Dynamic Platoon Formation in a Port Area

Master Thesis

Econometrics and Management Science
Master Operations Research and Quantitative Logistics

Author: Delaja Post
Student number: 444948

Supervisor: Dr. T.A.B. Dollevoet
Second reader: Dr. M.A. Mankowski

Company: TBA Group
Supervisor: G. Bast

October 25, 2021
ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics

Abstract

On a daily level we get reminded that something has to change to save the environment. Recent innovations have led companies to explore the opportunities of platooning. This thesis explores the potential of platooning in a port area by means of simulation. This problem can be modeled as a Pickup and Delivery Platooning Problem with Time-Windows. Due to the NP-hardness of this problem, this model cannot be used to test the performance of the simulation against. We test three different dynamic platoon formation strategies, among which two ad hoc and one group approach, by means of event based simulation. We construct a basic simulation in Java for parameter tuning and a sensitivity analysis on the speed and costs of trucks. From the simulation it is concluded that the ad hoc approaches show most potential. The sensitivity analysis advises to reduce the speed as this lowers the fuel consumption more, even though higher speeds generate higher relative fuel reductions. Further, even for large truck costs it remains favorable to apply platooning. Then, a more extended simulation is built in TBAs TIMESquare software in which we test the two ad hoc approaches to approximate the potential of platooning even more accurately. The fuel reduction of the two ad hoc approaches is comparable, however, the adjusted approach appears to be more interesting for companies as it is more cost efficient. After all, platooning is expected to save 10 up to 16% of the fuel and can save up to two million euros in total costs every ten years.

Contents

1 Introduction

About 80% of the total global warming potential of all greenhouse gases is caused by carbon dioxide (CO_2) (European Commission, 2006). Currently, road freight transport accounts for about a quarter of the transport CO_2 emissions in the EU. This increased over the past few years, despite all efforts, due to an increased amount of road freight traffic. In 2019 the EU adopted emission standards for heavy-duty vehicles to reduce the emissions with 30% by 2030, following the Paris Agreement (European Commission, 2019).

In the Netherlands, the Port of Rotterdam accounts for 13.5% of the nationwide produced carbon emissions (Port of Rotterdam, 2021a). This used to be 16%, but due to effective approaches the port managed to reduce its total carbon emissions with about 27% over the past four years. Given the pressure by the EU to reduce emissions and given the fact that ports these days do not compete based on size anymore but based on digitization and sustainability, the Port of Rotterdam is now aiming to become the 'smartest port'. Their aim is to be a carbon neutral port by 2050, for which one of their sub-goals is to make all port-related journeys by road emission free by 2040 (Port of Rotterdam, 2021b).

Daily around 12,000 road journeys occur through the port, which will become even more when the Container Exchange Route (CER) is finished. The Port of Rotterdam Authority is currently building this 17 kilometer road network on the Maasvlakte, which connects the deep sea terminals, depots, and customs with each other. This will enable a more efficient container exchange between the different facilities. This route will be operated by automated trucks, which will all drive separately over this dedicated road. However, a relatively new approach has caught the interest, which could significantly reduce the total fuel consumption. We are talking about 'platooning'.

A platoon is a group of electronically connected trucks following each other with a close inter-vehicle distance, where the leading truck decides the speed and route. The close distance between the trucks reduces the aerodynamic drag of the following trucks which leads to fuel and therewith emission savings. The total benefits depend on the inter-vehicle distance, velocity, and position of the vehicle in the platoon. According to Bonnet & Fritz (2000) a following truck can reach a fuel reduction of 21% if it drives 80km/h with an inter-vehicle distance of 10 meters, and 16% if either the distance is 16 meters or if they maintain a speed of 60km/h. In addition, platooning leads to better road utilization caused by less speed variability and smaller inter-vehicle distances. Finally, it is said to increase traffic safety by means of shorter reaction times and fewer human errors.

However, the potential of platooning is not only affected by its technical aspects, but also by the platoon formation process. The coordination strategy states whether platoons are formed by adjusting the scheduled departure times of trucks or by means of adjusting speed. Within the literature there are two versions of the platooning problem, one that schedules the platoons in advance and the other coordinates platoons in real-time (Bhoopalam et al., 2018). In general, little research has been done on this topic and most of it regards the first version. The problem considered in this paper, however, is the real-time platooning problem, which aims to minimize the total fuel costs. Different than other papers, we also include the assignment and therewith routing of the trucks in our problem. Therefore the aim of this research is to find the maximum potential of platooning on the CER. We will provide an exact model, called the Pickup and Delivery Platooning Problem with Time Windows (PDPPTW), as an offline solution approach, which assumes that the container arrivals are known in advance. Then, for the real-time problem we define three different methods of which two are ad hoc approaches and one is a group approach. As this research is done by means of an internship at TBA Group, we implement and tune the best approaches in the simulation software of TBA. The benefits of platooning are then determined by comparing our results with the results of TBAs current CER model.

The remainder of this paper is organized into six sections. In Section 2 relevant researches in the literature are discussed. Section 3 describes the problem in more detail and introduces the fuel model. The mathematical model with corresponding notation and graph structure are presented in Section 4. Then, Section 5 presents the three solution approaches for the on-line version of the problem. Here, we describe the buildup of the Java and TIMESquare models, and the assumptions that were made. Section 6 reports and analyzes the computational results of the previously described sections. In addition, a sensitivity analysis is done on the speed and the costs of an automated truck. Finally, we conclude this thesis in Section 7 and we discuss the research together with some recommendations on future research in Section 8.

2 Literature Review

Platooning is a relatively new concept that has caught increasing interest over the past few years. Previous researches have mainly been based on the technical aspects of platooning, the safety requirements, and the fuel consumption. Only a small group of researches have considered the problem of routing vehicles in such a way that platoons are formed in order to save fuel consumption (Bhoopalam et al., 2018). Boysen et al. (2018) concluded already that this problem is only solvable for small instances. They proved that this problem is NP-hard and therefore cannot be solved to optimality for large instances within reasonable time.

In general, the papers that review the platooning problem are similar in the way that they assume that the origin and destination of the trucks are already known, and that they have to be routed in such a way that platoons are formed. Then, these researches can be divided into two groups: papers that form platoons along the arcs (roads) by adjusting speeds, and the ones that form platoons at the nodes by letting some trucks wait for the others.

2.1 Platoon formation by speed adjustment

Liang et al. (2015) lower the speed of the leading truck while increasing that of the following truck to catch up. They introduce a pair-wise algorithm, but conclude that the savings from their simulations were less than expected due to the speed variations which increased the energy usage, and the platoon formation process that occurred later than expected which led to less platoon generated savings. Moreover, Van de Hoef (2016) states that this approach of finding common routes and defining platooning opportunities is too slow. Instead they present an exact approach to determine intersecting intervals of trucks.

A more real-time approach is given by Larson et al. (2013). They place controllers at junctions which determine if two approaching trucks from different roads can adjust their speeds in order to form a platoon at the junction. Note that the benefits of platooning must be larger than the additional required fuel to form the platoon. This, however, appeared to be intractable for more than four vehicles and they therefore propose again a pair-wise algorithm. Results show that trucks that are at most 13 kilometers away from one another could have potential to catch-up in the German high-way network. Since our roads are relatively short, this platoon formation strategy does not seem to be beneficial.

2.2 Platoon formation by waiting

Larson et al. (2016) consider both routing and platoon coordination. They coordinate platoons by delaying their departure times and test different waiting times. Their exact model then routes trucks such that platoons merge and split at intermediate points.

On the other hand, Boysen et al. (2018) solve real-world size problems to optimality by using a much simpler network in which all trucks share the same path. Like in our case, they included time windows in their model. By testing different numbers of trucks, waiting times, and platoon sizes they show the influence of these parameters on the fuel consumption.

Larsson et al. (2015) present both an exact model and two heuristics. However, they do not consider time windows in their model and therefore simply look which trucks are likely to take the same routes. Their second heuristic is a hub based heuristic which rates every edge for every truck based on the likeliness it will traverse that edge. Its solution is then improved by a local search. Differently, Adler et al. (2020) show a stochastic approach in which vehicles arrive at a start station at random intervals following a Poisson process and drive along a high way to the end station. They present three departure policies and randomly choose one of these to determine the moment at which the platoon may depart.

2.3 Assignment problem

The former approaches assume that the trucks are already assigned to a certain origin and destination for just one assignment, because of which only the route between these nodes is adjustable. Our problem, on the other hand, is a dynamic problem in which trucks still have to be assigned to pickup and delivery locations to deliver containers.

A related problem is the scheduling problem of batching machines. A batching machine, which processes several jobs simultaneously, can be seen as a platoon that transports several trucks at once. Cigolini et al. (2002) present a dynamic scheduling procedure for these machines which minimizes the flow time of jobs waiting to be processed. Interesting is that they note that there is a trade-off between waiting and the utilization rate, similar to our trade-off between waiting and the platoon length. They include this by forecasting future arrivals and reviewing the start time of the machine with every new job arrival.

Zhang et al. (2001) base the time at which the next batch will be processed on the arrival and processing time of a job, and update this at every new job arrival. Different from the previous paper which assumes a constant processing time for the entire batch, Zhang et al. (2001) assume different processing times for every job which are considered to be known in advance. Cigolini et al. (2002) do not require the processing time to be known but use parameter tuning to define the best value. Consequently, Cigolini et al. (2002) seems to be more applicable as we do not have clear processing times.

Another view is described by Yang et al. (1999), who present an on-line algorithm for truck fleet assignment problems. Their MIP is solved for every new job so that trucks can accept the assignment or change their assignments. In our network, trucks cannot easily turn around to change assignment. Moreover, containers arrive in large batches, because of which there might not be enough trucks available to assign all containers, which causes delays as the MIP will only be solved again when the next batch arrives.

Also for robots, dynamic role assignment is used often for cooperative assignments such as by Chaimowicz et al. (2002). These types of problems assume that each job requires a known amount of robots, whereas we do not know how long the platoon should be.

2.4 Conclusion

From literature research it can be concluded that, to the best of the author's knowledge, the platooning problem of routing and coordinating platoons has not been viewed in a dynamic or on-line setting yet. Consequently, related researches are not fully applicable to our on-line approach. Nevertheless, there are certain takeaways. Based on different simulations, it can be concluded that platoon formation along the road by means of speed variability is only interesting on long high-way roads. Moreover, one of the benefits of platooning is said to be the reduction in speed variability, which is neglected in this type of formation.

Therefore we will focus on platoon formation at nodes (terminals) by means of introducing waiting times. This maximizes the platooning potential by letting trucks platoon the entire way from origin to destination. Researches regarding this approach showed that the number of trucks, waiting times, and platoon lengths are strongly correlated. Therefore we test different fixed parameters to determine the best parameter values for our problem, and test adjustable parameters as is done by Cigolini et al. (2002).

To stimulate the platoon formation even more, we will propose certain assignment approaches that assign trucks to containers in a smart way. In addition, we will propose an exact model to determine the maximum platooning potential given that all container arrivals are known in advance.

3 Problem description

In this section we first discuss the background of the problem and give more insight into the assumptions for the model itself afterwards. In Section 3.2 we present a formula to measure the fuel consumption and corresponding values for the parameters.

The Port of Rotterdam is currently building the Container Exchange Route (CER) on the Maasvlakte, which is a 17 kilometer long road that connects the five deep sea terminals, empty depots, and customs with each other. From the beginning of next year this track will be used to exchange containers between the different facilities to enhance an efficient container flow through the Port of Rotterdam. Automated Road Trucks (ARTs) will transport the containers between the facilities without being interrupted by the other traffic on the Maasvlakte as this is a separate route. The Port Authority expects that after the start-up phase eventually around 1 million TEU (Twenty Foot Equivalent Unit) will be exchanged per year. TBA has developed a simulation model to study this container exchange and to determine the number of ARTs that will be necessary.

