types that fit on a single auto carrier. It is no easy task to determine optimal load factors because of the flexibility of the loading planes and platforms, the different types of trucks, and the varying geometries and weights of the cars. In practice, automotive OEMs and logistics service providers usually rely on experience or simple heuristics. A common approach is to cluster the vehicles in multiple categories according to their height and length. Each category is assigned a coefficient that indicates how many vehicles of this category will fit on a single truck. A combination of vehicles of one or multiple categories is assumed feasible, as long as the sum of their coefficients is below a certain threshold. Within these categories the actual vehicle dimensions or individual vehicle properties are not considered. This simple knapsack approach often results in combinations that do not fully utilize the available capacity of the auto carrier or in combinations that are infeasible. In this paper, we focus on the ACLP from the perspective of an automotive OEM. In this paper we develop two different methods for approximating realistic load factors and compare them with contributions of existing literature, the current industry solution and historic data.

1.1. Problem Statement

The decision maker is an automotive OEM (BMW Group) with a yearly production of about 2.5 million vehicles who aims to minimize distribution costs by utilizing the loading capabilities of the auto carrier trucks. Because of the great variances in vehicle dimensions, and weights and due to the complexity of modern auto carriers, the current heuristic is not sufficient for calculating optimal feasible load factors. The decision maker has detailed information about the cars as well as two different types of auto carriers that are common among logistics service providers in the European market. Because the storage capacity (parking space) in the factories is quite limited, the vehicles should be shipped as fast as possible. This means that as soon as there are enough vehicles available to build a single load for a specific destination, the OEM sends the transport order to a logistics service provider. Because of that it is not possible to pool a larger number of vehicles to build multiple loads simultaneously. A single auto carrier can load up to ten vehicles.

1.2. Contributions

We make the following contributions to the existing literature: (1) We formulate the ACLP as an assignment problem and model it using a mixed integer, quadratically constrained linear program (MILP). We recognize that auto carriers are highly flexible. Vehicles can be stacked on top of each other with the help of rotational platforms. We consider the resulting height and length effects within the model. This model also provides a feasible vehicle to slot assignment, thus, a guide on how to allocate the vehicles on the auto carrier. (2) We propose a new approach for solving the loading problem utilizing a geometric algorithm. For that, we adapt a classic nesting algorithm for irregular shapes based on the No-Fit Polygon (NFP) that uses a limited look-ahead greedy heuristic to generate realistic loading patterns. (3) We benchmark both of our methods with one of the more sophisticated approaches from literature (Dell'Amico, Falavigna, and Iori (2014)), the current practice in industry, and historic real-world data sets.

1.3. Outline

The remainder of this paper is structured as follows. Section 2 provides an overview of relevant literature. Section 3 explains the considered problem and its parameters. In Sections 4 and 5, the MILP and nesting approaches are presented in detail. Section 6 evaluates the approaches in comparison to benchmarks from literature and industry using real-world data. Section 7 provides a discussion and managerial insights before Section 8 concludes.

2. Literature Review

A general overview of finished vehicle distribution and related problems that largely focuses on the ATP has been given recently by Sun, Kirtonia, and Chen (2021). The ATP consists of two sub problems. The auto carrier loading problem (ACLP) revolves around assigning a given set of vehicles to auto carriers. The routing problem describes the decision-making process of finding the optimal sequence and route on which to deliver the vehicles to multiple dealers. Researchers either focus on one of the problems or consider the combination of both problems in different degrees of detail. The most common optimization objectives for studies that focus on routing is to minimize costs by reducing total travel distance, time, or the number of required auto carriers for a given set of vehicles (e.g. Tadei, Perboli, and Della Croce (2002), Dell'Amico, Falavigna, and Iori (2014) or Chen and Wang (2020)). Since in this paper we are only concerned with the ACLP, in Section 2.1 we concentrate on corresponding literature. Section 2.2 provides a brief introduction to two-dimensional nesting problems for irregular shapes and relevant literature.

2.1. The auto carrier loading problem

The ATP was first addressed by Agbegha (1992) and later published as a research paper by Agbegha, Ballou, and Mathur (1998). The authors focus exclusively on loading decisions and propose a so-called loading network with nodes representing possible placement slots on the auto carrier. These nodes are connected by arcs which define a certain unloading preference depending on the type of the auto carrier (A vehicle can not be unloaded if it is blocked by a vehicle with lower precedence). The optimization objective was to minimize the number of reloads given a specific sequence of dealers. Thus, vehicles that are to be delivered to the first dealers should be on slots with low precedence. The only loading constraints the authors include are single car constraints that restrict certain slots for certain types of cars. They formulate the problem as an assignment problem and solve it with a heuristic branch-and-bound algorithm. Chen (2016) expanded this precedence network approach by allowing vehicles to be reloaded to any slot after unloading. He also introduced pairwise limitations to restrict certain slots due to larger types of vehicles occupying two neighboring slots.

Literature that deals with the combination of loading and routing tends to not include vehicle-to-slot assignments. Instead, the loading problem is formulated as a simple knapsack or bin packing problem. Numerous authors of recent studies such as Bonassa, Cunha, and Isler (2019) or Chen and Wang (2020) group vehicles of different dimensions in a fixed number of classes. Each class is assigned a specific size coefficient. As long as the sum of the coefficients of a load is below a certain threshold, the load is considered feasible. This is how the loading problem is commonly approached in practice. Tadei, Perboli, and Della Croce (2002) formulated the ATP as a mixed integer linear program. They clustered the different target destinations into regions. The algorithm assigns vehicles to auto carriers and auto carriers to regions. The capacity of an auto carrier is defined by the combined available length of all loading planes. Vehicles are grouped into multiple classes based on their size. Each class is assigned a certain coefficient, similar to the approaches of Bonassa, Cunha, and Isler (2019) and Chen and Wang (2020). This coefficient is multiplied by the actual vehicle length to approximate the equivalent length of the vehicle on the auto carrier accounting for the flexibility of the loading planes. The algorithm does not specify individual vehicle-to-slot assignments but checks whether the equivalent length of the set of vehicles is less than the available length. Dell'Amico, Falavigna, and Iori (2014) propose an iterative local search algorithm for the routing, combined with an advanced feasibility check for the loading. They adapt the formulation for the ACLP of Tadei, Perboli, and Della Croce (2002) by considering the loading planes individually. They also take the height and length effects of different vehicle types into consideration (e.g. large vehicles on a lower loading plane reduce the available space on the loading plane above). A LIFO policy is imposed to avoid reloading when possible. A common feature among the models that consider vehicle dimensions is the utilization of coefficients to approximately model geometries or to account for the flexibility of the auto carrier. In reality, obtaining such coefficients is a laborious and impractical task. It requires extensive analysis of historic data and constant maintenance due to the continuously changing dimensions of the vehicles and changing properties of the auto carriers. None of theses approaches includes detailed weight restrictions. This can lead to infeasible loads in practice.

So far, very few authors have explored more advanced or alternative methods for solving the ACLP. In an unpublished working paper Venkatachalam and Sundar (2016) proposed a branch-and-price algorithm and claimed to solve the loading problem exactly. Indeed, their approach is the most sophisticated so far, considering numerous height, weight and length constraints, length and height effects of angling certain platforms and precise vehicle to slot assignments. However, they only consider a type of auto carrier that does not allow to angle inner platform structures. In their model it is only possible to reduce the effective height of vehicle pairs on vertical platform pairs by adjusting the platforms' angles. Vehicles are reduced to a single value for length, height and weight respectively. Because of that, the calculation of length and height effects caused by angling the platforms is not exact. Two research groups have tried to consider the actual intricate vehicle shapes. Liu, Smith, and Qian (2016) converted the shapes of the vehicles into polygons to calculate length and height effects when vehicles are angled. However, they considered the loading planes of the auto carrier separately and allowed to angle every vehicle on certain loading planes. Hu et al. (2015) combined a geometric bin-packing algorithm with a learning procedure to discover loading patterns from historic data. They present the only geometric bin packing approach so far within the ATP research area. They did not develop a new approach but implemented one of the heuristic algorithms proposed by Valle et al. (2012). They neither provide information about which specific heuristic they used nor if they adapted the algorithm.

2.2. Nesting Problems

According to the typologies proposed by Dyckhoff (1990) and Wäscher, Haußner, and Schumann (2007), nesting problems belong to the family of cutting and packing problems. They appear in a wide variety of industries. Cutting and packing problems occur when we try to cut (place) a predefined set of objects from (into) larger objects while respecting specific constraints. Examples of industries where cutting problems arise are garment, sheet metal cutting, and furniture industries. Packing problems appear for example in transportation or warehousing, where items are packed on shelves or truck beds. The problem variants involving irregular shapes are often referred to as nesting problems. Bennell and Oliveira (2008) provide a general overview of nesting problems and their applications. In the domain of transportation, research is mostly limited to packing regular shapes such as two-dimensional rectangles or three-dimensional boxes. Numerous papers deal with capacitated vehicle routing problems or vehicle routing problems with loading constraints for rectangle items. For surveys of integrated routing and packing problems, we refer the reader to Iori and Martello (2010) and Guastaroba, Mor, and Speranza (2022). To create an understanding for the complexity of the problem type we refer the reader to a notable publication from Iori, Salazar González, and Vigo (2007). They presented an exact approach for a vehicle routing problem with two-dimensional loading constraints for rectangle items. They partition customers into routes of minimum cost and take the weight and space constraints of the carrier into account. In the case of rectangular items, it is easy to divide a stock sheet or a rectangular container into a limited discrete set of possible locations. Irregular shapes, without orthogonal edges, with concavities or holes, pose a much greater challenge. The possible positions on the stock sheet or within the container are continuous, so there are infinitely many possible locations. Due to the increased complexity of the solution space, calculating good solutions is computationally intensive. The scientific work on cutting and packing problems with irregular shapes focuses primarily on the development of heuristic algorithms to solve large strip cutting or packing problems. Guo et al. (2022) provide a review for algorithms and strategies that have been proposed for the problems in recent years. A strip is a continuous sheet of material with a fixed height. In case we deal with a strip cutting problem, the objective is to cut out as many items as possible and at the same time minimize wasted material. The same problem can be applied to packing items into containers with fixed height and length. Critical for the performance and the results of a nesting algorithm is the so called geometric tool. Providing a comprehensive overview of tools for representing the geometric problems and algorithms for solving irregular packing problems is not within the scope of this paper. For that we refer the reader to Bennell and Oliveira (2008) or Leao et al. (2020). The geometric tool chosen to deal with the problem affects solution precision, computing time and the complexity of implementation. According to Bennell and Oliveira (2008), frequently used geometric tools are for example the raster point method, direct trigonometry, phi-functions and No-Fit Polygon (NFP). Raster point and NFP are popular tools for geometric packing operations with irregular shapes. In recent publications, researchers have started combining both tools to build more efficient hybrid heuristics. Mundim et al. (2018) proposed the No-Fit Raster (NFR) approach based on the dotted-board model by Toledo et al. (2013). They transform polygons into binary matrices and pre-calculate NFPs and inner-fit polygons (IFPs) by generating a discrete list of feasible points within a container. To solve the optimization problem, for every new item, they iterate through this list to find the best position for the item within the container according to a variety of placement rules. Souza Queiroz and Andretta (2020) improved the efficiency of the NFR approach by combining it with a biased random key genetic algorithm and a variable neighborhood search. Al Theeb, Hayajneh, and Jaradat (2021) built a mathematical programming model based on the dotted-board model and the NFR to solve the problem with a commercial solver. In this paper, we use the NFP because it combines the efficiency of the raster point method with the high accuracy of the direct trigonometry approach. Burke et al. (2007) provides a comprehensive algorithm for the creation of NFPs.

3. Problem description

The following section describes the investigated real-world problem in detail. First, we define our problem, derive assumptions for modeling the problem and state the objective. Next, we explain the features of modern European auto carriers and passenger vehicles, as well as the parameters and constraints of our problem.

3.1. Problem definition, assumptions and objective

We consider a vehicle distribution problem of a single automotive OEM (other players in the industry face similar restrictions). Automotive OEMs often need to transport large quantities of passenger vehicles from one hub to another along fixed routes (i.e., from a specific plant to a storage facility). When there is no rail network available, auto carriers are used. Logistics companies carry out the transports. Each transport is requested individually by the OEM. Remember, it is not possible to accurately predict at what time which cars will be available for which location. Also the storage capacity (parking space) is limited. Therefore, the process follows a FIFO principle, which is visualized in the flow chart in Figure 1. Vehicles are provided by a source (plant, previous logistics hub, or maintenance workshop) and parked in a large distribution storage facility. In this facility, vehicles are parked in rows. Each row is reserved for vehicles destined for one specific location. Once there are enough vehicles to build a full load in one row (green vehicles in the parking area in Figure 1), the logistics provider is notified, and a transport order is sent. The maximum number of cars in one load varies depending on the size and weight of the vehicles. Once the transport order is sent, the row is closed. If a new vehicle for the same destination arrives, a new row is opened up. The logistics service provider sends an auto carrier to pick up the vehicles. The row is now empty and a new load can be formed. Even if there are multiple rows with vehicles for the same location, the truck driver is not allowed to take vehicles from another row.

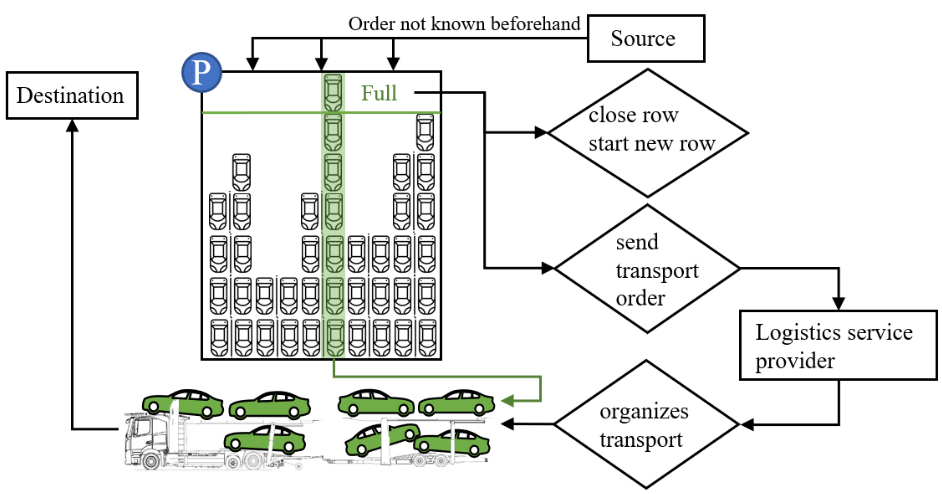

Figure 1 Flowchart of the current real world process

Based on these real-world constraints we make the following assumptions for our model formulations:

- Because the storage area is limited at all locations, vehicles are to be shipped as fast as possible. As soon as there are enough vehicles available to form a load for a specific destination, the transport order is shipped (FIFO).

- When the vehicles of one row are loaded on the auto carrier the truck driver can decide where to position which vehicle on his auto carrier. For every destination, we always only consider a single auto carrier, that has to be filled before the next auto carrier is considered.
- Since it is not relevant for the current problem of our industry partner, for our computational experiments we do not consider split deliveries (dealer deliveries). All vehicles are shuttled in batches on fixed routes from one location to another. Locations can be production plants, compounds, or transportation hubs such as ports.
- The decision maker has complete information about the shape, dimension, and weight of the vehicles and about the dimensions and restrictions of the auto carriers.

Our objective is to find a feasible arrangement for a given set of vehicles that respects all loading constraints to maximize the loading capacities of a given auto carrier in order to reduce outbound logistics cost.

3.2. Features of European auto carriers

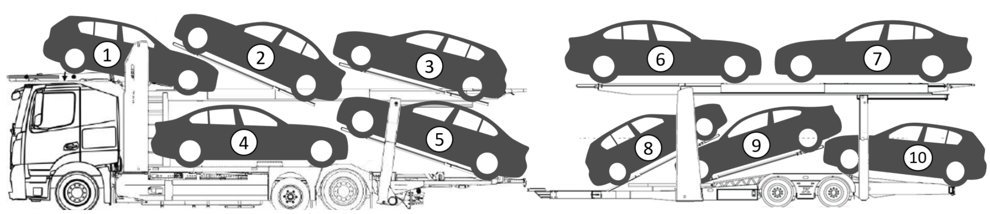

Figure 2 **Example for an auto carrier**

European auto carriers usually consist of a truck and a trailer, each holding a loading structure with two parallel horizontal loading planes. Every loading plane can hold a specific number of vehicles depending on its maximum length. Loading planes can be lifted vertically and certain loading planes are horizontally extendable. Loading planes can be static levels or they can consist of multiple independent platforms. Some of these platforms can be moved horizontally or can even be angled with the help of hydraulics. Figure 2 shows a typical auto carrier with four loading planes and ten possible vehicle positions. We refer to these positions as platforms. Some of these platforms (e.g. 2, 3, 5, 8 and 9) can be rotated (angled) in order to stack vehicles. When a platform is rotated, the effective length of the vehicle on that platform reduces while the effective height increases. These effects depend on the rotation angle of the platform. The rotation angle of the platform in turn depends on the height of the vehicle on the neighbouring platform. Vehicles can stand in a forward or backward position and shift forward or backward along the platform.

Several constraints limit the capacity of auto carriers. Governmental regulations only allow a certain length, height, and total weight for auto carriers (i.e. 20.75m maximum length, 4m maximum height and 40t maximum weight in Germany). In addition, every auto carrier has certain constraints regarding the weight of the vehicles. There are weight limits for the truck and trailer and for every level of the auto carrier. On top of that, every single platform also has a weight constraint depending on its position. If it is angled, it can hold less weight due to unfavorable weight distribution. Because of that, the weight constraints play a major role in the final allocation of the vehicles. Since some passenger vehicles are very heavy (up to 3t), they have to be positioned on two platforms to distribute their weight. The weight limitations are becoming more relevant as more electric vehicles are built. Electric vehicles are heavier than similar vehicles with combustion engines.

3.3. Parameters

Every auto carrier consists of multiple loading planes and every loading plane has a certain number of platforms that can hold a vehicle. Typical auto carriers have a set of up to ten platforms P, indexed by p and can load between four and ten vehicles. To check the feasibility of a certain set of currently available vehicles K, indexed by k, we must consider individual platforms, but also the interrelation of multiple platforms. For example, if we want to check the total weight on the upper loading plane of the truck, we must combine the weight of the vehicles on platforms (1), (2) and (3). To express these relationships, we define the following sets (where $\mathcal{P}(P)$ is the power set of P):

P_L: Set $L \in P_L \subseteq \mathcal{P}(P)$ represents a loading plane with platforms that have a common limitation on length (e.g. $P_L = \{\{1, 2, 3\}, \{4, 5\}, \{6, 7\}, \{8, 9, 10\}\}$ for the auto carrier in Figure 1).

For each set L there is a set of horizontally neighbouring platforms $h(L)$ (e.g., $h(\{1,2,3\}) = \{6,7\}$ and $h(\{4,5\}) = \{8,9,10\}$ for the auto carrier in Figure 1)

P_H: Set $H \in P_H \subseteq \mathcal{P}(P)$ represents a set of platforms that have a common limitation on height (e.g. $P_H = \{\{1, 4\}, \{2, 4\}, \{2, 5\}, \{3, 4\}, \{3, 5\}, \{6, 8\}, \{6, 9\}, \{7, 9\}, \{7, 10\}\}$ for the auto carrier in Figure 1).

P_A: Set of platforms $P_A \subseteq P$ that can be angled (e.g. 2, 3, 5, 8 and 9 for the auto carrier in Figure 1).

For each platform $p \in P_A$ there is also a platform $v(p)$ that holds the vehicle nested below platform p when p is angled (e.g. $v(2) = 1, v(3) = 2, v(5) = 4, v(8) = 9, v(9) = 10$ for the auto carrier in Figure 1)

P_{SP}: Set of sets of platforms $P_{SP} \subseteq \mathcal{P}(P)$ that can be combined to a split platform $Q \in P_{SP}$ to hold a single heavier vehicle (e.g. $P_{SP} = \{\{1, 2\}, \{4, 5\}, \{6, 7\}, \{8, 9\}, \{9, 10\}\}$ for the auto carrier in Figure 1).

P_T: Platforms $T \in P_T \subseteq \mathcal{P}(P)$ are either part of the truck or the trailer, with $\cup_{T \in P_T} = P$ (e.g. $P_T = \{\{1, 2, 3, 4, 5\}, \{6, 7, 8, 9, 10\}\}$ for the auto carrier in Figure 1).

Every vehicle $k \in K$ has a certain length l_k, height h_k, weight w_k and is of a certain type c_k. For every set of platforms, or collection of sets, there are certain restrictions in order to comply with legal requirements and the technical limitations of the auto carrier:

- L_L: Length of loading plane $L \in P_L$
- E_L^{max}: Maximum extension of loading plane $L \in P_L$
- $E_{L,\ h(L)}^{max}$: Maximum total extension of neighbouring loading planes $L \in P_L$ and $h(L)$ (maximum extension before the loading planes touch)
- H_H^{max}: Maximum allowed height of the vehicles on vertical platform pairs $H \in P_H$
- W^{max}: Maximum allowed total weight of the payload
- wp_p^{max}: Maximum allowed weight on platform $p \in P$
- wa_p^{max}: Maximum allowed weight on platform $p \in P_A$ if the platform is angled
- wc_Q^{max}: Maximum allowed weight if platforms p are combined to build a larger platform $Q \in P_{SP}$
- wt_T^{max}: Maximum allowed weight on truck and trailer $T \in P_T$
- wl_L^{max}: Maximum allowed weight on loading plane $L \in P_L$

4. MILP model

In this section, we present a mixed integer quadratically constrained assignment problem formulation for the ACLP ((1) to (19)). The MILP formulation finds a feasible load configuration for a given number of vehicles and a specific type of auto carrier. Similar to the formulation of Venkatachalam and Sundar (2016), we generate a precise vehicle to slot assignment and consider height, length, and weight constraints. We expand the concept by considering flexible loading planes and flexible inner platforms. The concept of split-platforms which inspired constraints (4) and (8) was introduced by Venkatachalam and Sundar (2016). Constraints (6) and (7) regarding continuous level extensions are taken over from the model formulation of Dell'Amico, Falavigna, and Iori (2014). Besides the standard assignment constraints (2) and (3) the remaining constraints are original.

4.1. Decision variables.

We now define the decision variables used in the MILP. Let x_{kp} be a binary decision variable that is equal to 1, if vehicle $k \in K$ is positioned on platform $p \in P$ and 0 otherwise.

$$x_{kp} = \begin{cases} 1 & \text{if vehicle k is positioned on platform p} \\ 0 & \text{otherwise} \end{cases}$$

Let a_p be a binary decision variable that is equal to 1, if platform $p \in P_A$ is angled and 0 otherwise:

$$a_p = \begin{cases} 1 & \text{if platform } p \in P_A \text{ is angled} \\ 0 & \text{otherwise} \end{cases}$$

Let sp_Q be a binary decision variable that is equal to 1, if platforms $Q \in P_{SP}$ are combined to a split-platform Q, to hold a heavier vehicle:

$$sp_Q = \begin{cases} 1 & \text{if } Q \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

Let d_p be a binary decision variable that is equal to 1, if the vehicle on platform p is oriented in a forward direction (in respect to the orientation of the truck) and 0, if the vehicle is oriented in the opposite direction.

$$d_p = \begin{cases} 1 & \text{if the vehicle on platform } p \text{ is oriented forwards} \\ 0 & \text{otherwise} \end{cases}$$

Finally, we define a continuous variable e_L that defines the extension of the loading plane $L \in P_L$. The maximum extension is limited by parameter E_L^{max} for every loading plane $L \in P_L$, i.e. $e_L \leq E_L^{max}$.

4.2. Objective function.

Our goal is to find a feasible combination for loading the set K of currently available vehicles k on the platforms $p \in P$ of the auto carrier. Thus, we maximize over our decision variable x_{kp} to determine the best possible load factor. Based on our problem definition we repeatedly solve the problem for every new vehicle that arrives. If the new vehicle does not fit, which is the case if our objective value is smaller than the cardinality of K, we terminate this process. The load is completed and the new vehicle that did not fit becomes the first vehicle in the next set for the following load.

$$max \sum_{p \in P} \sum_{k \in K} x_{kp} \tag{1}$$

4.3. Assignment constraints.

$$\sum_{k \in K} x_{kp} \leq 1 \qquad \forall p \in P \tag{2}$$

$$\sum_{p \in P} x_{kp} \leq 1 \qquad \forall k \in K \tag{3}$$

$$\sum_{k \in K} \sum_{p \in Q} x_{kp} \leq |Q| \left(1 - sp_Q\right) + sp_Q \qquad \forall Q \in P_{SP} \tag{4}$$

$$2 \cdot a_p \leq \sum_{k \in K} x_{kp} + \sum_{k \in K} x_{kv(p)} \qquad \forall p \in P_A \tag{5}$$

Constraints (2) ensure that every platform can only hold a single vehicle. Constraints (3) ensure that every vehicle can only be positioned on one platform. Constraints (4) restrict using a platform p when it is occupied by a vehicle on a neighbouring platform (split platform). The respective platform $p \in Q \in P_{SP}$ can hold heavier vehicles if combined with another platform ($sp_Q = 1$). Constraints (5) ensure platforms are only allowed to be angled when required (only if there are vehicles on p and $v(p)$).