The Port of Rotterdam however wants to be the 'smartest port' by being a leader in energy transition, digitization, and innovation to increase the efficiency, reliability, and competitive position of the supply chains through Rotterdam. That is why, platooning has aroused interest. Platoons are unmanned trucks that are electronically connected to each other, in which the front truck decides the speed and route. The worldwide interest in platooning has increased strongly over the past years as it can reduce fuel consumption significantly compared to driving trucks separately. These fuel and therewith emission reductions fit well in the goal of the Port of Rotterdam to be the smartest port.

3.1 Assumptions

We now get into more detail on the simulation models and assumptions. As we develop a simulation model, we are considering a dynamic or so-called on-line approach that makes real-time decisions. The original simulation model by TBA routes trucks on the CER to exchange the containers between the different facilities. Whenever a truck finishes its current assignment, their heuristic determines which container is closest to be picked-up and closest to the end of its deadline. In case there are currently no containers available, the trucks drive to the parking area to wait for the next batch of containers to arrive. As this is a dynamic problem, it is not known in advance when the next batch of containers will arrive. Therefore, these trucks wait as they do not know where and when the next batch of containers will arrive.

The Port of Rotterdam has determined an expected number of containers that will be exchanged between every pair of facilities per year. The simulation generates batches of containers following a uniform distribution based on a week volume of 20% above the average number of containers in the peak season. Every container has a time window of 48 hours and 95% of the containers should be delivered within this time window. Here, every container request is defined by a time window with a start and end time, an origin and destination, and a container size and weight. We consider two container types, namely 20 and 40 ft containers. Trucks can transport either one 40 ft container or two 20 ft containers, where for the last it holds that they may weigh at most 37 tons together and their weights may differ at most 5 tons.

For the future 1M TEU model we assume fully automated terminals and operations. This means that there will be no drivers necessary to drive the trucks at the terminals and therewith no boarding time. Moreover, handling times for loading and unloading are assumed to be shorter due to automation. In addition, there will be a restriction on the number of ART terminal visits at peak hours to prevent congestion, but we assume infinite capacity at the terminal. Consequently, the full automation creates more favorable circumstances for platooning as the higher number of containers will require more trucks and the shorter terminal turn-around times will lead to relatively more time for driving. That is why, we test our platooning simulation model on this scenario and compare this with the results obtained in TBAs software called TIMESquare.

Due to safety and technical reasons there are certain speed restrictions on the CER, which is maximum of 30 km/h near the terminals and a maximum of 20 km/h on the terminals. On the remainder of the road, they have decided to set a maximum of 50 km/h. However, as the benefits of platooning increase with higher speeds, we test several maximum speeds to determine with which speed the potential of platooning will be the highest.

3.2 Fuel model

In order to be able to measure the total fuel consumption with and without platooning, we define the following fuel model. This model is derived from Liang (2014) and is based on Newton's second law of motion. Then, the fuel consumption f_c is modeled as

$$\begin{aligned} f_c &= \frac{1}{\rho_d c_e} \int \delta(s) \left(m_t v(s) \frac{dv}{ds} + \frac{1}{2} c_D A_a \rho_a v^2(s) \phi(d_r) + c_r m g \cos\theta + m g \sin\theta \right) ds && (1) \\ &= k_E \int \delta(s) F_{engine} ds && (2) \end{aligned}$$

where the effective mass m_t and the indicator function δ are defined as

$$m_t = m + \frac{J_w + i_g^2 i_f^2 \eta_g \eta_f J_e}{r_w^2} \tag{3}$$

$$\delta = \begin{cases} 1 & \text{if } F_{engine} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

This means that δ is set to 1 when the truck is driving, and 0 when it brakes as otherwise "free" fuel would be generated, which is not the case. Next to this, we define $v(s)$ as the vehicle speed at position s.

However, to simplify the model we assume that the acceleration and deceleration phases are negligible. The assumption of constant speed is common in the literature of platooning problems as is the case for Liang et al. (2013), Larson et al. (2016), and Boysen et al. (2018). Consequently, we introduce the formula presented by Liang et al. (2013), which calculates the fuel consumption for a given distance and speed.

$$f_c = \frac{1}{\rho_d c_e} v \Big(\frac{1}{2} \rho_a A_a c_D v^2 \phi_{ijk} + m g c_r \Big) T \tag{5}$$

Table 7 in Appendix C presents the definitions of the other parameters of equation (1) and defines the values of the parameters that are used in equation (5). These values and definitions are based on Liang (2014) and Verbruggen (2021). Note, that for the vehicle mass m we define L_s as the weight of the container(s) transported on the vehicle at position s.

The benefits of platooning are taken into account in equation (1) via the air drag reduction factor ϕ. Namely, the second term between the brackets in the equation represents the drag force, which depends on the air density, the cross-sectional area of the vehicle, the drag coefficient, and the driving speed. Platooning then causes the so called slipstream effect, which means that a reduction in the aerodynamic drag arises behind a driving truck when a different truck drives closely behind it and therewith decreases the resistant force on the following truck. Consequently, this results in a reduction in the drag force of the following truck and with that reduces the fuel consumption. Since the air resistance is proportional to the square of velocity, the reduction becomes even larger when trucks drive faster.

Equation (1) shows that the value of ϕ depends on the inter-vehicle distance d_r, for which in general it holds that the smaller the distance, the lower the fuel consumption. Bonnet & Fritz (2000) show that a distance of 10 meters is considered to be a good distance as the reduction in fuel consumption remains constant or increases only slightly for shorter distances. Several papers, such as Alam (2011) and Hussein & Rakha (2020), show that the reduction factor for following heavy duty trucks is approximately 0.68 for the given 10 meters. They note that it is difficult to give an accurate value as it is influenced by many factors and even external conditions such as the weather. For the leading trucks they state that the reduction factor takes almost negligible values. Therefore, for the remainder of this paper, we consider an inter-vehicle distance of 10 meters with an air drag reduction factor ϕ of 0.68 for following trucks and 1 for leading trucks.

4 Off-line solution approach

The platooning problem that we are considering routes trucks in order to pick up containers at different pickup and delivery locations, and aims to minimize fuel consumption by forming platoons while satisfying these requests. As described in the previous section, the container requests have a certain time window within which the container should be picked-up and delivered. As only 95% of the containers has to be delivered within this time window, we speak of soft time constraints. This results in a combination of the Pick-up and Delivery Problem with Time Windows (PDPTW) as given by Desrochers et al. (1987) and the Coordinated Platooning Problem (CPP) by Larson et al. (2016).

In the PDPTW a group of vehicles, trucks in our case, drive around to pickup and deliver containers at different pickup and delivery locations within a certain time window. All trucks start at a depot and eventually end at the depot. The trucks are then assigned to certain customers (containers) while minimizing the total distance driven by trucks and meeting the time constraints. Here, it is assumed that a container is picked up and delivered by the same truck, while respecting the truck's capacity.

The CPP, on the other hand, simultaneously routes trucks and decides which trucks should platoon in order to minimize the total fuel costs. The CPP assumes that every truck departs from a given origin and drives to a given destination, and is therefore not completely applicable to our problem in which a truck can visit several customers in a yet unknown order. Therefore, we combine the PDPTW and CPP in order to allow for multiple pickup and delivery locations while aiming for forming platoons and therewith reducing the total fuel costs. This problem will be from now on presented as the Pickup and Delivery Platooning Problem with Time Windows (PDPPTW).

4.1 Pre-processing

In Section 3 it was stated that trucks can either carry one 40ft container or two 20ft containers, given certain weight restrictions. In terms of the graph, this would mean that a truck could visit two nodes in order to pickup and transport two containers at the same time. As this would increase the size of the graph significantly and as predefined container couples are not likely to have noticeable negative effects, we introduce a mathematical model to determine container combinations and therewith reduce the number of arcs and nodes. We view this problem as a Maximum Matching Problem (MMP) which divides nodes in pairs and therewith aims to maximize the number of pairs.

We define the graph $G(V, E)$ where the set of nodes V contains all 20 ft containers. We then define a node $v \in V$ as (f, f', s_i, e_i, q_i), with pickup facility f and delivery facility f' in the set of facilities F, s_i the pickup time, e_i the delivery time, and q_i the weight in tons of container i. These nodes are then connected to each other by means of the edges in E. These edges are constructed following the weight restrictions as given in Section 3. For clarity, this means that an edge is constructed between two nodes if they have the same pickup and delivery facility, if their joint weight is at most 37 tons, and if the weights of the two containers differ at most 5 tons. Finally, there must be an overlap in their time windows. In order to ensure reasonable transport times we assume that the overlap must be at least four hours, or otherwise no edge is constructed between the two nodes.

We introduce the decision variable x_e, which equals 1 if edge e is part of the matching, and 0 otherwise. In case an edge e is part of the matching, it means that the two nodes (containers) adjacent to edge e form a pair. Further, parameter w_e is the weight related to this edge, for which we test two values.

The model is then defined as follows:

$$\max \quad \sum_{e \in E} w_e \cdot x_e \tag{6}$$

$$\text{s.t.} \quad \sum_{e \text{ adjacent to } v} x_e \leq 1 \qquad \forall v \in V \tag{7}$$

$$x_e \in \{0, 1\} \qquad \forall e \in E \tag{8}$$

The objective aims to maximize the number of matchings in case $w_e = 1$ and maximizes the time window overlap in case we set w_e equal to the overlap in minutes of the time windows of both nodes adjacent to e. Constraints (7) ensure that at most one edge may be incident to each node. Finally, constraints (8) set binary conditions.

4.2 Notation

We define the directed graph $G = (V, A)$. The set of nodes is given as $V = \{0\} \cup N^+ \cup N^- \cup \{n\}$, in which 0 and n correspond to the same artificial depot. The trucks depart via node 0 and return to the depot via node n (to enable a correct determination of the variables). The set N corresponds to the set of customers, or container requests. Each request $i \in N$ is represented by a pickup node $i^+ \in N^+$ and a delivery node $i^- \in N^-$. Consequently, we introduce the set of origins $N^+ = \{i^+|i \in N\}$ and the set of destinations $N^- = \{i^-|i \in N\}$. Subsequently, all nodes in V are defined as (f, s_i, e_i), in which f corresponds to a certain facility in the set of facilities F, and the pickup time s_i and delivery time e_i represent the corresponding time window $[s_i, e_i]$. Note that for pickup nodes we assume that $e_i = \infty$ as there is no time restriction on the pickups.

In order to connect the nodes with each other, we introduce two different type of arcs. Here, we assume that the trucks directly drive to the next pickup node even if its early, instead of driving to the parking area when it cannot be assigned directly. This results in the set of arcs $A = A_{delivery} \cup A_{pickup}$.

The set of delivery arcs is defined as $A_{delivery} = \{((f, s_i, e_i), (f', s_i, e_i)) \mid (f, s_i, e_i) \in N^+$, $(f', s_i, e_i) \in N^-, \forall f, f' \in F : f \neq f', \forall i \in N\}$. These arcs represent the delivery of container request i at its destination $i^- = N_i^-$ right after it is picked up from its origin $i^+ = N_i^+$. The costs $C_{i^+i^-}$ of traversing this arc correspond to the fuel costs based on the distance between the two nodes. For these type of arcs, we define l_{i^-} as the total unload time at i^- and $t_{i^+i^-}$ as the time to drive from i^+ to i^-.