4.4. Length constraints.

If a platform $p \in P_A \subseteq P$ is angled, the effective length of the vehicle on that platform is reduced by a specific length reduction factor. The size of the length reduction factor of a vehicle k on a platform p depends on the orientation, class, and original size of the vehicles on p and on $v(p)$, respectively. The total available length of a specific loading plane $L \in P_L$ depends on the length of the loading plane L_L and the value of the additional extension e_L.

Because of the irregular shapes of the vehicles and because the loading planes are not perfectly horizontal levels, there is no linear relationship between the angle and the length and height effects. Because of that, we developed a procedure that approximates discrete angles and height- and length effects. Since the calculation is quite complex, here we only provide a short summary of the procedure. The calculations are detailed comprehensively in Appendix A. Figure 3 shows an outline of the procedure for two facing sedans k_1 and k_2. First, we calculate the required lifting angle α. This angle depends on the height of the bonnet ($H_{B,k2}$) of vehicle k_2 on the neighboring platform. We assume that vehicle k_1 is angled so that its front wheel-axis is directly above the front wheel-axis of the neighboring vehicle k_2. Next, based on the lifting angle α, the original length of the angled vehicle L_{k1} and the height of its bonnet $H_{B,k1}$ we approximate the height gain for the angled vehicle. Since we always angle the vehicle so that its wheel-axis aligns with the wheel-axis of the neighboring vehicle, its effective length can be approximated to the length of its wheelbase.

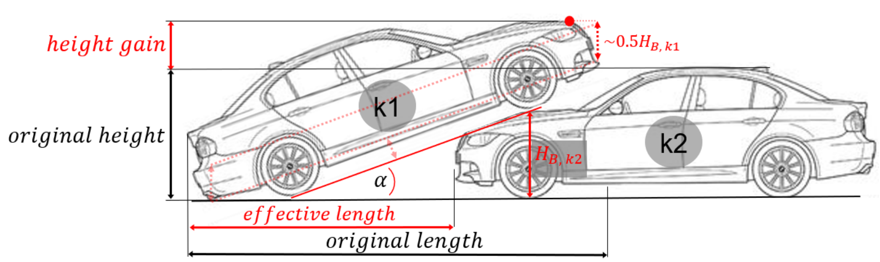

Figure 3 Calculation of the effective length and height if a vehicle is angled

$\lambda^{c_p,c_{v(p)}}_{d_p,d_{v(p)}}$: Length reduction factor depending on the class (c_p, $c_{v(p)}$) of the vehicles on p and neighbouring $v(p)$ and their orientation d_p, $d_{v(p)}$. The calculation of the length reduction factor is detailed in Appendix A.

l_{kp}: Effective length of vehicle k on platform $p \in P_A$ if the platform is angled

$$0 \leq e_L \leq E_L^{max} \qquad \forall L \in P_L \qquad (6)$$

$$e_L + e_{h(L)} \leq E_{L,h(L)}^{max} \qquad \forall L \in P_L : max(L) < max(h(L)) \qquad (7)$$

$$\sum_{p \in Q \cap P_A} a_p \leq |Q| \left(1 - sp_Q\right) \qquad \forall Q \in P_{SP} \qquad (8)$$

$$\sum_{k\in K}\sum_{p\in L\cap P_A} l_{kp}x_{kp}+\sum_{k\in K}\sum_{p\in L\setminus P_A} l_k x_{kp}\le L_L+e_L \qquad \forall L\in P_L \qquad (9)$$

$$l_{kp}=l_k-l_k a_p\lambda^{c_p,c_{v(p)}}_{d_p,d_{v(p)}} \qquad \forall p\in P_A \qquad (10)$$

Constraints (6) limit the maximum extension of individual loading planes and constraints (7) limit the maximum extension of neighboring loading planes. Constraints (8) ensure that if multiple platforms are combined to hold a heavier vehicle, none of these individual platforms can be angled. Constraints (9) restrict the available length on loading plane $L\in P_L$. If a platform is angled, the effective length of the vehicle on that platform is reduced by a certain amount that depends on the length reduction factor. The reduction amount is calculated in equation (10).

Note that for ease of understanding we write the problem as a mixed integer quadratically constrained problem. In the computational study, this is also our input to the solver, which obviously efficiently handles our problem instances. The only quadratic constraints are Conditions (9) and (11), which are straightforward to linearize with the help of auxiliary binary variables.

4.5. Height constraints.

The effective height of an angled vehicle is calculated in the same way the effective length is calculated. Instead of reducing the effective length due to the gained space for the vehicle nested beneath the angled platform on $v(p)$, we must account for the height increase of the vehicle on platform $p\in P_A\subseteq P$.

$\delta^{c_p,c_{v(p)}}_{d_p,d_{v(p)}}$: Height increase factor depending on the class, size and orientation of the vehicles on p and $v(p)$. The calculation for the height increase factor is detailed in Appendix A.

h_{kp}: Effective height of vehicle k on platform $p\in P_A$ if the platform is angled

$$\sum_{k\in K}\sum_{p\in H\cap P_A} h_{kp}x_{kp}+\sum_{k\in K}\sum_{p\in H\setminus P_A} h_k x_{kp}\le H^{max}_H \qquad \forall H\in P_H \qquad (11)$$

$$h_{kp}=h_k+h_k a_p\delta^{c_p,c_{v(p)}}_{d_p,d_{v(p)}} \qquad \forall p\in P_A \qquad (12)$$

Constraints (11) limit the height of vertical platform pairs. If one of the platforms is angled, the new effective height is calculated in the equations (12).

4.6. Weight constraints.

A variety of weight constraints restrict positions for certain cars or certain combinations of cars. In addition to weight constraints for the truck, trailer and the individual loading planes, there are also restrictions for each individual platform. There are different restrictions if the platform is in a horizontal or angled position or if multiple platforms p are combined to a split-platform Q. To model this fact, we introduce an auxiliary variable γ_p that represents the weight limit according to p's current configuration.

$$\sum_{k\in K} w_k x_{kp} \leq \gamma_p \qquad \forall p \in P \setminus P_A \tag{13}$$

$$\sum_{k\in K} w_k x_{kp} \leq a_p wa_p^{max} + (1-a_p)\gamma_p \qquad \forall p \in P_A \tag{14}$$

$$where\ \gamma_p = \begin{cases} wc_Q^{max} & \text{if} \quad \exists Q \in P_{SP} : p \in Q \wedge sp_Q = 1 \\ wp_p^{max} & \text{otherwise} \end{cases} \tag{15}$$

Constraints (13) and (14) ensure that the weight on platform p is not exceeding the maximum weight allowed on p in horizontal and angled position. If $p \in P_A$ is not angled, and a combined platform Q is used, in equation (15) the maximum allowed weight is increased to wc_Q^{max}. This means that each platform $p \in P_A \cap P_{SP}$ has three possible weight limits, depending on the decision if $p \in P_A$ is angled or not and if p is combined with another platform to a split platform $Q \in P_{SP}$.

$$\sum_{k\in K}\sum_{p\in L} w_k x_{kp} \leq wl_L^{max} \qquad \forall L \in P_L \tag{16}$$

$$\sum_{k\in K}\sum_{p\in T} w_k x_{kp} \leq wt_T^{max} \qquad \forall T \in P_T \tag{17}$$

$$\sum_{k\in K}\sum_{p\in P} w_k x_{kp} \leq W^{max} \tag{18}$$

Constraints (16) limit the maximum weight on each loading plane, (17) limit the weight on the truck and the trailer and (18) limits the total maximum payload.

4.7. Alternative objective function for split-deliveries.

In case a single auto carrier supplies multiple dealers, the vehicles on the auto carrier should be ordered in a way that minimizes reloading along the route. This means that ideally the vehicles destined for the first dealers can be unloaded more quickly than vehicles destined for later dealers. For publications dealing solely with the problem of ordering the vehicles to reduce reloading we refer the reader to Agbegha, Ballou, and Mathur (1998) and Chen and Wang (2020). We will focus on improving the initial load according to a given dealer sequence. Our formulation builds on the basic idea proposed by Agbegha, Ballou, and Mathur (1998). A precedence graph defines the unloading costs for a specific slot on the auto carrier. We display the precedence graph of the auto carrier from Figure 2 in Figure 4. We model the costs of reloading vehicles as a reward within the objective function. First, we introduce two new parameters:

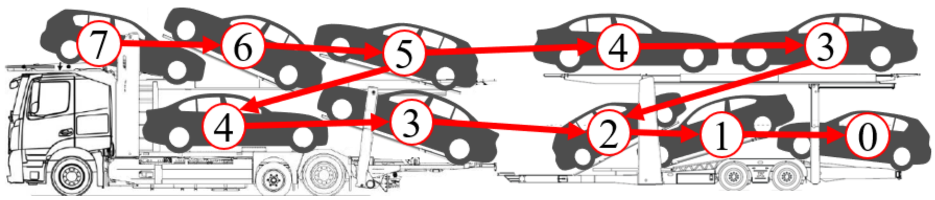

Figure 4 **Precedence graph for the auto carrier from Figure 2 when every platform is occupied.**

g_k: Priority of vehicle $k \in K$. A higher priority means earlier delivery. $g_k \in N$

c_p: Unloading cost for unloading a vehicle from platform $p \in P$. The cost is equal to the total number of platforms that must be unloaded to access the desired vehicle according to the precedence graph. $c_p \in N$

$$max \sum_{p \in P} \sum_{k \in K} x_{kp} + \frac{1}{\sum_{p \in P} \sum_{k \in K} x_{kp} g_k c_p} \tag{19}$$

The alternative objective function is displayed in function (19). The reward is the product of the priority and cost vectors as a fraction of one. Since it is a maximization problem, we add the reward to the original objective value. Ideally, high-priority vehicles are positioned on low-cost platforms in order to minimize the value in the denominator of the reward term. Of course, using this formulation the reward varies greatly with the number of dealers. However, the magnitude of the reward does not matter since it is always less than one and because our objective value is discrete. Ideally, the vehicles are ordered lexicographically according to their priorities. Vehicles with low priorities are positioned on higher cost platforms and vehicles with higher priorities occupy lower cost platforms.

5. Nesting approach

In the MILP approach in Section 4 we include detailed assignment constraints as well as relation-based constraints between auto carrier platforms. Vehicle weights and weight restrictions are exact. The calculation of the height and length effects is approximate because of their non-linear relationship. We do not consider the vehicle in its original, intricate shape but simplify it to length and height parameters as well as approximate values for the required lifting angle and the corresponding effect. In this section, we propose a geometric approach that respects the actual vehicle shapes and the dimensions of the auto carrier.

We transfer the ACLP to a geometrical nesting problem by considering the vehicles as irregular polygons which we pack inside an irregular container (the cargo area of the auto carrier). Vehicles above a certain width can simply not be loaded on the auto carrier. Thus, we can reduce the shape of the vehicles and the container to their two-dimensional lateral profile. The goal is to find a feasible arrangement of vehicle polygons within the container so that:

- All vehicle polygons are completely inside the container
- The vehicle polygons do not overlap with each other

Since the container area is continuous we do not generate vehicle to slots assignments. Therefore, we cannot exactly assign the polygons to individual platforms. Instead, we aim to indirectly satisfy the constraints from the MILP by including certain conditions within our nesting algorithm. The following sections describe the geometric tool, the selection and placement heuristics, and the limited look-ahead placement algorithm. The geometric tool explains how we represent the different geometric objects and how the geometric objects interact with each other. The nesting heuristic explains the assumptions, rules and objective of our nesting approach (nesting algorithm). The selection and placement heuristic is divided in a pre-layout phase and the limited look-ahead algorithm. In the pre-layout phase we decide the initial ordering of the items. The limited look-ahead algorithm determines the positions and orientations of the geometric objects within the container using the framework of our geometric tool while following the rules and objective described in the nesting heuristic.

5.1. Geometric tool

One of the key elements to design a geometric nesting algorithm is the choice of the so called geometric tool. The geometric tool refers to a method to describe geometric objects and their behavior in a way that can be understood by computers. In the domain of irregular packing problems with non-convex shapes and containers, the No-Fit-Polygon (NFP) and raster methods are the most common approaches due to their flexibility and performance.

The NFP_{AB} between two polygons (A) and (B) can be generated by orbiting polygon (B) around the boundary of polygon (A) while tracing a reference point of polygon (B). The resulting shape, the NFP, is the original static polygon (A) aggregated with the path of the reference point of polygon (B). This reference point can be any vertex of polygon (B). Therefore, there is no single unique NFP for two polygons. Since the reference vertex of polygon (B) can move along the edges of polygon (A) continuously, the position of polygon (B) is continuous. For the so-called Inner-Fit-Polygon (IFP), a polygon (D) is positioned inside another polygon (C). IFP_{CD} is generated by moving polygon (D) along the inner boundary of polygon (C) while tracing the path of a reference point of polygon (D). Examples for the NFP and IFP are displayed in Figure 5 and Figure 6.

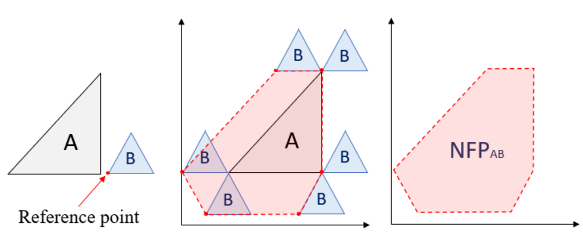

Figure 5 Example for NFP of two polygons

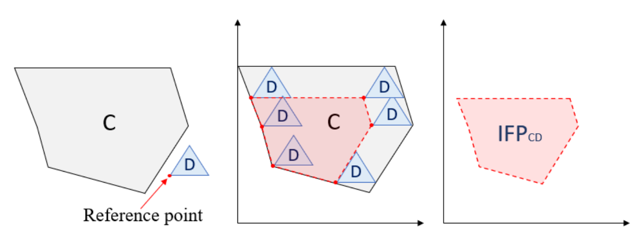

Figure 6 Example for IFP of two polygons

The main advantage of the NFP is that it can be pre-calculated. Through pre-calculation the otherwise computationally intensive algorithm becomes a fixed cost in terms of computation time (Toledo et al. (2013)). A recent concept, the No-Fit-Raster (NFR), combines the NFP with the raster approach. Like the NFP, the NFR also allows pre-calculation. Both methods differentiate between No-Fit (NF) and Inner-Fit (IF). We compared both, the raster and the NFP approach in terms of accuracy and computational time for our application. For generating the No-Fit and Inner-Fit polygons we used the approach provided by Burke et al. (2007). Generating the NF- and IF- raster (IFR) is quite straightforward. A description of an algorithm can be found in Mundim et al. (2018). The results of the comparison and a deeper analysis of NFP versus raster methods for our application can be found in Appendix B. The comparison shows that pre-calculating the IFP and IFR is feasible and can be done in less than an hour for all vehicles at nine different angles and varying degrees of detail of the geometric representation. Neither for the NFP nor the NFR, it is practical to pre-calculate the No-Fit relations due to a very high number of combinations (53 vehicle types in two orientations at nine different angles). Performance-wise, on average the raster method is slightly faster than the NFP approach for our application. However, for the NFP approach, we can use a workaround for avoiding costly dynamic NFP calculations which we explain in the next section.

5.2. Nesting heuristic

The assumptions and requirements stated in Subsection 1.1 also apply for the geometric nesting approach. We need detailed information about the dimensions of the vehicles and the auto carriers. In particular, for the auto carriers we know the minimum and maximum length of the levels and the height at certain characteristic points. Based on these measurements and the legally allowed limitations we generated polygons that represent the loading areas (containers) of the auto carriers. The outlines of the vehicles are extracted from product images and scaled to the correct size using a software tool. Instead of allowing continuous rotation, which would require much larger computational efforts, we determined multiple discrete angles based on Appendix A and through experimentation. We pre-calculated the IFPs for all vehicle polygons at nine discrete angles in forward and backward orientation for the containers of two auto carrier types. All IFP relations are stored in a look-up table. The IFP relations include every discrete position of the vehicle polygon inside the container, along with the translation vector that points toward the next position. To further reduce computation time and simplify the inclusion of weight constraints, we take advantage of the fact that the auto carriers only have two levels. We exploit that aspect in our heuristic by splitting the containers into two semi-connected zones mimicking the top and bottom levels of the auto carriers. Polygons are either inserted at the top or bottom level and must always touch either the top or bottom edge of the containers with at least one vertex. This also prevents polygons from floating between the levels which is not realistic considering our application. Usually, the objective of packing problems is to pack the items as tightly as possible in order not to waste any space. We do not only focus on finding an arrangement of items that is tightly packed but that also considers the constraints of our use case. We do not aim to minimize the height of the auto carrier but consider the maximum allowed height as a constraint. Because of that, we do not gain anything by leaving space above the polygons in the upper level. By splitting the container into zones, we can indirectly include additional constraints. For each level, we know the maximum number of platforms that can be angled as well as the weight constraints depending on the platform positions (see Section 4.6).

Another big advantage of splitting the levels is that we do not have to calculate any NFPs. Every polygon moves around the inner edges along the top or bottom of the container until it reaches the end of the level or until it touches another polygon that is already placed in the container. These movements are not calculated dynamically. Instead, we use the look-up table to find the currently best position within the container (i.e. the leftmost feasible position from the table) and the next translation vector. Now we only have to move the polygon for the appropriate amount along the translation vector (towards the first infeasible position) until it reaches the last possible position. The final position is either the end of the level or the point where the polygon touches but does not intersect a polygon that is already inside the container. The polygon does not move any further along the outer exterior boundaries of the other polygons that are already in the container. Instead, the same polygon is inserted two times, once on the top and once on the bottom level. On both levels, the polygons should be positioned as far to the left as possible. This means we iterate between a left-top fill rule on the top and a left-bottom fill rule on the bottom level. This placement rule is inspired by the zig-zag rules proposed by Mundim et al. (2018). In case there are multiple positions with the same performance we choose the top-most position on the upper and the bottom-most position on the lower level. In our application, the best position among all positions is where the largest x-coordinate of the new polygon is the smallest (i.e. the rightmost point is as far left as possible). We have also tested a right-fill rule but observed a worse performance. The area of the truck (on the left) is more restrictive and intricate than the area of the trailer (on the right). Because of that the algorithm struggles to find good arrangements to fill the area above the driver's cab of the truck. Additional tests showed that top and bottom fill rules lead to big gaps on the left and right ends of the auto carrier container. A visual scheme and further explanations of the individual steps of the algorithm and placement rules are provided in Section 5.3.2.

5.3. Selection and placement heuristic

The objective of the placement heuristic is to find the currently best possible position within the container (considering the polygons that are already inside the container) for each vehicle polygon. The algorithm takes one vehicle polygon at a time and determines the most favorable angle, direction and position on the top or bottom level of the container using a limited look-ahead algorithm following the top / bottom left-fill principle.

5.3.1. Pre-layout phase. As defined in the problem description in 1.1, the vehicles are provided on-line in a fixed sequence, and we follow a FIFO principle. In the pre-layout phase, we determine an initial set of vehicles according to multiple acceptance criteria:

- The sum of the individual vehicle weights in the initial set is below the maximum allowed weight limit of the payload
- The total surface area of bounding boxes around the vehicles in the initial set (minimum enclosing rectangle around the vehicle polygon) is less than the surface area of the container.
- The maximum number of “heavy” vehicles is not reached yet. Heavy vehicles have to stand on two platforms simultaneously due to their weight.

A variety of aspects can impact the computational intensity of a nesting algorithm. If the item shapes are simple, if there are not many different types of items, if the rotation is limited, and if the container is rather small, the computation times can be quite low. Because of that, researchers often run hundreds or even thousands of iterations with random item orders and select the order sequence with the best nesting performance. Another approach, which can be more suitable for complex items, is the so-called best-fit approach. For the best-fit approach, the algorithm selects the next item and its angle based on which item-angle combination displays the best performance. This means, that for every new item, every single item and angle must be tested, and the order is determined by the algorithm. An approach with a completely random selection of items and random angles is not practical in our case due to the large differences between vehicle types and the high number of angles and since we use a computationally more intensive limited look-ahead algorithm (which will be further explained in the next Subsection 5.3.2). Considering that, we propose the two following strategies to determine the order of the vehicle polygons:

- Limited look-ahead with increasing size: The vehicles are ordered in a single fixed sequence by increasing size from the smallest to the largest vehicle according to the area of their bounding box. The ascending order makes sense because the truck is more constrained regarding weight and height than the trailer. In the real-world process, larger vehicles tend to stand further back on the auto carrier.

- Limited look-ahead with height consideration: The initial sequence is the same as in the previous strategy. If there is a pair of vehicles that is too tall when stacked on top of each other at any position of the truck or the trailer, one of the vehicles is moved forward in the ordering. This process is repeated until every pair of consecutive vehicles in the order is below the height limit. If the set does not allow such reordering due to many tall vehicles the process is terminated.

5.3.2. Limited Look-ahead algorithm. The limited look-ahead algorithm is an adaption of a best-fit approach with fixed order. Instead of evaluating the performance of a single polygon, the algorithm always considers two polygons simultaneously. The motivation behind this approach is that by always evaluating the performance of a polygon together with the next polygon in line we can efficiently find matching pairs. Figure 7 shows a graphical scheme of the main steps of the algorithm.

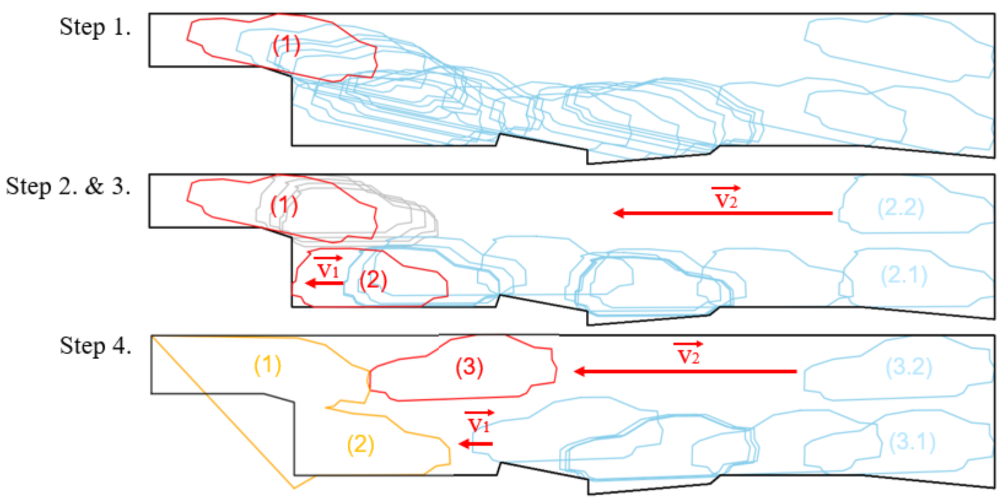

Figure 7 **Graphical scheme of steps 1. to 4. of the limited look-ahead algorithm**

- **Step 1:** For the first polygon (1) we simply check all possible positions for the polygon at a specific (arbitrary) angle within the container by accessing the pre-calculated look-up table. We choose the position with the smallest maximum x-coordinate. Step one is displayed in the first image in Figure 7. The blue polygons mark all possible locations, and the red polygon shows the best position.
- **Step 2:** From the second polygon (2) onward, we separate the container in top and bottom level. First, polygon (2) is inserted in the lower level (position (2.1)), then in the upper level (position (2.2)). From right to left we check all possible positions within the container on both levels. If at any position polygon (2) intersects polygon (1), it is considered infeasible. We then

select the last feasible position in the sequence (which is one position prior to the infeasible one) and apply the translation vector $\vec{v_x}$, that points towards the infeasible position, to every point of polygon (1). We also have to reverse the vector and apply it to every point of polygon (2). We calculate the intersection points between polygon (1) and (2) along the translation vector and trim the vector to a length so that the polygons touch but don't intersect. After applying the trimmed vector, we save the final position of polygon (2). After this is done for every angle of polygon (2) at top and bottom level, we choose the angle that results in the position with the smallest maximum x-coordinate. This position can either be at the top or the bottom level. Step 2 is displayed in the second image in Figure 7, where polygon (2) marks the final position. Blue polygons symbolize feasible, and grey polygons infeasible positions.

- **Step 3:** Step 1 and step 2 are looped for every possible angle combination of polygon (1) and polygon (2). For 9 angles that means 81 iterations in total. Out of all these combinations we choose the angle combination with the best performance. This is the combination with the left-most position of polygon (2). In case there are multiple combinations with similar performances for polygon (2), among these we choose the combination that results in the best performance for polygon (1). Now we permanently fix the angle of polygon (1). In order to test whether it makes sense to reverse polygon (1), we temporarily reverse its orientation and once more nest polygon (2) at all possible angles (another 9 iterations). If reversing the position of polygon (1) only slightly affects the performance of polygon (2), we choose the reverse orientation. Otherwise polygon (1) stays in a forward position. The position of polygon (1) is now permanently fixed.
- **Step 4:** For the next polygon, polygon (3), we must consider both, the fixed polygon (1) and the still flexible polygon (2). Now we repeat step 1 to step 3 for Polygon (2) and polygon (3). Since we now also have to consider polygon (1), whose position is fixed, for every angle of polygon (2) we merge it with polygon (1) to a create a new polygon (1 + 2). We remove the inner vertices of (1 + 2) which significantly reduces the computation time for the oncoming polygons. This procedure is inspired by Oliveira, Gomes, and Soeiro Ferreira (2000). The orange polygon in the third image in Figure 7 is the merged polygon. In this example, the best position for polygon (3) considering the performance of the next polygon (4) is now on the upper level.

Steps two to four are repeated for every polygon until the container is filled.