The set of pickup arcs is given as $A_{pickup} = \{((f, s_i, e_i), (f', s_{i'}, e_{i'})) \mid (f, s_i, e_i) \in N^- \cup \{0\}$, $(f', s_{i'}, e_{i'}) \in N^+ \cup \{n\}, \forall f, f' \in F : f \neq f' \vee f = f', \forall i, i' \in N : i \neq i', s_i + l_{i^+} + t_{i^+i^-} + l_{i^-} + t_{i^-i'^+} + l_{i'^+} + t_{i'^+i'^-} + l_{i'^-} < e_{i'} + W\}$. These type of arcs represent picking up a container after the truck just delivered all its containers and is therefore empty. It either drives empty to a different terminal to pickup a container (then $f \neq f'$) or it remains at its current terminal to pickup a container ($f = f'$). Therefore these arcs flow from a delivery node $i^- \in N^-$ to a pickup node $i'^+ \in N^+$ of a different container request. These arcs are only created so that the container can be delivered with a maximum delay of W minutes. This is based on the start time of the previous container, the corresponding driving and loading times to deliver the previous and this container, assuming one can drive there immediately. In addition, we create these arcs to connect depot 0 with all pickup nodes $i'^+ \in N^+$ and to connect all delivery nodes $i'^- \in N^-$ with depot n. The costs correspond to the fuel costs $C_{i^-i'^+}$ for driving to the next node, which equals zero in case the truck stays at the same terminal. Here, we define $l_{i'^+}$ as the load time at i'^+, and $t_{i^-i'^+}$ as the time to drive from i^- to i'^+ (which is zero in case $f = f'$).

All previously explained sets and parameters are given in Table 1. Further, other sets, parameters, and variables are given that will be used in the formulation of Section 4.3.

Table 1: Sets, parameters, and variables with descriptions for the PDPPTW model

Name	Description
Sets	
A	Set of all arcs
$A_{delivery}$	Subset of A containing delivery arcs
A_{pickup}	Subset of A containing pickup arcs
F	Set of facilities
K	Set of all trucks
N	Set of all container requests
N^+	Set of all pickup nodes of the container requests
N^-	Set of all delivery nodes of the container requests
V	Set of all nodes
Parameters	
e_i	The end time of the time window of container i
η	Constant air drag reduction factor
l_i	The (un)load time at node i
M	Big M, large positive penalty constant
s_i	The start time of the time window of container i
t_{ij}	Travel time from node i to j
W	Maximum allowed delay in minutes after delivery time
Variables	
A_{ik}	Time that truck k arrives at node $i \in V$
C_{ijk}	Fuel consumption of truck $k \in K$ to traverse edge $(i, j) \in E$
D_{ik}	Time that truck k departures from node $i \in V$
p_{ijk}	Binary decision variable set to 1 if truck k is a follower in a platoon along edge $(i, j) \in A$
ϕ_{ijk}	Reduction factor that is set to 0.68 if truck k is a follower in a platoon between node i and $j \in V$, and 1 otherwise
$q_{k,m,i,h}$	Binary decision variable set to 1 if truck k from node $i \in V$ follows truck m from node $h \in V$ given that $f_i = f_h$, and that they drive to the same destination, and 0 otherwise
v_i	Binary decision variable set to 1 if container request $i \in N$ is not delivered within its time window, and 0 otherwise
w_{ik}	Time that truck k waits for other trucks to form a platoon at node $i \in N$
x_{ijk}	Binary decision variable set to 1 if vehicle k traverses edge $(i, j) \in A$
y_{ik}	Binary decision variable set to 1 if vehicle k arrives at node $i \in V$ after the start of corresponding time window e_i, and 0 otherwise
z_{ik}	Time that truck k waits for the start of the time window at node $i \in N$

Figure 1 gives a visual representation of the set-up of the graph. This example shows that only a delivery arc flows between the pickup and delivery node of the same container request. Then out of delivery nodes several pickup arcs flow to different pickup nodes. These pickup nodes can either have the same or a different facility as location. Note that delivery nodes are also connected to pickup nodes with a time window after the time window of the delivery node. This means that trucks can arrive too early at the terminal and therefore have to wait till the start of the time window before it can start loading.

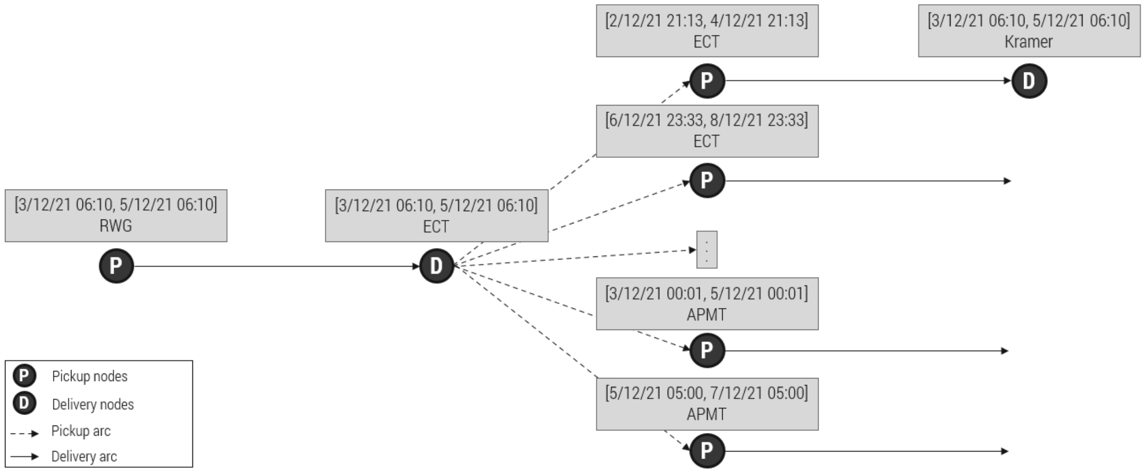


Figure 1: Example of the buildup of the graph.

Zooming into the graph, Figure 2 presents which events occur between two nodes along an arc. Underneath the arc we introduce different parameters and variables that will be used in the formulation, and give a short definition above the arc (for more details view Table 1). An arc between two nodes thus starts with the departure at the start node and ends at the departure from the end node. A truck k departs from node i in the direction of node j at time D_{ik}, after which it arrives at j at time A_{jk} after driving for t_{ij} minutes. In case we consider a container pickup and the time window of j has not started yet, then the truck will wait for z_{jk} minutes till the start s_j. Then the unloading in case of a delivery arc and loading in case of a pickup arc starts. After this the truck waits for w_{jk} minutes for other trucks to drive as a platoon to the next location.

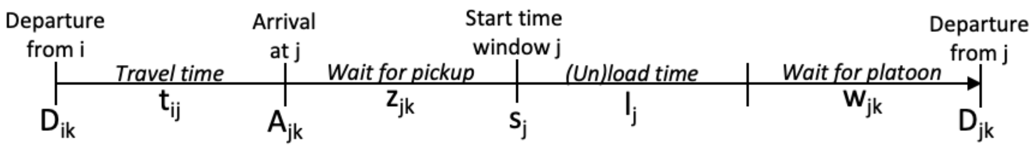


Figure 2: Visual representation of the events that occur along an arc.

4.3 Formulation

We now present the formulation for the PDPPTW. The objective aims to minimize the total fuel consumption. Given constant speed we assume constant fuel costs on roads based on their distance as derived from equation (1). The reduction factor ϕ_{ijk} is then set to 0.68 (given $\eta = 0.32$) if the truck is a follower in a platoon. The full price is charged in case the truck is a leader.

$$\min \quad \sum_{k \in K} \sum_{(i,j) \in A} C_{ijk} \tag{9}$$

Here we define C_{ijk} and ϕ_{ijk} as

$$C_{ijk} = \frac{1}{\rho_d c_e} v\Big(\frac{1}{2}\rho_a A_a c_D v^2 \phi_{ijk} + mgc_r x_{ijk}\Big) t_{ij} \tag{10}$$

$$\phi_{ijk} = x_{ijk} - \eta \cdot p_{ijk} \tag{11}$$

To preserve feasible truck routes, we define constraints (12)-(15). Constraints (12) ensure that every request is serviced by exactly one truck. Following constraints (13) and (14), a truck must leave and return to the depot once. Finally, flow constraints (15) require that a truck must also leave a node if the truck visits that node.

$$\sum_{j \in V\backslash\{0\}} \sum_{k \in K} x_{ijk} = 1 \qquad \forall i \in V\backslash\{0, n\} \tag{12}$$

$$\sum_{j \in V\backslash\{0,n\}} x_{0jk} = 1 \qquad \forall k \in K \tag{13}$$

$$\sum_{i \in V\backslash\{0,n\}} x_{ink} = 1 \qquad \forall k \in K \tag{14}$$

$$\sum_{i \in V} x_{ihk} - \sum_{j \in V} x_{hjk} = 0 \qquad \forall k \in K; h \in V\backslash\{0, n\} \tag{15}$$

Constraints (16)-(19) preserve a correct determination of the time. Constraints (16) determine the arrival time at j based on the departure time at i and the travel time from i to j. In case a truck arrives before the pickup time of the container at the terminal, the truck will have to wait some time which is determined by constraints (17). We introduce constraints (18) in order to enable the trucks to form a platoon. It defines a waiting time before the truck departs to its next destination. The linearizations of the previous constraints are given in Appendix A. Finally, constraints (19) state that the variables can only be larger than zero if a truck visits the node.

$$x_{ijk} \cdot (D_{ik} + t_{ij} - A_{jk}) = 0 \qquad \forall (i, j) \in A; k \in K \tag{16}$$

$$\sum_{(i,j) \in A} x_{ijk} \cdot (\max\{0, s_j - A_{jk}\} - z_{jk}) = 0 \qquad \forall j \in N^+; k \in K \tag{17}$$

$$x_{ijk} \cdot (D_{jk} - A_{jk} - l_j - z_{jk} - w_{jk}) = 0 \qquad \forall (i, j) \in A; k \in K \tag{18}$$

$$A_{jk} + D_{jk} + w_{jk} + z_{jk} \leq M \cdot \sum_{(i,j) \in A} x_{ijk} \qquad \forall j \in V; k \in K \tag{19}$$

Constraint set (20)-(22) sets the time constraints on the container requests. Constraints (20) ensure that container i can only be picked up after its pickup time. Constraints (21) form a soft constraint on the delivery of container i before its delivery time. Note that the delay can be at most W minutes after the deadline e_j. Subsequently, constraints (22) require that at least 95%

of the containers should be delivered within its time window.

$$D_{ik} \geq s_i \cdot \sum_{(i,j)\in A_{delivery}} x_{ijk} \qquad \forall i \in N^+; k \in K \tag{20}$$

$$D_{jk} - w_{jk} \leq e_j + W \cdot v_j \qquad \forall j \in N^-; k \in K \tag{21}$$

$$\sum_{i\in N^-} v_i \leq 0.05 \cdot |N^-| \tag{22}$$

The following constraints capture the platoon formation. Constraints (23) and (24) ensure that two trucks form a platoon if their departure times are equal. Constraints (25) state that each truck can at most have one follower (directly behind it). Next to this, constraints (26) require that the trucks can only platoon if the leader and followers respectively, visit the given nodes. Further, constraints (27) and (28) define whether a truck platoons on a given arc.

$$D_{ik} - D_{hm} \leq M(1 - q_{k,m,i,h}) \qquad \forall k,m \in K : k > m; \; i,h \in V : f_i = f_h, i \neq h \tag{23}$$

$$D_{ik} - D_{hm} \geq -M(1 - q_{k,m,i,h}) \qquad \forall k,m \in K : k > m; \; i,h \in V : f_i = f_h, i \neq h \tag{24}$$

$$\sum_{k\in K}\sum_{i\in V} q_{k,m,i,h} \leq 1 \qquad \forall m \in K : k > m; \; h \in V : f_i = f_h, i \neq h \tag{25}$$

$$q_{k,m,i,h} \leq 0.5 \cdot \left(\sum_{(i,j)\in A} x_{ijk} + \sum_{(h,l)\in A} x_{hlm} \right) \qquad \forall k,m \in K : k > m; \; i,h \in V : f_i = f_h, i \neq h \tag{26}$$

$$p_{ijk} \leq x_{ijk} \qquad \forall (i,j) \in A; k \in K \tag{27}$$

$$p_{ijk} \leq \sum_{m\in K}\sum_{h\in V} q_{k,m,i,h} \qquad \forall (i,j) \in A : f_i = f_h, i \neq h; \; k \in K : k > m \tag{28}$$