5.3.3. Indirect implementation of constraints. On top of the limited look-ahead algorithm and the rules proposed in Subsection 5.3, we implemented further conditions to achieve realistic vehicle polygon arrangements and to indirectly satisfy certain constraints of the MILP model:

- In addition to the separation of upper and lower level we also separate the container into truck and trailer area. For that we draw an imaginary line between truck and trailer. If a polygon is more on the right side of the line it is considered to be on the trailer, if it is more to the left it is considered to be on the truck. This allows us to respect the weight constraints for every level. If a vehicle does not fit on a certain level due to its weight, it is forced to the level above or below. If positioning the vehicle on the other level is also not possible, the vehicle stays on the original level and the weight is added to the trailer instead of the truck. This process is performed dynamically while the algorithm cycles through the different angles and polygons.
- The separation of the levels and the separation of truck and trailer enables us to generate vehicle-to-slot assignments with the nesting approach. On every level, we can just count the polygons from left to right.
- We only allow as many polygons to be rotated as there are platforms that can be angled. Additional polygons, and polygons of vehicles above a certain weight limit are only allowed to be angled to adapt to the slope of the top or bottom levels. Since we know which level a vehicle is standing on, we can also consider the weight limits of the individual platforms which can depend on whether the platforms are angled or not.
- If the performance of forward and reverse orientation of a vehicle polygon is similar, we select the reverse orientation because it will be easier to match with one of the following vehicles. This is decided by a threshold value.
- At predefined positions within the container the set of allowed angles is restricted. I.e. it is not allowed to angle a vehicle if it is positioned directly behind the drivers cab.

Figure 8 shows the final result after running a full cycle of the limited look-ahead algorithm considering the indirect constraints. The dashed line marks the imaginary separation between truck and trailer.

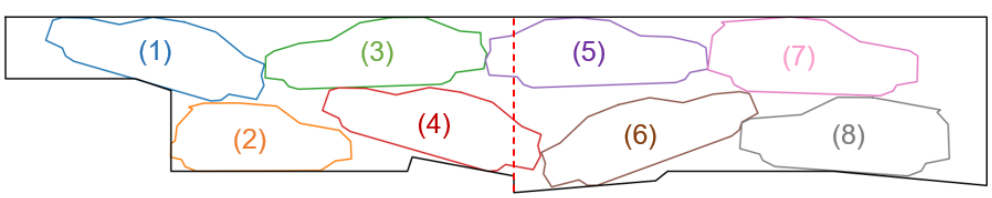

Figure 8 **Fully loaded auto carrier following the limited look-ahead algorithm**

6. Computational Experiments

We conducted a series of computational experiments using real-world data. In Section 6.1, we present the approaches considered. In 6.2, we explain the procedure we use to evaluate the feasibility of the generated auto carrier loads and Section 6.3 informs about the test instances. In Section 6.4 we compare the different layout and nesting strategies proposed in 5.3.1. In Section 6.5 we compare the results of the ACLP for randomly generated test loads based on route specific vehicle volumes. Finally, in Section 6.6, we evaluate all approaches by using large sets of vehicles with fixed sequences based on historical data to estimate potential savings. The sequences of the vehicles are taken directly from real world data (the respective transport calendar for each route).

All experiments were performed on a PC with an AMD Ryzen 5 3600 CPU at 3.6 GHz and 16 GB of RAM. We implemented the algorithms in Python 3.9.10 and used Gurobi 9.5.1 for solving linear programs.

6.1. ACLP approaches

In the following, we introduce the ACLP heuristics which we used for our benchmarks. For more detailed information, we refer the reader to the cited literature and to the respective sections of this paper.

- Industry Solution (**IS**): The current IS is based on an on-line knapsack approach with very simple capacity constraints. The different vehicle types are clustered in five groups (nicknamed T-shirt sizes in the industry) depending on their size. A coefficient is assigned to each of these groups. Vehicles are allowed to be added to the current auto carrier load, as long as the sum of their coefficients is below a certain threshold.
- ACLP model formulation by Dell'Amico, Falavigna, and Iori (2014) (**DFI**): The DFI heavily relies on coefficients. Vehicles are clustered into 14 groups and every group has up to eight coefficients for length and height effects. The levels of the auto carrier are constrained by length. Large vehicles can reduce the available length on the level above or below based on their groups' height coefficient. The effective length of a vehicle on its own level is calculated based on the actual length of the vehicle and the length coefficient of its group. We did not change anything within the model formulation. The authors provided data on the categories and the corresponding coefficients. We were able to assign most of our vehicle types to one of these categories. Only very large limousines did not seem to fit in any of the groups. Because of that we added another category and determined suitable coefficients. We did that by calculating the average height and length of the vehicles in a similar category (i.e. mid-sized limousines) and then scale up the coefficients to better represent very large limousines. The DFI does not consider weight constraints.
- The MILP formulation presented in Section 4 (**MILP**).
- Nesting algorithm (**NA**): We investigate the two strategies presented in Subsection 5.3.1 and denote them as limited look-ahead with increasing size (NA-LAH) and limited look-ahead

with height consideration (NA-LAS). Furthermore, we also compare a basic best-fit with random order approach (NA-Rand), which is common in the literature. Remember, basic best-fit strategies find the best position and angle for the current item without considering the next items. In our case the best position is the left-most feasible position in the container. We propose the following best-fit strategy with random order: We randomly generate the order of the vehicles. For each vehicle polygon, we determine the angle with the best performance using the best-fit rule. That means for every polygon we evaluate 9 different angles. We compare these three strategies in regards to solution quality and computation time with one-another. Following, we choose the best strategy for our benchmark with the other ACLP approaches. In Subsection 5.3.3 we explained that we use a threshold value for deciding if a vehicle should be oriented forward or reverse. Based on experiments we have decided that if the performance of a vehicle polygon in forward direction is only 5% better than in reverse direction, we choose reverse direction.

6.2. Check load procedure.

We are not aware of any research group that considered checking the loads generated by their ACLP heuristic for feasibility. However, this should be a decisive criterion, especially for integrated loading and route optimization. Infeasible loads caused by the ACLP can disrupt the calculated routing plan and schedule or can cause additional costs due to unplanned emergency transports. Manually checking the loads using real vehicles and auto carriers is not economically feasible. Because of that, we validate our results with a two-step validation procedure. First, we check if the same set of vehicles was transported on this route by a single auto carrier in the past. If this is not the case, we check the load using a combination of the MILP model and the NA. The MILP checks whether the platform-related constraints are complied with and generates several feasible load arrangements. This is done by using the ”PoolSolutions” feature of the Gurobi optimization suite. Since our objective function is discrete, by using this feature we can obtain multiple solutions with the same objective value that fit within the constraints of the MILP model. To avoid symmetry issues (interchanging two vehicles of the same type), we filter out solutions with different type to platform arrangements. The ordering of the generated arrangements are used as the input sequence for the NA. Because of the first step (checking against historic data), we presumably only have to check a part of the original set of loads. This allows us to extend the set of discrete angles from seven to twelve for the nesting algorithm without increasing the total computing time too much. A load is declared infeasible if the nesting algorithm does not find any arrangement to be feasible. By combining the MILP with the nesting approach we check for position related constraints and geometric validity simultaneously.

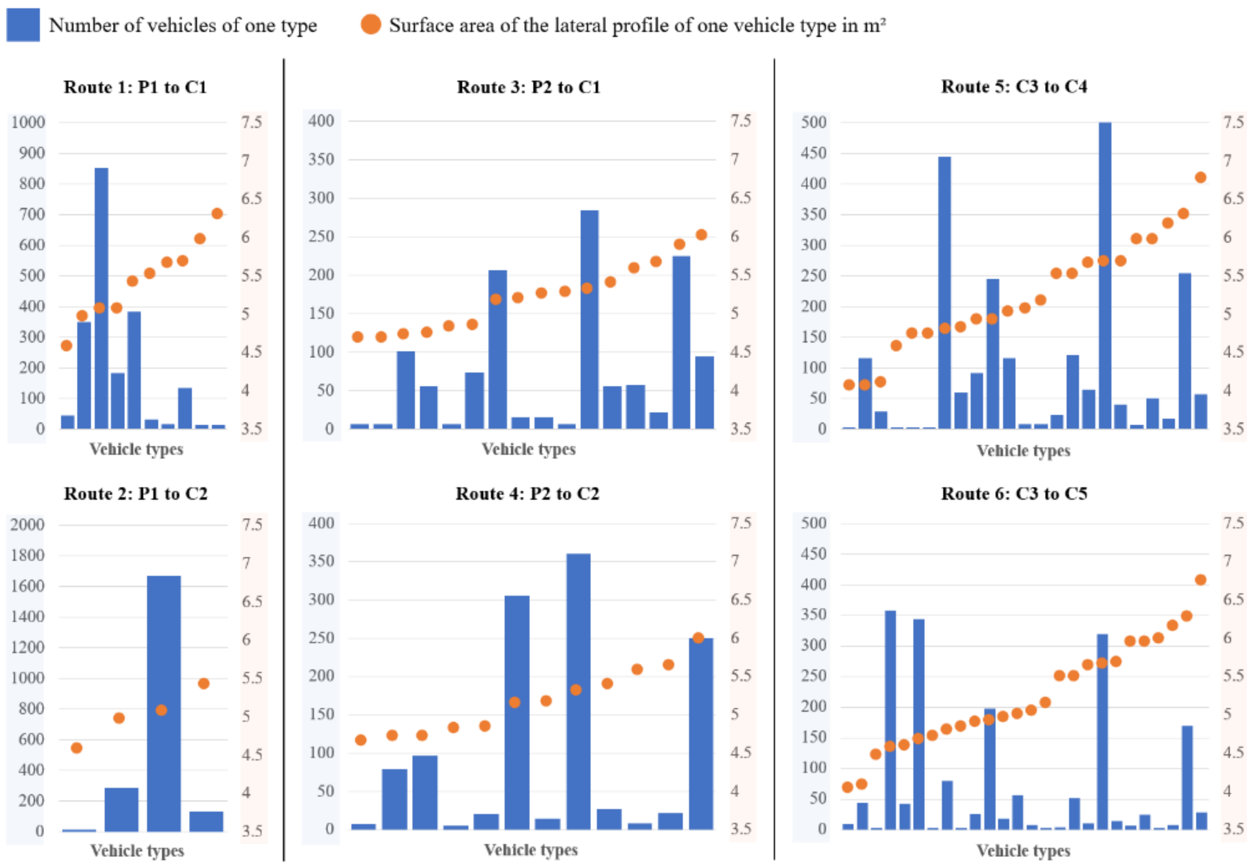

Figure 9 **Vehicle distribution for the outgoing routes from plants P1, P2 and compound C3. The data relates to the last quarter of 2021.**

6.3. Test instances

Our industry partner BMW Group provided us with recent historic data for six high-volume transport routes which are served by auto carriers. Four of these routes connect two plants (P1 and P2) with two compounds each (C1 and C2). The two remaining routes are from another compound (C3) to compounds (C4) and (C5). The compositions of the vehicle type mixes on the routes from the plants are rather homogeneous. These routes mostly serve to transport vehicles manufactured within the respective plant. The vehicle mixes on the routes from C3 to C4 and C5 are more heterogeneous. This is because C3 is used to distribute vehicles from all over the world to the local markets. Figure 9 shows the vehicle mix for each of the six routes. To display their variety in terms of size, Figure 9 also includes information about the dimensions of the vehicle types. The dimension of a vehicle type is expressed by the surface area of its lateral profile. F or o ur s imulations, we have implemented two types of auto carriers for all of the considered ACLP approaches. The first auto carrier type is very similar to the one considered by Dell'Amico, Falavigna, and Iori (2014). It also matches the one displayed in Figure 2. The second auto carrier has a shorter truck and a longer trailer. Based on the information provided Dell'Amico, Falavigna, and Iori (2014) and historical data, we adjusted the coefficients of the IS and the DFI for the second auto carrier type. Adjusting the coefficients is quite simple because the auto carriers only differ in terms of the length of the individual levels and the number of platforms that can be angled. In the case of the DFI, we changed the length coefficients by comparing the number of rotational platforms. For example, if there are three rotational platforms instead of two on one level, the length reduction coefficient is multiplied by 3/2. If there are no rotational platforms the length reduction factor is zero. We further adjusted both length- and height coefficients very slightly by trying out different values and checking against historical data.

6.4. Comparison of different layout and nesting strategies

We randomly created 300 loads of 10 vehicles each, based on the vehicle mixes on the different routes from the previous subsection. For the NA-Rand strategy we try one version with 10 iterations and another one with 100 iterations. For every strategy we calculate the average number of cars that fit inside the container (average load factor (ALF)) and the average computing time (Avg. Time) for all 300 loads. Table 1 shows the results for each strategy. Strategy NA-LAS and NA-

Table 1 Comparison for different nesting strategies.

	NA-Rand (10)	NA-Rand (100)	NA-LAS	NA-LAH
ALF	6.72	6.83	7.04	7.08
Avg. Time [s]	2.91	27.83	4.28	4.33

LAH outperform the best-fit with random order approach with 10 iterations by 4.76% and 5.35% respectively with a small increase in computation time. Increasing the iterations from 10 to 100 substantially increases the computation time. Even with 100 iterations the performance of NA-Rand is worse than NA-LAS and NA-LAH which rely on the limited-look ahead algorithm. Since both limited look-ahead strategies require the same amount of time (marginal difference mostly due to varying CPU load) we will go ahead with NA-LAH. NA-LAH performs 0.56% better than NA-LAS. For our test set there was not a single case where NA-Rand outperformed strategies NA-LAS or NA-LAH. Figure 10 displays an example where the three strategies lead to deviating load factors. Only NA-LAH could find an arrangement where all vehicles fit within the container. For the other strategies, the red polygon did not fit inside the container. Among the strategies that failed to find a feasible arrangement, NA-LAS showed the best performance (least amount of overlap).

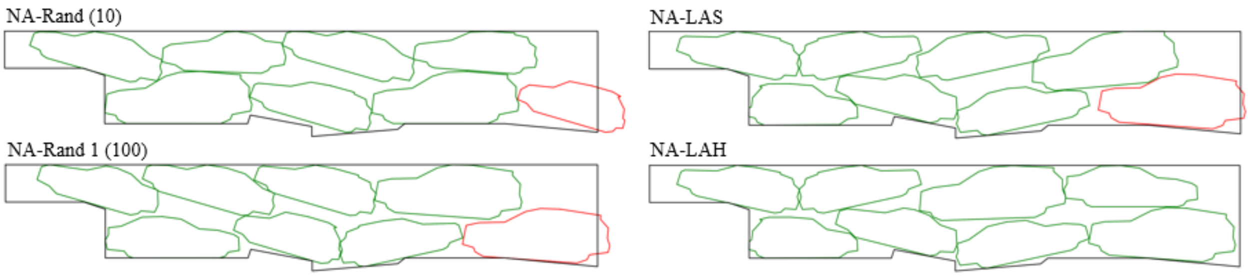

Figure 10 **Visual comparison of the performance of the different strategies for an example load.**

6.5. Comparisons and evaluation for randomly generated loads

For the first comparison, we randomly generated 500 loads of 10 vehicles each for all six routes. The vehicle mix of in total 5000 vehicles is based on the extrapolation of historical data for the respective routes. The individual loads are run through independently. Vehicles that do not fit in the current load are not considered in the next load. For all ACLP approaches, we calculate the average load factor (ALF) and the percentage of infeasible loads (INF) according to the check load procedure. For the calculation of the ALF, every load that is not feasible is corrected first. We indicate the performance increase (Perf) of the DFI, MILP, and NA in comparison to the IS in percent. To show the significance of the performance increase of the ALF, we provide the 95% confidence intervals for the absolute difference between means (95% CI ALF). Evaluating the performance of the ALF is useful to quantify potential monetary savings. However, the mean value only expresses the actual improvement to a limited extent because the variance of the load factors is very small. The decision that one more vehicle fits on the auto carrier, e.g., 7 instead of 6, has little effect on the ALF. If we consider optimization not at vehicle level but at auto carrier level the impact is much greater. At auto carrier level we evaluate the absolute number of times that an additional vehicle can be loaded in comparison to the IS. This is highly relevant for our optimization problem because for the first 6 vehicles the decision whether the vehicles will fit is significantly easier than for the additional vehicle 7. We introduce another performance indicator that states the total number of cases in which the load factor is larger in comparison to the IS (LF+) and 95% confidence intervals (95% CI LF+) for the number of cases among all sets in percent. The calculation time (Time) refers to the entire set of 500 loads and is given in minutes. Tables 2 to 5 show the simulation results for all ACLP approaches for each of the sets.

It turns out, that the MILP outperforms the other approaches on all sets in regards to Perf and the number of infeasible loads. On average, the MILP performs 1.36% better and causes 33.33% less infeasible loads compared to the IS. It appears, that none of the approaches performs significantly better than the IS on homogeneous sets (1 and 2). On more heterogeneous sets, the DFI, MILP and NA show significantly better results than the IS. The performance of the NA is 0.98% better than the IS, which places it between DFI and MILP. The performance of the NA also decreases with the increasing heterogeneity of the vehicle sets, especially on Route 6. Looking at the CIs for the ALF, the significance of the results is not entirely clear. However, by evaluating the utilization through the percentage CI of the total number of load factor improvements we observe an increase of up to 16.2% for the MILP. An increase in utilization rate of 16.2% means that every 6.17 auto carriers an additional vehicle can be loaded. On average, the DFI increases the utilization of the IS by 3.37%. The MILP increases the utilization of the IS by 9.37% and the utilization of the DFI by another 6.00%. The NA increases the utilization of the IS by 6.63%. In terms of computing time, the approaches differ greatly. The computation time does not only include the time for optimization but also the time to import the parameters and the time to construct the models in case of the DFI and the MILP. The NA takes the longest for the 500 test loads with an average of 34.18 minutes. Even though we do not reconstruct the complete model in every iteration, the time for constructing the model for the MILP is around twenty times slower than for the DFI. Model construction refers to the time Gurobi needs to import all parameters and generate the constraints. In terms of optimization, the MILP is on average fifteen times slower than the DFI.

6.6. Comparisons and evaluation for historic loads

To evaluate the economic effect of the ACLP approaches, we do further simulations based on historical vehicle sequences. We no longer simulate the loads independently but based on a fixed vehicle order. The performance of the respective optimization approach is indicated by the number of auto carriers that are required to transport the complete sequence of vehicles. Unlike in the previous simulation, vehicles that no longer fit into the current load are moved back to the beginning of the queue and have to be considered in the next load. The results of the simulations can thus be directly compared with historical data (HD). Performance indicators are the load factor (LF), the performance increase (Perf) in comparison to HD, and the number of required auto carriers (AC) for the given sets of vehicles. In practice, the IS is currently only used for routes 1 and 2 and for loads that contain vehicle types that are manufactured in P1 (approximately 90% of all loads). All other loads for routes 1 to 6 are created manually by employees based on personal experience. Even though one of the approaches might perform worse, or only slightly better than the manual process, it might still be worth it to use the approach. This is because the automation with the help of optimization software can save costs (salaries) for the required employees. We again use the check load procedure (see Section 6.2) to evaluate the feasibility of the loads. Table 6 shows the historic data along with the simulation results for all ACLP approaches. It appears, that only the MILP approach can match or improve the manually created loads for each of the six routes. The NA performs slightly worse (-0.12%) on Route 4 and the DFI performs worse on Route 3 and

Table 2 Simulation results for the IS

Route	ALF	Inf [%]	Time [min]
1	7.064	1.00	0.01
2	7.022	0.00	0.01
3	6.586	2.60	0.01
4	6.768	0.00	0.01
5	6.854	4.20	0.01
6	7.418	1.20	0.01
AVG	6.952	1.50	0.01

Table 3 Simulation results for the DFI in comparison to the IS

Route	ALF	Perf [%]	95% CI ALF	LF+	95% CI LF+ [%]	Inf [%]	Time [min]
1	7.074	0.14	0.010 ± 0.034	5	1.00 ± 0.87	0.80	0.32
2	7.028	0.09	0.006 ± 0.019	3	0.60 ± 0.68	0.00	0.24
3	6.624	0.58	0.038 ± 0.061	19	3.80 ± 1.68	2.00	0.21
4	6.814	0.68	0.046 ± 0.050	23	4.60 ± 1.84	0.00	0.21
5	6.880	0.38	0.026 ± 0.057	13	2.60 ± 1.39	3.00	0.36
6	7.494	1.02	0.076 ± 0.069	38	7.60 ± 2.32	1.80	0.30
AVG	6.986	0.48	0.039 ± 0.044	16.83	3.37 ± 1.46	1.27	0.27

Table 4 Simulation results for the MILP in comparison to the IS

Route	ALF	Perf [%]	95% CI ALF	LF+	95% CI LF+ [%]	Inf [%]	Time [min]
1	7.094	0.42	0.030 ± 0.036	15	3.00 ± 1.50	1.20	6.31
2	7.028	0.09	0.006 ± 0.019	3	0.60 ± 0.68	0.00	5.76
3	6.748	2.46	0.162 ± 0.058	81	16.20 ± 3.23	1.00	5.64
4	6.898	1.92	0.130 ± 0.046	65	13.00 ± 2.95	0.00	4.45
5	6.962	1.58	0.108 ± 0.068	54	10.80 ± 2.72	2.00	6.11
6	7.544	1.70	0.126 ± 0.070	63	12.60 ± 2.91	1.80	5.98
AVG	7.046	1.36	0.091 ± 0.053	46.83	9.37 ± 2.33	1.00	5.71

Table 5 Simulation results for the NA (Strategy 2) in comparison to the IS

Route	ALF	Perf [%]	95% CI ALF	LF+	95% CI LF+ [%]	Inf [%]	Time [min]
1	7.090	0.37	0.026 ± 0.027	12	2.40 ± 1.34	1.40	33.31
2	7.028	0.09	0.006 ± 0.019	3	0.60 ± 0.68	0.00	35.80
3	6.701	1.75	0.125 ± 0.072	56	11.20 ± 2.76	1.40	23.98
4	6.867	1.46	0.102 ± 0.063	48	9.60 ± 2.58	0.20	29.00
5	6.929	1.09	0.082 ± 0.045	36	7.20 ± 2.27	1.20	44.21
6	7.502	1.13	0.084 ± 0.052	44	8.80 ± 2.48	2.60	38.75
AVG	7.020	0.98	0.080 ± 0.046	33.17	6.63 ± 2.02	1.13	34.18

Table 6 Simulation results for ordered vehicle sets from historic data.

Route	HD		IS			DFI			MILP			NA		
	LF	AC	LF	Perf [%]	AC	LF	Perf [%]	AC	LF	Perf [%]	AC	LF	Perf [%]	AC
1	6.880	297	6.973	1.35	294	7.003	1.79	292	7.045	2.39	291	7.039	2.31	291
2	7.060	298	7.077	0.24	297	7.077	0.24	297	7.077	0.24	297	7.077	0.24	297
3	6.719	186	6.642	-1.15	188	6.699	-0.30	187	6.724	0.07	186	6.721	0.03	186
4	6.787	178	6.711	-1.11	180	6.754	-0.49	179	6.787	0.00	178	6.779	-0.12	179
5	6.840	331	6.808	-0.47	333	6.852	0.18	331	6.912	1.05	328	6.877	0.54	329
6	7.344	249	7.335	-0.12	249	7.386	0.57	247	7.447	1.40	245	7.401	0.78	246
AVG	6.938	256.5	6.924	-0.21	256.8	6.961	0.33	255.6	6.998	0.86	254.1	6.982	0.63	254.7

route 4. By using the MILP it is possible to save up to six auto carriers on route 1. Since the data only refers to one quarter of a year, the annual savings in complete auto carriers loads can amount to up to 24 for route 1. On route 2, which is the most homogeneous, the performance differences between the approaches are negligible. On route 4, only the MILP matches or slightly improves the results compared to the manually created loads. It shows that the current IS on average performs worse than the manually created loads.

7. Discussion and managerial insights

In this chapter we first discuss the simulation results of every approach individually. We evaluate the performance in comparison to the current industry solution and identify possible influencing factors. Following, we make more general observations on the advantages and disadvantages of the approaches in regards to solution quality, performance, and applicability. We also comment on the relevance of accuracy of the ACLP in regards to combined loading and routing problems.

7.1. Individual evaluation of the ACLP approaches

- Current industry solution (IS)

Remember, for the IS vehicles are clustered into a few groups and every group is assigned a certain coefficient. Determining these coefficients is no easy task since it requires in-depth analysis of historic data. Even small changes to the coefficients can greatly alter the results. It should be possible to improve these coefficients, and thus the load factor, using learning algorithms. However, an analysis of the simulation results shows that the main cause for infeasible loads is the violation of weight restrictions. Since the weight is not considered in the IS, this approach will always lead to a certain number of errors. In practice, this will become increasingly problematic with the ramp-up of battery electric vehicle production because electric vehicles are usually heavier than vehicles with combustion engines (Currently on the considered routes only around 3% are electric vehicles). Simply adding weight as a constraint for the knapsack heuristic will not be enough since there are weight restrictions not only for the auto carrier as a whole but also for individual platforms.