Finally, the domains of the variables are defined. Constraints (29), (30), (31), (33), (36), and (39) set non-negative conditions. Further, constraints (34), (35), and (38) set binary conditions.

$$A_{ik} \geq 0 \qquad \forall i \in V; k \in K \tag{29}$$

$$C_{ijk} \geq 0 \qquad \forall (i,j) \in A; k \in K \tag{30}$$

$$D_{ik} \geq 0 \qquad \forall i \in V; k \in K \tag{31}$$

$$p_{ijk} \in \{0,1\} \qquad \forall (i,j) \in A; k \in K \tag{32}$$

$$\phi_{ijk} \geq 0 \qquad \forall (i,j) \in A; k \in K \tag{33}$$

$$q_{k,m,i,h} \in \{0,1\} \qquad \forall k,m \in K : m < k; i,h \in V : f_i = f_h \tag{34}$$

$$v_i \in \{0,1\} \qquad \forall i \in N^- \tag{35}$$

$$w_{ik} \geq 0 \qquad \forall i \in V; k \in K \tag{36}$$

$$x_{ijk} \in \{0,1\} \qquad \forall (i,j) \in A; k \in K \tag{37}$$

$$y_{ik} \in \{0,1\} \qquad \forall i \in V; k \in K \tag{38}$$

$$z_{ik} \geq 0 \qquad \forall i \in V; k \in K \tag{39}$$

5 On-line solution approach

In this section we introduce three solution approaches for the on-line problem, which are implemented and tested on the CER network (with all facilities and containers) by means of simulation. A fixed number of trucks start at the parking area at the beginning of the simulation. We then distinguish three types of events which provoke a certain decision or action. Firstly, whenever a truck delivers its current assignment it is said to be idle and can therefore be assigned to a new container if available. This is the so called container assignment process. Secondly, over the time horizon new batches of containers arrive at the facilities. Consequently, we define the truck assignment procedure in which we search for an idle truck every time a new container becomes available for pickup. Finally, we set certain requirements which state when a platoon is allowed to depart to its destination. Every time a truck joins a platoon at a certain facility we make a decision on whether the considered platoon is allowed to leave or not based on these platoon formation requirements.

As already stated, we make use of two different types of simulations to test the three approaches: a more basic simulation in Java and an extensive simulation in TBAs software. In Section 5.1 and 5.2 we discuss these two simulations respectively and discuss the assumptions which distinguish them from each other. We then go into more detail on the three platoon formation approaches that are tested and the decisions that are made following their corresponding events. In addition, we will describe the structure of the simulation in Java for every approach. Correspondingly, Section 5.3 presents the two ad hoc approaches and Section 5.4 presents the group based approach. Finally, we present in Section 5.5 the base model in which we do not platoon and which can therefore be used to measure the potential of platooning against.

5.1 Java simulation assumptions

The Java simulation is an event based simulation which is built up from scratch by introducing different types of events. These events are based on the previously described events, but may differ slightly for every approach. The simulation itself is considered to be a basic simulation in the sense that we make several assumptions which simplify reality. Consequently, this simulation is mainly used to test and tune the models to determine the best parameter values for each approach assuming a fixed number of trucks. This will give an indication on which parameters should be tested in TBAs simulation model.

For a start, we assume a constant speed over the entire planning horizon, meaning that a truck drives at the same speed the entire way from facility to facility. Consequently, the driving times between the facilities are fixed and an underestimation of the real driving times. Obviously, it is not likely that a truck will constantly drive one single speed due to the acceleration and deceleration period at the terminals, not to mention the speed variability due to congestion at junctions. Consequently, the fuel consumption generated by this simulation is likely to be an overestimation of the real world fuel consumption as the speed is expected to be higher than in reality. On the other hand, congestion does lead to unnecessary accelerating and braking, which increases the fuel consumption as well. However, as we do not take the influence of acceleration and deceleration into account in this thesis, it is expected that the fuel consumption of the Java model is higher than the fuel consumption generated in TIMESquare. Moreover, we assume fixed driving and (un)loading times at the terminals. Following, we do not take the distance driven on the terminal into account for the total fuel consumption, but only use the terminal driving times to correctly schedule the next event.

Another assumption is made regarding the assignment of trucks. It might for instance occur that a batch of a hundred containers arrives at a certain facility. Our simulation could assign a truck to every container at once in case there are currently a hundred trucks available, even though it is not likely that a facility has space to (un)load that many trucks at once. We also use the pre-processing procedure of Section 4.1 to combine the containers already in advance, because of which one container arrival can exist of two 20ft containers. Finally, we assume that the trucks carry an infinite amount of fuel because of which the trucks never have to refuel.

In Section 5.3.4 and 5.4.3 we elaborate on the structure of the simulation model for the considered approach.

5.2 TIMESquare simulation assumptions

TBAs software is also an event based simulation but it is more advanced in the sense that it aims to simulate reality in a near real time manner. As mentioned, TBA already built a simulation for the CER model for separately driving trucks. That model makes use of ad hoc assignment, meaning that a truck is immediately assigned to a new container (if available) when it finishes its current assignment and a newly arrived container is immediately assigned to an idle truck if possible. The container assignment procedure determines the container that is closest to the current location of the truck, aims to assign two 20ft containers together, and it takes into account whether or not a container is within four hours from its deadline. The truck will drive back to the parking area in case the truck cannot be assigned to a container. A newly arrived container in turn is assigned to the idle truck that is closest to its pickup location.

This model is adapted by implementing new container assignment and platoon formation decisions for each approach. The truck assignment that we use is already the same as in TIMESquare and is therefore not changed. For the platoon formation process we provide the model with the parameters and schedule the departure in a similar way as for the Java simulation. The container assignment in TIMESquare is done by choosing the container with the lowest score, where the score is mainly based on the driving time to the pickup location and the slack time till the deadline. This score is adjusted by subtracting the remaining waiting time of the corresponding platoon, such that the platoon with the longest time is favored.

The vehicles drive in a real life manner by means of braking whenever a destination is reached, an obstacle is encountered or to turn a corner. Similarly, acceleration is applied in case a truck departs from a certain location or to increase the speed on straight roads. The loading and unloading is simulated in a similar real time manner, causing different loading and unloading times for every terminal visit. On both the road and the terminal, fuel Equation 5 is called every half a second and uses the current speed, weight, and platoon position of the vehicle to determine the fuel that was consumed during that time. This generates a more reliable and accurate approximation of the total fuel consumption than the Java model. However, equivalent to the Java model, it is assumed that trucks have an infinite amount of fuel and therefore never have to refuel. Finally, to take the maximum capacity of the loading process into account, one cannot assign trucks in such a way that more trucks are on their way to the terminal than there are spaces currently on the terminal. We then look for a container at a different terminal with capacity left or drive back to the parking area in case all terminals are fully occupied.

5.3 Ad hoc approach

The following two ad hoc approaches that are explained, are based on the idea that a decision has to be made every time a truck finishes its current assignment or at the occasion that a new container is available for pickup. The two methods are formed by combining the container assignment algorithm of Section 5.3.1 and the truck assignment of Section 5.3.2 with one of the two platoon formation methods of Section 5.3.3. The container assignment algorithm is then used to assign trucks to a new job in a smart way, and the truck assignment algorithm is designed to assign them the other way around, namely container to truck. Simultaneously, one of the platoon formation methods is used to determine when platoons should leave.

5.3.1 Container assignment

The container assignment heuristic is called whenever a truck finishes its current assignment and it aims to determine a new assignment for the truck (if available). We assume that a truck has finished its assignment whenever the truck has successfully delivered and unloaded its assigned container(s) at the corresponding destination. Over the horizon of the simulation, container batches will arrive at the different terminals waiting to be picked up and delivered at the correct facility. We assume that at each facility a separate platoon is formed for every destination. Figure 3 presents a visual representation of this, where you can see the waiting platoons on the left. It shows that a platoon is formed in the direction of both B and C separately even though the trucks in the direction of C could have platooned with the others for a large part of the road till facility B. Each platoon will remain waiting for a certain time (as presented on the right of the platoons).

Instead of just looking at the distance and the deadline of the available containers, we now also take the platoons into account. The algorithm starts by checking if one of the containers is within a certain time period from its deadline, meaning that it has priority over the other containers in order to be able to deliver it on time or limit the delay of the delivery. In case such containers exist, the truck is assigned to the container that has the highest priority. Given the urgency of this container, the corresponding platoon will leave immediately when the truck joins in order to fasten the delivery.

There are three possible situations in case no priority containers exist: there are containers waiting to be picked up at the current facility, there are only containers available at different locations or there are no containers at all waiting to be picked up. We prefer full driving over empty driving, as driving empty leads to extra fuel consumption and therewith reduces the benefits of platooning.

Consequently, the algorithm first evaluates the first case. We start by determining the number of trucks that is assigned to the platoon (i.e. the trucks that are on their way from the stack to the platoon and the ones that already joined the platoon). For the platoons for which this number is lower than the maximum platoon length, we determine the remaining waiting time of that platoon. Subsequently, we define which container destination has currently the highest waiting time. The idea is that the longer the platoon waits the higher the chance that the truck can join this platoon. There is still some time required to load the container on the truck, therefore it will take some time before the truck will join the platoon. Moreover, if the number of trucks that already joined the platoon plus the ones that are on their way to the platoon are equal to the maximum platoon length, then you know for sure that the truck will not join the platoon. Given the destination, it is decided which container with that destination is currently nearing its deadline the most and then that one is assigned to the truck. If there are no platoons waiting, we simply choose the container with the closest deadline.

Figure 3 presents a truck which just unloaded a container at facility A and is now asking for a new assignment. As we prefer full over empty driving, the truck can choose between the three containers that are waiting to be picked up at facility A. There is one container available that has to be delivered at B and two that have to be delivered at C. There are platoons waiting that are heading to both terminals, but the platoon into direction C is expected to wait longer than the platoon to B. Therefore the truck will transport a container for facility C. Now it can choose between two containers, but the highest container has a deadline that is closer than the bottom container. Therefore, the algorithm decides that this truck will transport the container with destination C and time window [01/10/21 08:10, 03/10/21 08:10].

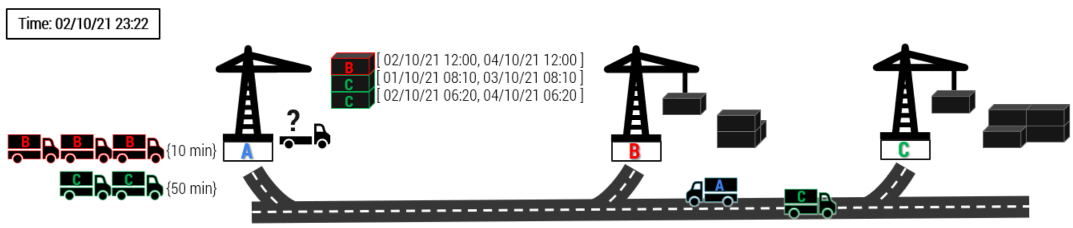


Figure 3: Example for the scenario that containers are available for pickup at the current terminal.