- The ACLP approach by Dell'Amico, Falavigna, and Iori (2014) (DFI)

The DFI is considerably more accurate than the IS but suffers from similar problems. Since length restrictions are taken into account, adapting the model to new auto carrier types is more straightforward. Every vehicle group has eight coefficients to model length and height effects for the different auto carrier levels. Thus, there are more adjustable parameters to calibrate the load factor compared to the IS. Large vehicles on the lower or upper level reduce the available length of the level above or below. Therefore, height and length coefficients must often be calibrated in parallel, which requires a lot of work and experience. In practice, doing this manually for new types of vehicles and auto carriers is undesirable in the long run. Similar to the IS, the DFI neglects important weight restrictions. This results in a considerable number of infeasible loads. Neither the IS nor the DFI include precise vehicle to position assignments. As such, they cannot simply be expanded to include all of the important weight restrictions. The weight restriction are a decisive factor in where a vehicle can be positioned on the auto carrier. This is a major drawback of the DFI. It also affects the LIFO formulation proposed by the authors which orders the vehicles according to the order of dealers. In practice, the LIFO policy proposed by Dell'Amico, Falavigna, and Iori (2014) will simply not be feasible considering the ramp-up of electric vehicle production. Since their formulation simply does not include exact assignments it is not possible to accurately calculate potential error rates. The main advantage of the DFI compared to the approaches proposed in this paper is lower computing time.

- Mixed integer linear programming approach (MILP)

The MILP approach improves the DFI in two ways. First, since the vehicles are assigned to individual slots on the auto carrier, weight restrictions can be adhered to very precisely. Second, the coefficients for length and height effects are calculated based on the actual vehicle geometries. Although this approach requires more input parameters, it saves a lot of effort since the coefficients do not have to be determined manually. Even though compared to the other approaches the MILP approach creates less infeasible loads, there are still errors. To ensure the replicability of the results, we calculated the coefficients as shown in Appendix A. For the practical application, there is of course the possibility to manually adjust the coefficients. This will slightly reduce the performance but will lead to less or even no infeasible loads.

- Nesting algorithm (NA)

The performance of the nesting algorithm in terms of solution quality is right between the DFI and the MILP. As shown in Section 6.4, the ordering of the vehicles and the nesting algorithm itself have a significant impact on the solution. The relevance of the ordering is also further illustrated by the performance increase of the NA when using the check load procedure which was presented in Section 6.2. This is because, in contrast to loading the vehicles ordered by increasing size, the loading sequence generated by the MILP considers the properties and respects the constraints of the auto carrier. Because the auto carrier was separated into four zones, mimicking the different levels, and because of the other constraints proposed in Subsection 5.3.3 only a handful of errors were due to violation of weight constraints. Most of the errors, especially on route 6, are caused by loads that consist of many small vehicles. For these loads, the NA tends to fit more vehicles on the auto carrier than what is feasible in the real world.

7.2. General observations

None of the approaches can significantly improve the load factor for homogeneous loads. This is somewhat expected. In the development phase of a new vehicle type, the load factor of a completely homogeneous load only consisting of vehicles of that type is determined manually. Thus, it is easier to maximize the capacities of the auto carrier and it is more straightforward to calculate the loading coefficients for homogeneous loads. Following, there is not much room to optimize the load factor. If the load does not consist of the same but very similar types, this advantage persists. The more heterogeneous a load, the more difficult it is to balance the coefficients and determine the best possible load factor. Compared to the other approaches, the NA is the only approach that generates tangible, visual results that can easily be analyzed by human operators. The results of the MILP are accurate regarding the vehicle to slot assignments, but the pre-calculated angles are approximate and the natural shape of the auto carrier is not taken into account. Since the DFI does not generate any concrete assignments, the results are very vague regarding the actual positions of the vehicles and rotational angles of the platforms. The computational experiments show that a combination of the MILP with the NA, such as the check load procedure, can significantly increase capacity utilization while producing fewer errors compared to only using the MILP. At the same time, the visual output generated by the NA gives the operators an inclination on how and where to position the vehicles. However, a combination of both approaches can require up to three minutes of computing time for a single load. In the future, it might make sense to develop a new approach that combines the advantages of the MILP, which are precise vehicle to slot assignments and consideration of loading constraints, with the geometric assessment provided by the NA in a more integrated approach.

For the problem we address, computing time is a minor factor. This is because in practice we have to follow a given production sequence and because the vehicles stand in a holding zone for several minutes before they are assigned to a load. If the ACLP is to be combined with routing optimization, computing times are of major concern. This is when it can be desirable to use a quicker approach.

What is common among all the ACLP approaches is that the error rate increases with increasing heterogeneity of the vehicle mix. Since the decision maker in our problem statement is a single OEM, the total number of different vehicle types is limited. From the perspective of a logistics service provider working with multiple OEMs, there is a higher risk of infeasible loads since the vehicle mix can be much more heterogeneous. This shows that an efficient yet low error ACLP approach is a critical success factor for the performance of integrated auto carrier loading and routing problems. Researches usually assume that their loading algorithm always produces feasible combinations. Errors are therefore not taken into account in the optimization. In reality, however, errors lead to increased overhead and can even affect the entire transport schedule.

Transporting six vehicles with a single truck even though seven vehicles could have fit has significant impact on transportation costs. E.g., on route 5, the costs to transport a single load are around 800€. When only six vehicles are transported instead of seven, the cost per vehicle is 133€ instead of 114€. For some vehicles this route is only one of many on their journey to the customer. Based on the data provided by our industry partner, an increase of 1.36% in capacity utilization of auto carriers can amount to seven digit savings year over year. In terms of sustainability, increasing the load factor by 1.36% means the avoidance of over 500 tons of CO2 per year considering the auto carrier transport volume for only Germany.

8. Conclusion

In this paper, we presented two approaches for solving the auto carrier loading problem for a real-world transportation problem. The goal of the ACLP is to find a feasible arrangement for a given set of vehicles on a single auto carrier. First, we improved current state-of-the-art model formulations by modeling the ACLP as a mixed integer linear program with quadratic constraints and binary decision variables. Second, we adapted and implemented a nesting algorithm that solves the problem geometrically.

We have shown that the MILP approach outperforms the current industry solution and a state-of-the-art approach from the literature. In comparison with the industry solution on average, we can increase the load factor by 1.36%, increase the utilization in 9.37% of cases and decrease errors by 33.33%. That increase of utilization means that on average every tenth auto carrier can load an additional vehicle.

We do not have to rely on historical loading data to determine loading coefficients. Instead, we developed a procedure that calculates height and length effects based on objective input parameters (vehicle geometry and size). The presented methodology refers to the German market. Because of the replicable modular modeling approach, the MILP model formulation can be easily extended to other markets and different types of auto carriers. The nesting algorithm performs worse, but still increases the load factor by 0.98% and the utilization by 6.63% compared to the current industry solution. Both approaches perform better than the state-of-the-art ACLP formulation from literature and produce less errors because they include weight constraints. As passenger cars are becoming more heavy, this is an important aspect. The increase in performance comes with a cost in terms of higher computation time. Since we isolated the ACLP due to our problem statement, we did not require fast computing times. We acknowledge that our proposed algorithms might be less suitable for application in integrated routing and loading problems with large data sets. However, with the help of simulations, we have shown that more approximate approaches can lead to solutions which are often infeasibile. Based on these findings we propose that future work on the ACLP and ATP should be more attentive to real-world constraints and the complexity of modern auto carriers. In case the solution approach is supposed to be implemented for day-to-day operation, and not only for occasional strategic decisions, it is very important to also consider the effects on an operational level. A more visual approach, such as the nesting algorithm presented in this paper, can help decision makers such as transport planners or even truck drivers to perform their tasks.

Future work might evaluate the effect of relaxing the strict LIFO requirements so vehicles can be pooled to build multiple optimal loads for the same destination at once. This can increase the load factor but at the same time will require more storage capacity (a bigger parking lot). As a next step, future work may assess the performance of integrated routing and loading problems considering a more sophisticated ACLP formulation such as the MILP approach proposed in this paper. Another future contribution could be to expand or adjust the MILP formulation to be more accurate in terms of geometric precision. With the correct outputs, it might be possible to create a visualization that is close to the actual arrangement of vehicles on the auto carrier in the real world. An accurate visualization could not only support planning decisions but also support the truck drivers on an operational level. With the check load procedure in Section 6.2, we pitched a first idea to combine a MILP formulation with a nesting algorithm. A more integrated approach may leverage the benefits of both approaches while avoiding their drawbacks. Another research direction, albeit in a different field, could be to use ACLP modeling approaches to improve the designs and features of new auto carrier types.

Appendix A: Calculation of the loading coefficients

The coefficients for the effective height increase H^{coeff}_{Gain} and effective length decrease L^{coeff}_{Reduc} are calculated for a vehicle k_1 that is on a platform angled above a second vehicle k_2. In the following we explain how these coefficients are calculated for the application in this paper. We only use the height, length and type of a vehicle as an input. Thus, the calculations are approximate. We assume that k_1 is at an angle so that the facing wheel axle of k_1 is centered above the facing wheel axle of k_2. Every vehicle is of a certain type t. There are four types of vehicles, displayed in Figure 11: Limousine, Hatch, SUV and compact cars. Every vehicle has a certain height H, length L and wheelbase L_W. The wheelbase is either known or can be approximated to 60% of the total length for limousines and hatches and 70% for SUVs and compact cars. To determine the required angle α we also need to know the height of the bonnet H_B and rear H_R of vehicle k_2 above its wheel axles. This value depends on the total height and type of vehicle k_2. Figure 12 shows the measuring points H_B and H_R and Table 7 informs about their relative height according to H.

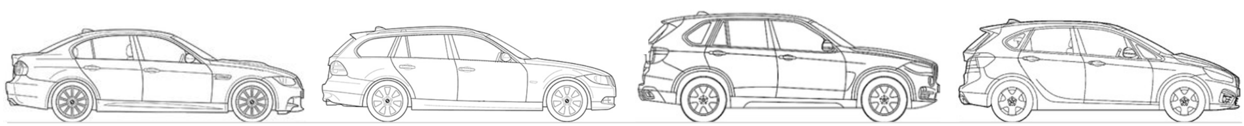

Figure 11 **Vehicle types from left to right: Limousine, Hatch, SUV, Compact**

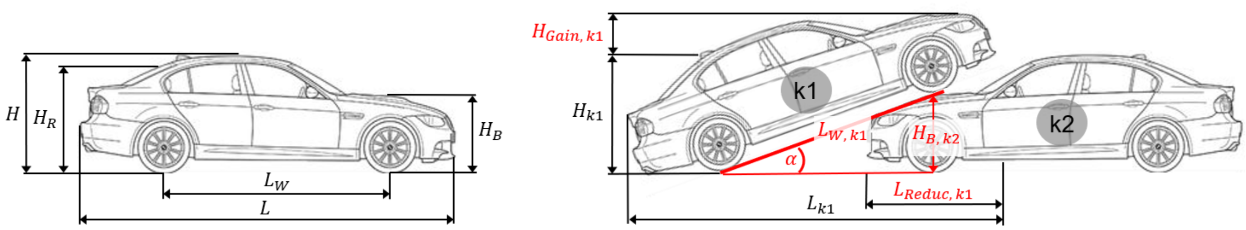

Figure 12 **Calculation of the coefficients for effective height increase and length decrease**

Table 7 **Calculation coefficients for the height of bonnet and rear for the different types of vehicles**

Type	Limousine	Hatch	SUV	Compact
H_B	0.55	0.55	0.60	0.50
H_R	0.80	1.00	1.00	1.00

For the example displayed Figure 12 the required angle α can be calculated using L_{W,k_1} and H_{B,k_2}:

$$\alpha = \sin^{-1}\left(\frac{H_{B,k2}}{L_{W,k_1}}\right) \tag{20}$$

Based on the vehicle types considered in this paper α is between 16° and 22° if k_2 is facing k_1 and between 25° and 37° if k_2 is not facing k_1. The orientation of both vehicles k_1 and k_2 is decided by the decision variables d_p and d_{vp}. For calculating the approximate height gain coefficient H^{coeff}_{Gain,k_1} for vehicle k_1, we need the length and height of vehicle k_1 and angle α. As displayed in Figure 12, angle α describes the lifting angle of the wheel axle of vehicle k_1. If we now want to calculate the effective height of vehicle k_1, we have to apply angle

α for the full length L_{k_1} of vehicle k_1. Additionally, we define a value to model the orientation and original height of vehicle k_1. As shown in Figure 13, this value amounts to approximately half of the height of the bonnet or rear of vehicle k_1, depending on its orientation. The length reduction is approximated based on the length of vehicle k_1 and its wheelbase L_{W,k_1}. Since the facing axles of both vehicles are centered, the new effective length of vehicle k_1 is approximately its wheelbase. The equation for the calculation of the height gain coefficient H^{coeff}_{Gain,k_1} for facing vehicles is displayed in (21) (If the vehicles do not face each other, $H_{B,k1}$ is replaced by H_{R,k_1}). The length reduction coefficient L^{coeff}_{Reduc,k_1} is calculated in equation (22).