In case there are only containers available at other locations, we focus on minimizing the number of kilometers driven empty. We determine which location is closest to the current location of the truck and consider the containers that are waiting to be picked up at that location. Then the container with the closest deadline is assigned to the truck. It can be concluded that we do not take the platoons into account for this type of assignment, because we think that the time till joining the platoon at the the pickup location is too long. This is because it is likely that the relevant platoon at the pickup location has already left once the truck arrives, as this truck first has to wait at its current terminal for the platoon to depart, then has to drive to the pickup terminal, and still needs time to load the container before it joins the platoon.

Considering Figure 4, the truck would join the platoon in the direction of facility B as there are no containers at A left. Again, in case there would be no platoons waiting at A, the truck would simply be the leader of the platoon in the direction of B. The truck will therefore be assigned to pickup the container into direction C with time window [01/10/21 08:10, 03/10/21

08:10] as that container is closest to its deadline of all containers at terminal B. This means that another truck cannot pick this container up anymore, but will be picked up by this truck.

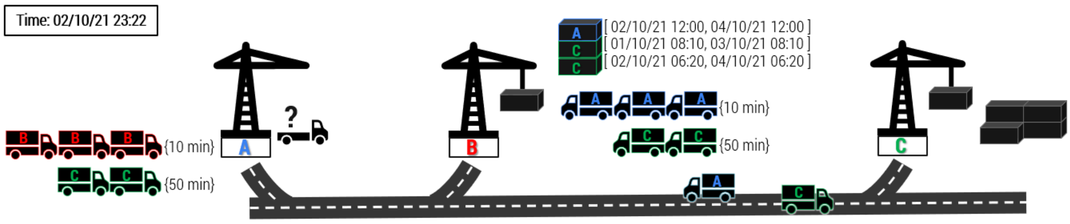


Figure 4: Example for the scenario that there are no containers available for pickup at the current terminal.

Note that in the case that there are no containers waiting to be picked up at all, then the idle trucks will drive back to the parking area where they will wait until they are assigned to a new container.

This process is formalized in Algorithm 1 in Appendix B.

5.3.2 Truck assignment

It has been explained that batches of containers appear at different facilities over the planning horizon. The truck assignment decision is based on the idea that whenever a batch becomes available for pickup at a terminal, we search for an idle truck for every container of that batch and assign the container. As we send trucks without an assignment back to the parking area, this means that idle trucks are always at the parking area in this case.

5.3.3 Platoon formation

In order to enable platoon formation we have decided to let trucks wait at the terminal for a certain amount of time for other trucks to join. We determine two type of approaches to define the amount of time that the trucks are allowed to wait. In Section 5.3.3.1 we define a fixed approach, whereas in Section 5.3.3.2 we introduce a more flexible approach.

5.3.3.1 Fixed approach

Literature research shows that several papers conclude that a certain platoon length or waiting time gives better results, and that these values are strongly related to each other. We test different platoon length and waiting time combinations to find the best combination. In practice this means that platoons are allowed to depart when the maximum length or waiting time that we set is reached. Moreover, note that the platoon will also leave if the container of one of the trucks reaches its priority state (4 hours before its deadline).

The process of checking whether a platoon satisfies the requirements to depart is given in Algorithm 2 in Appendix B.

5.3.3.2 Adjusted approach

The second approach is an approach in which the departure moment of the platoon is completely determined by a certain waiting time (meaning that there is no maximum platoon length here). The idea is derived from Cigolini et al. (2002), who propose an on-line scheduling procedure for batching machines. A batching machine processes multiple jobs simultaneously and cannot accept arriving jobs when it is processing a batch of jobs. Whenever a batching machine finishes its process, it has to be decided if the waiting batch matches the capacity. In case it is less than the capacity, it is possible to wait for other jobs to increase the load. This results in a trade-off between capacity utilization and waiting time. This is closely related to the formation of platoons. The longer the platoon waits the higher the savings, as the platoon length will increase, but the longer the platoon waits the closer the containers will get to their deadlines.

The authors have generated a formula which defines that the sum of delays (or waiting times in our case) of all jobs is equal to the processing time. A common approach is to set the maximum waiting time equal to k times the processing time of the machine. Cigolini et al. (2002) adapt this approach by taking future arrivals into account. They propose the following formula:

$$[T_{dep} - T_{prev}]B_0 + \sum_{i=B_0+1}^{B_n-1} [T_{dep} - t_i] + [T_{dep} - t_n] = k \cdot PT \tag{40}$$

In our terms, we define T_{dep} as the departure time of the platoon and therewith the to be determined variable. Then T_{prev} is the time at which the previous platoon departed. The number of trucks that are waiting at the time the previous platoon departed is given by B_0. Normally a machine is started and during its process time new jobs arrive that will be loaded when the machine finishes its current batch. However, there is no processing time in case of platoons as a new platoon can be formed immediately when the previous platoon leaves. Therefore B_0 will be equal to 0, because of which the first part of the equation cancels. Then B_n defines the number of jobs in the queue when the nth job arrives and t_i the arrival time of the ith job. After all, the first part determines the delay of the trucks that were already in the queue when the previous platoon departed. The second term defines the delay of the arrivals after the previous platoon left and before the current arrival. Finally, the third term determines the delay of the current arrival. Therefore, every time a new job arrives, the trade-off is reviewed and a new departure time is determined. Note, again, that the platoon will also leave immediately if the container of one of the trucks reaches its priority state before the scheduled departure time. Cigolini et al. (2002) assume $k = 1$, but to make it more realistic to our problem and as we do not have a processing time, we will test several values for the term $k \cdot PT$.

Figure 5: Example of formula 40

Figure 5 gives two examples of job arrivals, given that the previous platoon left at time 0.

As noted, the formula is reviewed every time an arrival occurs. For this example we assume that $k = 1$ and $PT = 3$. Starting by the left timeline, the first arrival occurs at $t_1 = 1$. For the first arrival, the first two terms cancel such that $[T_{dep} - 1] = 3$, which results in $T_{dep} = 4$. This means that, if no new truck arrives in the meantime, the platoon will leave at $t = 4$. However, a second truck arrives at $t = 3$, which gives us the formula $[T_{dep} - 1] + [T_{dep} - 3] = 3$ and therewith $T_{dep} = 3.5$. Now, instead of departing at $t = 4$ the platoon will depart at $t = 3.5$. This shows that every time a truck arrives, the departure time of the platoon is brought forward to reduce the delay of the previous arrivals. The next arrival occurs at $t = 4$ which is after T_{dep} and therefore will not be part of that platoon, but will be the leader of the new platoon. Therefore, the formula is renewed by setting $T_{prev} = 3.5$.

We now review the second timeline of Figure 5, where again the first arrival occurs at $t = 1$ so that $T_{dep} = 4$. In this case however, the second arrival already appears at $t = 1.2$ so that $[T_{dep} - 1] + [T_{dep} - 1.2] = 3$ and therefore $T_{dep} = 2.6$. Note that this time T_{dep} is much smaller after the second arrival than for the left timeline. The idea behind this is that there was a large gap between the first and second arrival in the first example, because of which the second arrival is close to the departure time and therefore does not have a large delay. Next to this, because of this large gap you expect the next arrival to have a similar large gap, so you do not want to lower T_{dep} too much, hoping that the next arrival will fit in. For the second timeline on the other hand, the second arrival occurred quickly after the first one, which results in large delays for both jobs if T_{dep} is not lowered. Moreover, based on the previous arrival you expect the next arrival also to appear soon and therefore see no need for long waiting times till T_{dep}. This clearly shows the trade-off between capacity utilization and waiting time, as for the second timeline long waiting times do not seem necessary as capacity utilization seems to increase at a faster rate than for the first timeline.

Finishing the example, the third arrival appears at $t = 2$ which is before the current T_{dep} such that $[T_{dep} - 1] + [T_{dep} - 1.2] + [T_{dep} - 2] = 3$ and $T_{dep} = 2.4$. The first three trucks will then platoon and depart at $t = 2.4$ as no new arrivals occur in the meantime. After all, this shows the strength of the formula in using arrival forecasts to determine the departure time of the platoon.

The procedure for determining the new departure time of the platoon every time a truck joins a platoon is presented as Algorithm 3 in Appendix B.

5.3.4 Simulation structure

The simulation is modeled in Java and is built up out of a list of events. For the ad hoc approach, we distinguish four different events that represent different situations. The simulation starts at time zero from which the time passes by jumping from timestamp to timestamp of the given events, until the end of the planning horizon is reached. We explain them in further detail below.

Arrival event:

The Arrival event represents the arrival of a (new) container at a certain facility. This triggers the truck assignment of Section 5.3.2 and therewith aims to assign this container to an idle truck (at the parking area). In case of a successful assignment, this truck will immediately join the platoon at the parking area with the pickup location as destination. For this we do not schedule a Join Platoon event, but immediately check if the requirements for platoon departure, as described in Section 5.3.3, are met.

Depart event:

The Depart event is called when the requirements for departure (Section 5.3.3) are met because of which the platoon departs. The following step differs depending on whether the truck is in a platoon to deliver its container or to pickup its container. For loaded trucks in the platoon we schedule a Ready event after the driving and unloading time, as these trucks are on their way to deliver the container and are therefore idle after delivery. For empty trucks a Join Platoon event is planned after the driving and loading time, as these trucks are on their way to pickup a container because of which their next event will be the moment they join the platoon at the pickup location to deliver the container at its destination. For the idle trucks in the direction of the parking area, we schedule a Ready event after the time needed to drive to the parking area.

Join Platoon event:

The Join Platoon event represents the moment that a truck joins a platoon. Consequently, it must be checked whether the platoon requirements are met as a result of this arrival. Depending on the platoon formation strategy, the requirements are checked by Algorithm 2 for the fixed platoon formation of Section 5.3.3.1 and Algorithm 3 is used for the adjusted approach of Section 5.3.3.2. The platoon then either departs directly and creates an event similar as is done in the Depart event, or the Depart event is rescheduled to the new departure time (in case of the adjusted approach) or nothing happens.

Ready event:

The Ready event occurs after a truck has unloaded its container at the correct delivery location and is therefore now idle and available for a new assignment. By means of the container assignment process as described in Section 5.3.1, we determine a new assignment (if available). After a successful assignment to a container at its current location, we schedule a Join Platoon event after the related loading time, as the truck will join the platoon to the delivery location once it has loaded the container. If the truck is assigned to a container at a different location, then the truck will immediately join the platoon into the direction of the pickup location. For this we do not schedule a Join Platoon event but immediately check if the departure requirements for the platoon are met. Finally, if there are currently no containers available then we make the truck join the platoon in the direction of the parking area immediately and check again the departure requirements.

To summarize the previously described events and their relation, we present in Figure 6 for every event the events that can be constructed when the considered event occurs by means of the dashed arcs. As it might occur that several event types are scheduled at the same timestamp and as they are closely connected to each other, we order them based on logic as shown in Figure 6 by means of the solid arcs.

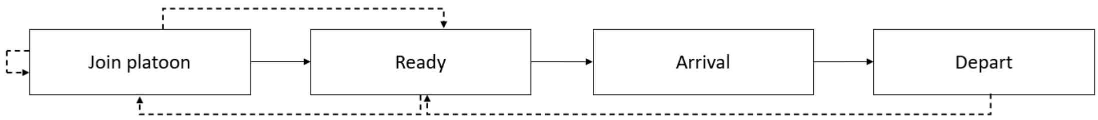


Figure 6: The order and relation of the simulated events of the ad hoc approach.

The Join Platoon event is always viewed as first to make sure that the truck can still join the platoon if it has not left yet. If Depart would be called first, the platoon would leave without the truck even though the truck was ready to join. Consequently, this maximizes the platoon length. Next to this, Ready and Arrival both generate Join Platoon events. Ready bases the container assignment choice on the waiting time of the platoon, if Join Platoon is viewed afterwards and makes the platoon depart this might mean that the platoon leaves before the truck joined the platoon and therefore it might have been better to choose a different container. Then after Join Platoon, we view Ready before Arrival to encourage the truck to choose from the already available containers. Finally, we simulate Arrival before Depart as Arrival can generate a Join Platoon which then could be added to the platoon before it departs with Depart.

5.4 Group approach

The group approach is based on the idea that trucks drive in fixed groups for the entire planning horizon. We assume that the groups are predefined and formed at the start of the simulation at the parking area, by simply defining groups of the given platoon size. It might occur that the trucks cannot be divided into groups of exactly the considered group size, then the remaining trucks simply form a group. The method consists of a container assignment and truck assignment decision, which are described in Section 5.4.1 and 5.4.2 respectively.

5.4.1 Container assignment

In the beginning of the simulation all trucks start at the parking area, which enables trucks to drive together already from the beginning. This gives rise to the idea to use group assignment instead of individual assignment. This means that whenever a group of trucks finishes its current assignments, the group will be assigned to the closest batch of containers with the same origin and destination. They leave the terminal when every truck is loaded and will be assigned to a new assignment when all trucks unloaded the containers at the destination. This way, trucks start platooning from the start of the simulation and continue driving in a platoon for the entire planning horizon. The platoon will not be assigned to a batch of containers in case the available batches are smaller than the number of trucks in a platoon. Then, the trucks will drive back to the parking area to wait until a batch is large enough. Only if the deadline of a container in a batch is nearing, the platoon will pick them up anyways (causing one or more trucks to drive empty). We test different platoon sizes in order to determine which size generates the highest potential for platooning.

5.4.2 Truck assignment

Similarly to the container assignment, we assign several containers to a platoon at once. Whenever a new batch of containers arrives at a facility, we check whether there are now enough containers with the same origin and destination to load all the trucks of one of the idle platoons. Given there are sufficient containers, we assign the containers that are closest to their deadline to one of the idle platoons at the parking area.

5.4.3 Simulation structure

The group simulation model has different events as it has different triggers than the ad hoc approach. Again we start at time zero and flow through time following the timestamps of the scheduled events until all containers are delivered. We distinguish three event types and explain them in more detail in the remainder of this section.

Arrival event:

The Arrival event represents the arrival of a (new) container at a certain facility. This triggers the truck assignment algorithm of Section 5.4.2. Consequently, the model aims to assign the now available containers (with the same origin and destination) to an idle platoon. In case of a successful assignment, we schedule a Ready event at the time at which the last truck has unloaded its container(s) at the destination (as this might differ among the trucks based on the number of containers they transport). On the other hand, when the available containers cannot be assigned to a platoon, then a Deadline event is scheduled for the new container at the latest time at which the container has to be picked up (four hours before the deadline).

Deadline event:

The Deadline event relates to the occurrence that a container reaches its priority state, meaning that it has to be picked up right away to be delivered before its deadline. We check if an idle platoon exists and assign the containers with the same origin and destination of the idle container to the idle platoon that is closest to the pickup location of the containers. Consequently, a Ready event is scheduled at the time at which the last truck has unloaded its container at the destination.

Ready event:

The Ready event is called when all trucks in a group have finished their current assignment and are therefore now in an idle state. This asks for the container assignment procedure as discussed in Section 5.4.1. When a group could be assigned to a batch of containers, we schedule a Ready event at the time at which the last truck has unloaded its container. Otherwise we send the platoon back to the parking area and schedule a Ready event after the corresponding driving time.

Once more we summarize the relation between the previously described events by visualizing it by means of Figure 7. For each event we present which other events it can trigger via the dashed arcs. The solid arcs in turn denote the order in which the events are scheduled given that those occur at the same timestamp.

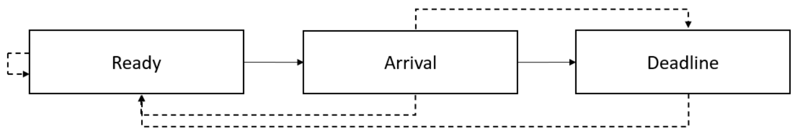


Figure 7: The order and relation of the simulated events of the group approach

Ready is viewed first to encourage the trucks to choose containers that are already in the system. In case there are not enough, the Arrival event can still trigger the group to depart if there are enough containers after this arrival. Finally, Deadline is considered last to make sure that all containers that arrive at the same time as the deadline are in the system and therewith maximize the number of containers on the trucks when the group is pushed to leave by the deadline.

5.5 Base model

Finally, we simulate the base case in which trucks drive separately and do not aim to form a platoon. This way, we can determine the benefits generated by the three platooning approaches. In Section 5.5.1 we explain the container assignment process and we discuss the truck assignment in Section 5.5.2. Afterwards, we define the structure of the simulation model in Section 5.5.3.

5.5.1 Container assignment

We implement a similar approach as the ad hoc approach, but neglect the part that regards the waiting time of platoons. This means that whenever a truck finishes its current assignment we determine if there are containers available for pickup. If so, we first check if one of those has reached its priority state and assign that one. Otherwise, assign the container that is closest to the current location of the truck and is closest to its deadline. The truck drives back to the parking area in case there are currently no containers in the system that are ready for pickup.

5.5.2 Truck assignment

The procedure is defined in the exact same way as for the previously described ad hoc approaches. Whenever a batch becomes available for pickup at a facility, we search for an idle truck for every container of that batch and assign the container to that truck.

5.5.3 Simulation structure

For this base model we only define two types of events. Similar to the other approaches, the simulation starts at time zero and moves forward by means of chronologically following the timestamps of the scheduled events.

Arrival event:

The Arrival event represents the arrival of a (new) container at a certain facility. Consequently, the truck assignment process of Section 5.5.2 is executed. Given an idle truck is found, we schedule a Ready event after the time required to pickup the container and deliver it at its destination (i.e. sum of the driving, loading and unloading time).

Ready event:

The Ready event corresponds to the successful delivery of a container by a truck. We then determine a new assignment for the truck by means of the container assignment of Section 5.5.1 and if found, we schedule a Ready event at the time at which the truck has unloaded the container at its destination. Otherwise, the truck is sent back to the parking area and a Ready event is created afterwards.

To conclude this section, we visualize the events that can be triggered by each event by means of the dashed arcs. Finally, the solid arc shows that we view the Ready event before the Arrival event whenever these two events are scheduled at the same timestamp (like the previous approaches).

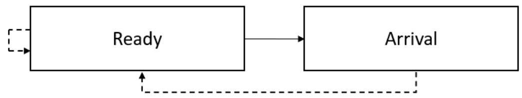


Figure 8: The order and relation of the simulated events of the base model

6 Computational study

In this section we present and analyze various types of results generated by different mathematical methods and approaches. For this research we have used Java 2020-12 and IBM CPLEX version 12.6.3, and performed our runs on a Intel® Core™ i7 2.70GHz with 16 GB. In Section 6.1 we give an overview on the data used for the mathematical models of Section 4 and the Java simulation of Section 5.

We start the analysis by reviewing the pre-processing of the data in Section 6.2. Thereafter, we evaluate the mathematical model and discuss the usefulness of it in Section 6.3. Then we look into the performance of the three approaches of Section 5.3 and 5.4 considering a Java implementation. Here, we first look into the predictability of the parameters that are chosen for the three approaches in Section 6.4.1. Thereafter the behavior of the three approaches is reviewed and compared with the case in which there is no platooning in Section 6.4.2. These results are then later used to define the parameter settings that are tested in TBAs software. Subsequently, in Section 6.4.3 a sensitivity analysis is performed on different speeds to determine the influence of the velocity on the number of trucks and the fuel consumption. Finally in Section 6.4.4, we decide on the influence of the costs for the trucks on the potential of implementing platooning. To conclude this analysis, we analyze the results for the approaches after implementing it in TBAs software in Section 6.5.

6.1 Data

We start by giving an extended view on the environment that we are considering and the corresponding parameters. The CER area consists of nine different locations between which containers are interchanged, which are APMT2, BSC/Kramer, Customs, Distripark, DR Depots, ECT, Euromax, RPH, and RWG. Next to this, we introduce a parking area, called Area 21, from which the trucks will start in the beginning of the simulation and will return when they are not in use. Figure 9 gives a graphical representation of the locations of the different facilities. The assumed distances between those facilities are presented in Table 8 in Appendix C. Subsequently, the driving times between the facilities can be calculated by means of those distances and the related driving speed.

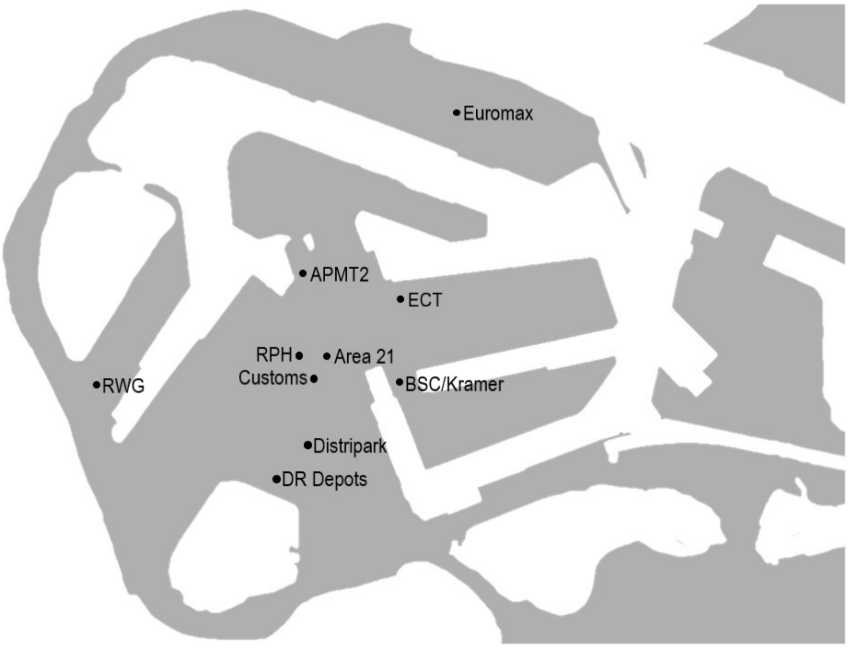


Figure 9: Map of the Container Exchange Route

As stated, we assume constant (un)loading and driving times on the terminals in the Java simulation. The (un)loading time is assumed to be 15 minutes per container (irrespective of whether it is a 20ft or 40ft container), meaning that (un)loading takes 30 minutes whenever a truck transports two 20ft containers. Further, we set the terminal driving times equal to 20 minutes, so 10 minutes to get to the stack and 10 minutes to get off the terminal. Here we assume that we do not count another 20 minutes, whenever a truck is already at a terminal and gets a new assignment at the same terminal.

The time horizon considered in the following simulations is set to be three weeks (i.e. 21 days) where we consider the first week as the warm-up period. This means that the results are based on the last two weeks in order to get results that simulate reality more accurately and therewith mimic the situation in which platooning is in full operation.

Ten different datasets have been generated to be able to generate reliable results by means of ten different replications. The data sets range from 56268 to 56389 containers. As this number will also change and differ from each other after the pre-processing, we take a weighted average of the results obtained by the replications based on the number of containers of the given replication.

6.2 Pre-processing

The data shows that the order lists contain 56,326 containers on average, of which 16,944 20ft containers. The matching model in Section 4.1 is used to make couples that can be transported together on one truck. In general, we assume that a container is nearing its deadline when the current time is within 4 hours before the deadline. In order to make feasible and plausible container couples we assume that the overlapping time window of two containers must be at least four hours. Therefore two nodes in the graph are, next to the other requirements, only connected to each other by an arc when their time windows overlap at least 4 hours.