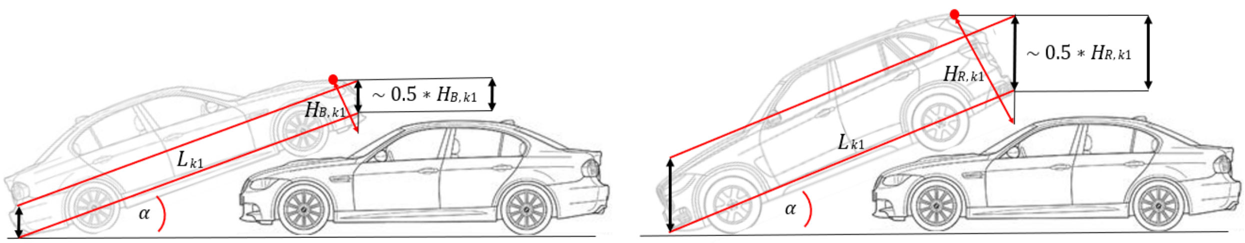

Figure 13 Further calculations of the coefficients for effective height increase and length decrease

$$H^{coeff}_{Gain,k_1} \approx \frac{\sin(\alpha) \cdot L_{k_1} + 0.5 \cdot H_{B,k_1}}{H_{k_1}} \tag{21}$$

$$L^{coeff}_{Reduc,k_1} \approx \frac{L_{W,k_1}}{L_{k_1}} \tag{22}$$

Adjusted for the model formulation in Section 4:

$$\text{Height increase factor } \delta^{c_p, c_{v(p)}}_{d_p, d_{v(p)}} = H^{coeff}_{Gain,k_1} \tag{23}$$

$$\text{length reduction factor } \lambda^{c_p, c_{v(p)}}_{d_p, d_{v(p)}} = 1 - L^{coeff}_{Reduc,k_1} \tag{24}$$

This approximation assumes that the vehicles are standing on perfectly horizontal levels. The slope of levels or platforms of the auto carrier that are not horizontal are taken into account indirectly within the height restrictions.

Appendix B: Comparison of No-Fit-Polygon vs. Raster approach

Raster methods are approaches that divide a continuous container area into discrete zones. The geometric information of the container and the item shapes are transformed into binary matrices. In contrast to the NFP and IFP, this means that there is a limited number of positions for every item inside the container. In the most basic application, a reference point on the boundary of the item is positioned on every discrete point within the container. The matrix of the item is added on top of the matrix of the container. If every point of the item is inside the container the position is considered feasible. An example is displayed in Figure 14. The resolution of the grid influences the accuracy of raster methods. A high resolution yields high accuracy but is more costly in terms of computation time.

In the following sections of this appendix, we will compare different aspects of the No-Fit-Polygon approach and the raster approach regarding solution quality and computation time for their application on the ACLP.

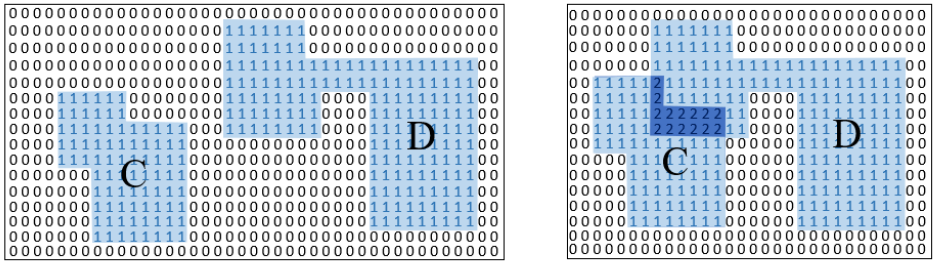

Figure 14 Overlap test for rasterized polygons.

The comparisons include shape representation, No-Fit and Inner-Fit pre-calculation, and nesting performance. For the polygon approach, the items are represented as a set of vertices connected by lines. For the raster approach the items are converted to binary matrices. The accuracy of the geometries increases with more vertices in the case of the polygons and higher mesh resolution in the case of the raster representation. Figure 15 shows the effect of reducing the number of vertices of a polygon on the left and decreasing the resolution of the matrix on the right. The initial image of the vehicle has a resolution of one pixel per centimeter. We observe that reducing the number of vertices decreases the accuracy of the overall shape while reducing the resolution decreases the accuracy of the representation of the edges.

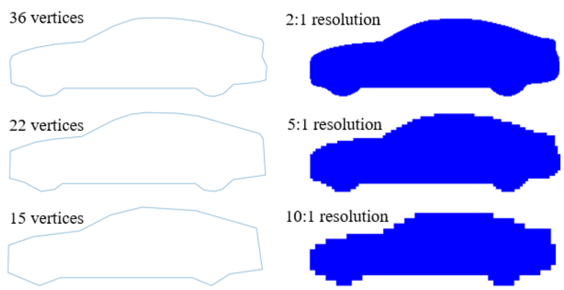

Figure 15 Comparison of geometries NFP vs. Raster approach

B.1. Computation times for pre-calculating all NFP/NFR and IFP/IFR relations

There are 53 different types of vehicles and each vehicle should be rotated at nine different angles. Vehicles can also be oriented forward or backward. For the IFP/IFR that means we have to pre-calculate a total of $53 \cdot 2 \cdot 9 = 954$ combinations for a single auto carrier. For NFP/NFR we must calculate the relations between all types of vehicles in both orientations, at all nine angles. This results in $53 \cdot 53 \cdot 4 \cdot 81 = 910.116$ combinations. Table 8 informs about the computing times for NFP/NFR and IFP/IFR for 15, 22, and 36 vertices and a resolution of 10:1, 5:1, and 2:1 respectively. In total, we have computed 50 different NFPs / NFRs and 50 different IFPs / IFRs. The average time is given in seconds.

Even though the computing times are rather short individually, looking at the number of combinations for the No-Fit relations shows that pre-calculation is not practical. Even for the raster approach with a resolution of 10:1 (which is a very rough representation of the item), the total theoretical calculation time for all NFR relations amounts to more than 70 hours. Pre-calculating the IFP and IFR is much more promising with computing times between 10 and 60 minutes depending on the detail of shape representation.

Table 8 Computing time for IF and NF at different levels of detail.

	Polygon (vertices)			Raster (resolution)		
	15	22	36	10:1	5:1	2:1
Inner-Fit [s]	0.86	1.95	3.78	0.61	1.02	2.32
No-Fit [s]	0.88	4.02	56.31	0.28	1.76	9.03

B.2. Computation time and accuracy of the nesting algorithm

Figures 16 and 17 show a direct comparison of the results of the NFP and raster approach using the limited look-ahead algorithm described in Subsection 5.3.2. Without pre-calculation, for nine different angles and vehicles in forward and backward orientation, the algorithm requires 89 seconds for the NFP approach with 15 vertices and 65 seconds for the raster approach with a 10:1 resolution. For 22 vertices and a 5:1 resolution, the algorithm requires 211 seconds and 147 seconds respectively. However, compared to the NFP approach, looking at the figures the raster approach suffers much more from the decrease in resolution than the NFP approach from reducing the number of vertices. For neither approach, it is practical to calculate the NFP and NFR relations. However, if we consider the workaround of splitting the auto carrier into an upper and lower level, which is proposed in Section 5.2, we do not need to calculate any NFPs. Instead, we only have to do intersection tests. For the NFP approach, this means checking if any of the exterior lines of polygon (A) intersect with any of the exterior lines of polygon (B), which is simple algebra. For the raster method, we have to check if both polygon matrices overlap at one or more points. This means every time we move one of the polygons a single step we have to loop through the matrices of both polygons to check for overlaps. Rather than checking all possible positions within the container grid, we only check cells that are within the polygon matrices. For example, this can be achieved by first identifying the bounding box of each shape and then only checking the cells within these bounding boxes. Assuming a 5:1 grid resolution on average we would require anywhere between 80 and 400 checks for every new item if we go through the feasible IFR positions pixel by pixel resulting in 0.04 to 0.16 seconds of computing time for every intersection test. In comparison, testing for line intersection requires 0.003 to 0.006 seconds. Even if we would reduce the resolution of the raster or use a heuristic to reduce the number of checks by a factor of ten, the NFP approach still outmatches the raster approach in terms of computing times.

Figure 16 **Nesting with 5:1 resolution vs. 22 vertices**

Figure 17 **Nesting with 10:1 resolution vs. 15 vertices**

References

Agbegha GY, 1992 *An optimization approach to the auto-carrier problem*. Ph.D. thesis, Case Western University, Cleveland.

Agbegha GY, Ballou RH, Mathur K, 1998 *Optimizing auto-carrier loading. Transportation Science* 32(2):174–188.

Al Theeb NA, Hayajneh MT, Jaradat MY, 2021 *New strategy to improve the dotted board model for solving two dimensional cutting and packing problems. Computers & Industrial Engineering* 159:107467.

Bennell JA, Oliveira JF, 2008 *The geometry of nesting problems: A tutorial. European Journal of Operational Research* 184(2):397–415.

Bonassa AC, Cunha CB, Isler CA, 2019 *An exact formulation for the multi-period auto-carrier loading and transportation problem in brazil. Computers & Industrial Engineering* 129:144–155.

Burke E, Hellier R, Kendall G, Whitwell G, 2007 *Complete and robust no-fit polygon generation for the irregular stock cutting problem. European Journal of Operational Research* 179(1):27–49.

Chen F, Wang Y, 2020 *Downward compatible loading optimization with inter-set cost in automobile outbound logistics. European Journal of Operational Research* 287(1):106–118.

Chen H, 2016 *A heuristic algorithm for the auto-carrier loading problem. International Journal of Shipping and Transport Logistics* 8(1):21–50.

Dell'Amico M, Falavigna S, Iori M, 2014 *Optimization of a real-world auto-carrier transportation problem. Transportation Science* 49(2):402–419.

Dyckhoff H, 1990 *A typology of cutting and packing problems. European Journal of Operational Research* 44(2):145–159.

Guastaroba G, Mor A, Speranza MG, 2022 *Integrated Vehicle Routing Problems: A Survey*. Salhi S, Boylan J, eds., *The Palgrave Handbook of Operations Research* (Springer, Cham), 73–103.

Guo B, Zhang Y, Hu J, Li J, Wu F, Peng Q, Zhang Q, 2022 *Two-dimensional irregular packing problems: A review. Frontiers in Mechanical Engineering* 8.

Hu ZH, Zhao Y, Tao S, Sheng ZH, 2015 *Finished-vehicle transporter routing problem solved by loading pattern discovery. Annals of Operations Research* 234(1):37–56.

Iori M, Martello S, 2010 *Routing problems with loading constraints. Top* 18(1):4–27.

Iori M, Salazar González JJ, Vigo D, 2007 *An exact approach for the vehicle routing problem with two-dimensional loading constraints. Transportation Science* 41(2):253–264.

Leao AA, Toledo FM, Oliveira JF, Carravilla MA, Alvarez-Valdés R, 2020 *Irregular packing problems: A review of mathematical models. European Journal of Operational Research* 282(3):803–822.

Liu J, Smith AE, Qian D, 2016 *The vehicle loading problem with a heterogeneous transport fleet. Computers & Industrial Engineering* 97:137–145.

Mundim L, Andretta M, Antónia MC, Oliveira JF, 2018 *A general heuristic for two-dimensional nesting problems with limited-size containers. International Journal of Production Research* 56:70–732.

Oliveira JF, Gomes AM, Soeiro Ferreira J, 2000 *Topos – a new constructive algorithm for nesting problems. OR Spektrum* 22(2):263–284.

Souza Queiroz LR, Andretta M, 2020 *Two effective methods for the irregular knapsack problem. Applied Soft Computing* 95:106485.

Sun Y, Kirtonia S, Chen ZL, 2021 *A survey of finished vehicle distribution and related problems from an optimization perspective. Transportation Research Part E: Logistics and Transportation Review* 149(2):102302.

Tadei R, Perboli G, Della Croce F, 2002 *A heuristic algorithm for the auto-carrier transportation problem. Transportation Science* 36(1):55–62.

Toledo F, Carravilla M, Ribeiro C, Oliveira JF, Gomes AM, 2013 *The dotted-board model: A new mip model for nesting irregular shapes. International Journal of Production Economics* 145:478–487.

Valle AMD, de Queiroz TA, Miyazawa FK, Xavier EC, 2012 *Heuristics for two-dimensional knapsack and cutting stock problems with items of irregular shape. Expert Syst. Appl.* 39(16):12589–12598.

VDA, 2020 *Annual report 2020: The automotive industry in facts and figures.* Report, Verband der Automobilindustrie, Berlin.

Venkatachalam S, Sundar K, 2016 *Branch-and-price algorithm for an auto-carrier transportation problem. arXiv: Optimization and Control* Working paper, Texas A&M University, College Station TX.

Wäscher G, Haußner H, Schumann H, 2007 *An improved typology of cutting and packing problems. European Journal of Operational Research* 183:1109–1130.

DOI: 10.1287/trsc.2022.0373
URN: urn:nbn:de:hbz:465-20240515-145442-7

This is the "Authors Accepted Manuscript" version of: Jäck, C., Gönsch, J., & Dörmann Osuna, H. Load Factor Optimization for the Auto-Carrier Loading Problem. *Transportation Science* 2023, 57:6, 1696-1719.

The final article version (Version of Record) was published online 17 October 2023 and is available at: https://doi.org/10.1287/trsc.2022.0373