Two different objectives are tested to form container couples, the first being to maximize the number of couples made and the second to maximize the overlap of the time windows of the container couples. In the first case we simply set $w_e = 1$ for all $e \in E$, where for the second we set w_e equal to the overlap of the time windows of the two nodes $u, v \in V$ that are adjacent to edge $e \in E$. Table 2 presents the results obtained for the two different objectives. It can be concluded that the second objective generates slightly larger overlapping time windows, but at the cost of around sixty less container couples and an increased running time. Based on this large running time, it is decided to use the Matching Problem with $w_e = 1$ for the models in the coming sections.

Table 2: Computational results of the Matching Problem for the objective that maximizes the number of couples and the one that maximizes the overlap of the time windows of the container couples

Time window	**Min. (min)**	**25_{th}pctl (min)**	**Med. (min)**	**75_{th}pctl (min)**	**Max. (min)**	**Avg. (min)**	**Duos (#)**	**Running time(sec)**
Max. couples	244	1,950	2,795	2,880	2,880	2,343	5,190	75
Max. time windows	287	2,829	2,880	2,880	2,880	2,760	5,126	1,226

6.3 Mathematical model

The purpose of this model was to determine the fuel reduction that could be reached under the assumption that all information on the container arrivals is already known in advance. This could then be compared with the fuel reduction reached by the simulation models in order to analyze the performance of the simulations.

However, the Pickup and Delivery Problem is a variation of the Vehicle Routing Problem, which is known to be NP-hard. Consequently, our PDPPTW model is not able to solve to optimality within reasonable time and for reasonable number of containers. It takes up to an hour to solve the problem for only a hundred containers. This is not a workable number as the simulation models are run for thousands of containers. Therefore it is decided not to perform the comparison between the exact model and the simulations.

6.4 Online solution approaches in Java

In this section we discuss the results that were generated by implementing the three approaches in Java. To determine the total fuel consumption we use equation 5 of Section 3.2. This formula is called every time a platoon leaves, meaning that the fuel consumption for that given distance and speed is determined. The results are obtained by determining the lowest possible fuel consumption for a given number of trucks. This means that we search for the best parameter set for each number of trucks. We define the following parameters: the number of trucks $K = \{75, 76, \ldots, 115\}$, the platoon length $L = \{2, \ldots, 8\}$, and the waiting time $W = \{5, 10, \ldots, 240\}$

in minutes (where $PT = W$ for the adjusted approach). This means for the fixed approach that we run for each number of trucks in K each parameter combination of L and W in order to find the combination that gives the lowest fuel consumption for the given number of trucks. Then, for the adjusted approach we test for every number of trucks in K every waiting time value in W and then determine which waiting time value gives the lowest fuel consumption for the given number of trucks. The same procedure is done for the group approach but this time we run for every number of trucks in K every value in L. As stated, we run ten replications of the previously described options and take the average of these results. Here, one replication simulates three weeks in which we view the first week as a warm-up period and therefore only the last two weeks are used for the results. First, we discuss the reliability and predictability of the parameters sets in Section 6.4.1. Subsequently, we dive deeper into the results by analyzing the behavior of the three approaches in Section 6.4.2. Then in Section 6.4.3 we give a sensitivity analysis on the velocity. Finally, we determine the influence of the costs for the trucks on the potential of platooning in Section 6.4.4.

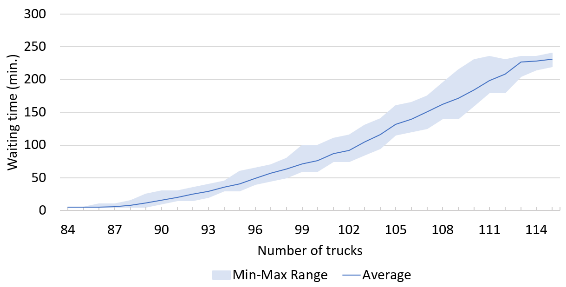
6.4.1 Parameter predictability

Before we look into the results generated by the three approaches, we first view the models on a higher level by reviewing their parameter choices. We ran ten replications, meaning that we ran the simulation models for ten different order lists with containers. Because of this, it can occur that the best parameter set that is found for a given number of trucks differs among the ten replications. Therefore, in this section we want to look into the reliability and predictability of choosing the best parameters. Figure 10 presents the average parameter value that is chosen as best for a certain number of trucks for each approach and for each parameter type that they use. Further, the minimum and maximum range between which values are found over the ten replications are presented. Consequently, we show for the adjusted approach the values for W (or PT) that were chosen for a given number of trucks over the ten replications. The same thing is done for the fixed approach for parameter W and L. Finally, we present for the group approach which values of L were chosen. These results are based on the situation in which we assume a speed of 50 km/h.

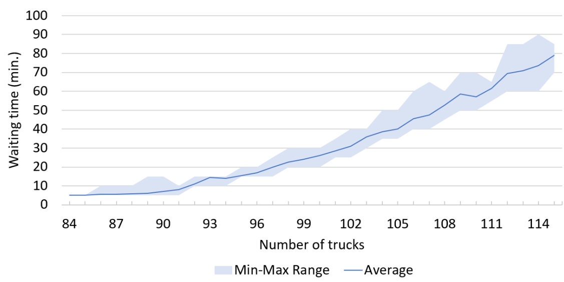
Figure 10a presents the values that were chosen for the waiting time in the adjusted model. The curve increases slowly but steady and the range band shows that there does not seem to be a large variation in the values that were chosen. Consequently, after determining the best parameter value for a certain number of trucks, one can guess the values for other numbers of trucks in a quite straightforward way. For the fixed approach we find a similar line for the average of the waiting time parameters and also the range reaches comparable widths in Figure 10b. Therefore, the waiting time values are also predictable for the fixed approach. After all, it is obvious that these curves have a positive slope as a higher number of trucks gives the model room to increase the waiting time in order to create longer platoons and therewith reduce the fuel consumption (while still delivering 95% of the containers on time).

Let us now look at the reliability of the values for the platoon length. For the fixed approach it can be seen in Figure 10d that the average platoon length increases in a linear way, but from 90 trucks the graph starts to show some unexpected behavior. In addition, the range between the minimum and maximum value found, seems to differ significantly, especially for smaller number of trucks. The figure shows that the platoon length reduces slightly whenever the waiting time increases in order to make up for the longer waiting time. Then, the platoon length increases again when the waiting time remains constant and the number of trucks increases, as the increased number of trucks gives room for longer platoons. This way, the values of the platoon length fluctuate constantly and appear to depend on the waiting time and the number of trucks.

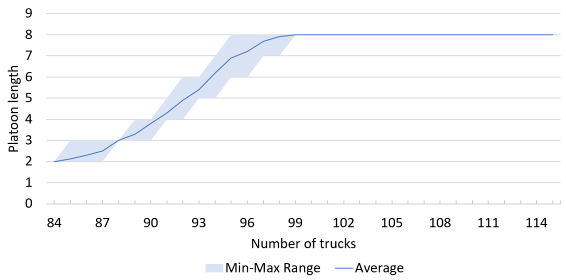
The group approach, on the other hand, presents in Figure 10c a steadier curve with less variability in the chosen values. This can be explained by the fact that this parameter is not influenced by a waiting time parameter. After a hundred trucks, all replications even reach a consensus on a platoon length of eight trucks. It can therefore be concluded, that the parameter values of the group approach can be approximated significantly easier and better than for the the fixed approach. In the end, it is reasonable that the the platoon length increases when the number of trucks increases as more trucks allow for longer platoons to reduce the fuel consumption (while meeting 95% of the deadlines). Logically, the curve finally stagnates at eight trucks as this is the maximum number of trucks that we allow in a platoon.



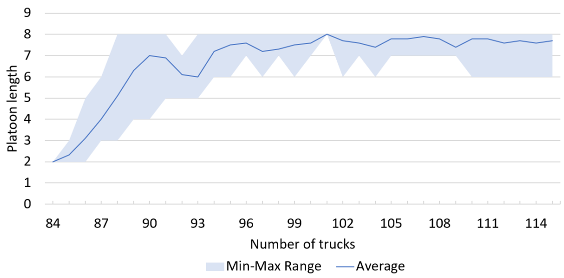
(a) Adjusted approach (PT)



(b) Fixed approach (W)



(c) Group approach (L)



(d) Fixed approach (L)

Figure 10: Parameter settings for different number of trucks based on ten replications

6.4.2 Computational results in Java

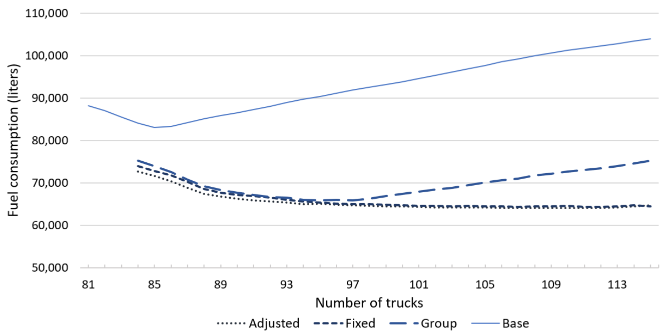
In this section we go into more detail on the results of the different approaches. Again, the presented results are generated under the assumption that a speed of 50km/h is maintained. To obtain the results in Figure 11, we determined for each instance the results that corresponded to the parameter set that caused the lowest fuel consumption for the given number of trucks and took the average of those results. Note that this parameter set is not the same for all ten replications, but we determined the best set per replication as Section 6.4.1 already showed that the best parameter set differs per replication. We aim to observe and explain the behavior of the approaches, and weigh their potential against the outcomes of the base case. Here we define the base case as the original CER model in which trucks drive separately and do not platoon together, as described in Section 5.5.

We start the analysis by evaluating the ad hoc approaches, i.e. the adjusted and fixed approach, by means of Figure 11a to 11h. The figures show that these two approaches generate comparable results, not only in terms of fuel consumption but also in the way they form platoons. Remarkable is that the fuel consumption stagnates from around 97 trucks. Looking at the driven distance, one can see that the number of kilometers driven, both for the total as the empty kilometers, is relatively constant regardless the number of trucks. Of these kilometers, the models can drive about 70 to 80% in a platoon for larger number of trucks. Diving into the formation of these platoons, we see that the more trucks are used, the longer the platoons become. Here, the adjusted approach seems to generate slightly longer platoons than for the fixed approach. The time to form these platoons, however, is almost exactly the same for both approaches and increases in a linear manner with the number of trucks. The consequence of these longer platoons and waiting times is that the occupancy rate of the trucks remains relatively the same. Here we define the occupancy rate as the total percentage of time that a truck is busy with an assignment, which includes the loading, platoon formation, and driving process. Subsequently, the handling times of the containers, which is the time from container arrival till delivery, also do not differ significantly. After all, we can conclude that the ad hoc approaches in terms of fuel consumption and driving distance are robust towards variability in the number of trucks. In terms of platoon formation, we see that the model aims to create longer platoons for higher number of trucks by increasing the maximum platoon lengths and waiting times, in order to keep the fuel consumption low.

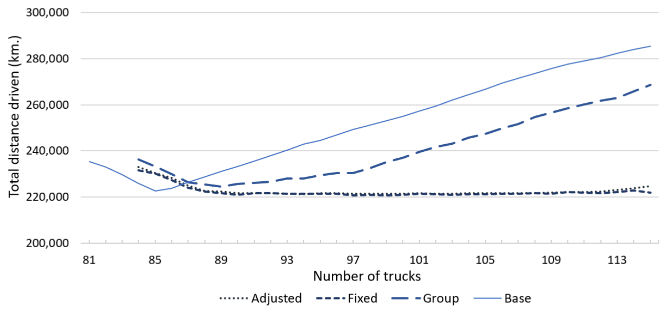
Different results were found for the group approach. For a start, we evaluate the total fuel consumption that is generated with this model. The model seems to give similar results as the ad hoc approaches for lower number of trucks, but the fuel consumption of the group approach starts to increase for the number of trucks where the other two approaches stagnate. Regarding the distance that is driven, it can be noticed that both the total distance as the empty distance curves demonstrate a comparable shape. The percentage of those kilometers that were driven in a platoon are slightly higher and stagnate at around 85%. The platoon length of the platoons, on the other hand, are chosen differently than in the other approaches and therewith result in a higher average platoon length. The model aims to create longer platoons for higher number of trucks in order to reduce the number of platoons and therefore reduce the number of platoon leaders (which do not have the benefits of platooning). In turn, large platoons make it more difficult to find a batch that is large enough to fill the platoon, which increases the number of empty kilometers. As finding such a batch is not always successful, this leads to a significantly lower occupancy rate than the other approaches as it simply has to wait longer till the next assignment. This rate declines slightly as the platoon size remains the same for larger number of trucks, because of which more platoons are created and this reduces the chance that enough containers are available to fill a platoon entirely. Consequently, the handling time decreases steadily from the point where the platoon takes a length of eight. Though, it must be said that the handling times are slightly higher than the other approaches for lower number of trucks.

In conclusion, the group approach might be as favorable as the other approaches for smaller number of trucks, but is outperformed for larger number of trucks. In addition, as the fuel consumption is also slightly higher for the smaller number of trucks and handling times appear to be higher here, the ad hoc approaches would be favored.

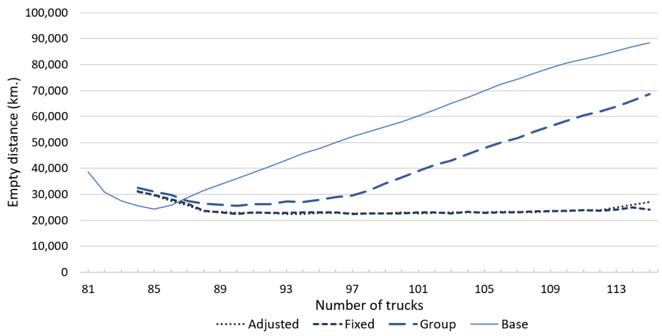
Given these results and insights, we compare the approaches with the base case and determine the potential of each of the approaches. The base case is already feasible for 81 trucks whereas the other approaches need at least 84 trucks to satisfy enough container deadlines. In general, the fuel consumption curve of the base case takes significant higher values than the platooning approaches. The base case reaches a minimum at 85 trucks and then steadily increases due to an increase in the empty kilometers. A higher number of trucks obviously results in a lower occupancy rate and therewith lower handling times. The handling times are also lower than the platooning approaches as it does not include the platoon formation process. Nevertheless, it also appears to cause the container assignment to become less efficient. More trucks will lead to less available assignments whenever a truck finishes its current assignment, which results in more empty kilometers because it has to drive to a different location or has to drive back to the parking area. At the same time one would expect that a newly arrived container has more idle trucks to choose from and therefore is more likely to find a truck closer to its pickup location, which decreases the fuel consumption. Further research showed that for 95 trucks an idle truck has on average 6 locations from which containers can be picked up, whereas for 115 trucks this decreases to around 4 pickup locations on average. On the other hand, whenever a container arrives, there are on average 0.10 locations with an idle truck for the model with 95 trucks. For 115 trucks this increases to 0.30 locations with idle trucks. Even though the previous shows relatively a larger increase, a container still has on average less than one location to choose an idle truck from. The reduction from 6 to 4 container pickup locations, however, results in two less locations because of which the empty miles increase significantly as the majority of the containers are not assigned at arrival but whenever a truck becomes idle.



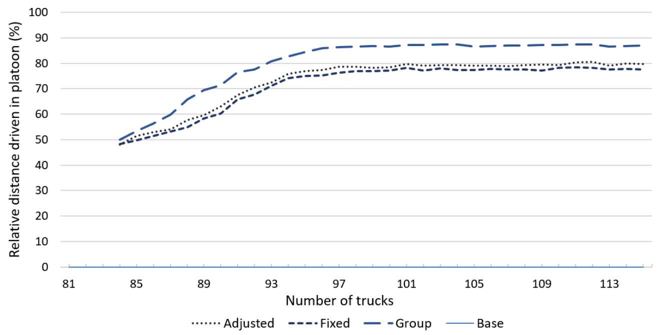
(a) Average total fuel consumption in liters



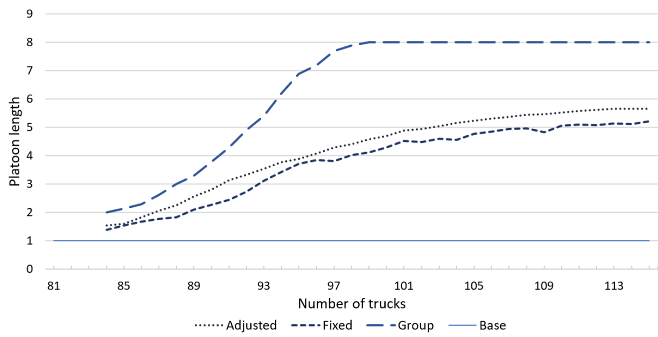
(b) Average total distance driven in kilometers



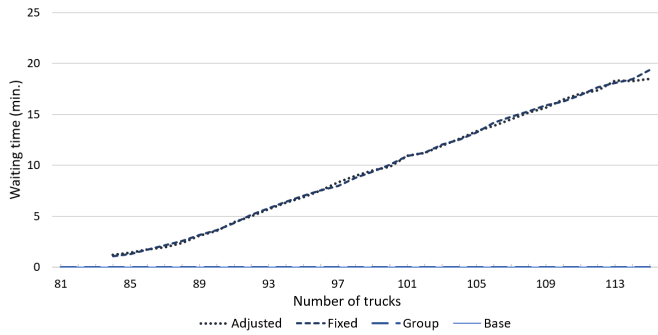
(c) Average total kilometers driven by driving empty to a different pickup location or the parking area



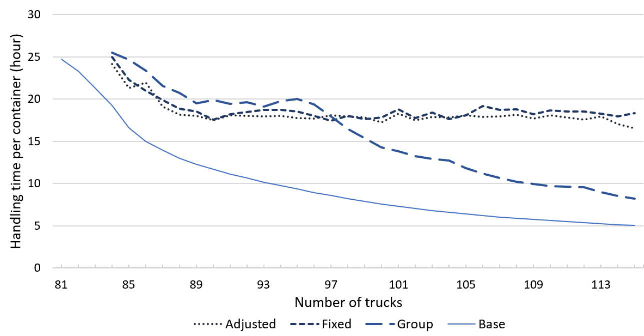
(d) Percentage of kilometers that is driven in a platoon (i.e. driven as a follower)



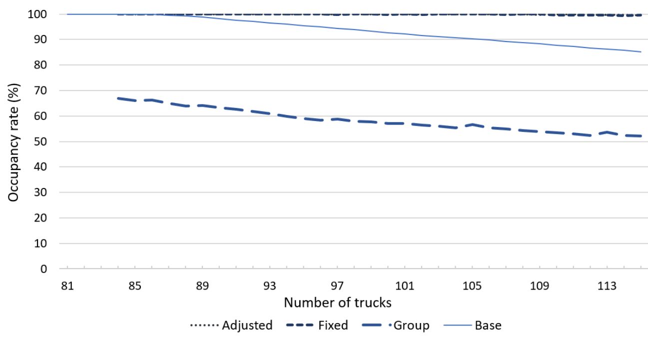
(e) Average number of trucks in a platoon



(f) Average waiting time to form a platoon



(g) Average handling time per container in hours



(h) Average occupancy rate

Figure 11: Computational results for the three online approaches and the base case without platooning, while considering a speed of 50km/h

Given that the fuel consumption stagnates from 97 trucks for the ad hoc approaches and increases for the group approach, we now describe the potential that can be reached up till this number of trucks. It can be concluded that all three can reach similar benefits, but the minimum fuel reduction of the ad hoc approaches is significantly higher than the group approach, as presented in Table 3. On average, the adjusted and fixed approach outperform the group approach again with a slight preference for the adjusted approach. Finally, the analysis of this section has shown that the ad hoc approaches seem favorable in terms of fuel consumption and robustness, but the results cannot prove a significant difference between the adjusted and fixed approach. Therefore, we continue this research in TBAs software only with the ad hoc approaches.

Table 3: Percentage of fuel consumption that can be saved for the three approaches for maximally 97 trucks relative to the base case with 81 trucks

	Adjusted	**Fixed**	**Group**
Minimum	17.5%	16.1%	14.6%
Maximum	26.6%	26.2%	25.2%
Average	23.8%	22.8%	22.0%

6.4.3 Velocity sensitivity analysis

In this section we discuss the influence of the velocity on the benefits of platooning by means of a sensitivity analysis. We test different values for the speed $v \in \{30, 40, 50, 60\}$ in equation 5 of Section 3.2. Similar as in Section 6.4.2 we run ten replications of the three approaches for the given speed and determine the parameter set that generates the lowest fuel consumption for the given number of trucks.

Section 6.4.2 already concluded that a higher number of trucks often generates higher fuel reductions as longer platoons can be created. However, it is not likely to assume that a company will buy many more trucks than necessary just to reduce the fuel consumption with a few percent, as this results in high costs. Therefore, we use the simulation results to determine the number of trucks and corresponding fuel consumption (per two weeks) that is most cost efficient over a term of ten years. This period of ten years is based on the assumption of TBA that trucks have a lifespan of around ten years.

In order to define the total costs we have to make a decision on the cost of a truck. Research shows that there is no clarity yet on the expected costs of an autonomous truck. Huff (2021) estimate that the self-driving technology can cost up to around 100,000 dollars. Assuming that a normal truck costs around 90,000 euros, we assume in this sensitivity analysis an average cost of 200,000 euros for an autonomous truck and a diesel price of 1.304 euros, which is based on the average diesel price between 2014 and 2021 (Shell, 2021). We do not take the fluctuations of the diesel price into account and do not consider the interest that would be obtained if the money would be on the bank instead, as we think this to be out of the scope of this thesis.

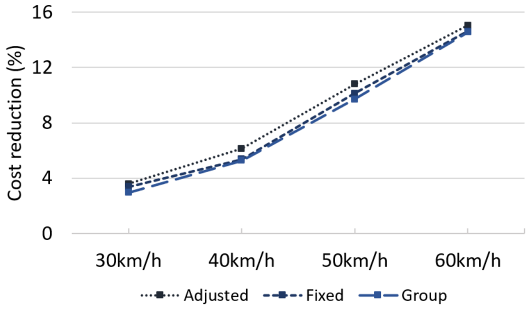
Further, it was stated that the number of orders generated are based on peak weeks and are therefore on average 20% higher than an average week. Consequently, we divide the fuel consumption by factor 1.2 to be able to determine plausible costs. It must be said that this is an approximation as the fuel consumption does not necessarily scale linearly in the number of containers. Nevertheless, we have chosen to approximate it this way as using the fuel consumption for a peak weak is likely to be an overestimation for the largest part of the year. A simulation that is run for a time horizon of an entire year including expected peak and quiet periods would result in a more accurate approximation. However, we decided to simulate only three weeks due to the time limits set for this thesis. After all, the total cost is built up of the costs to buy the trucks at the beginning, and the costs for fuel consumption over a time period of ten years.

Figure 12 presents the results for the four different speeds. Figure 12b visualizes for each speed the reduction in fuel per two weeks relative to the fuel consumption of the given speed for the base case without platooning. The positive slope can be explained by the fact that the part in the fuel formula that is related to the air drag contains the speed to the third power. This is then multiplied by the reduction factor in the platooning models, which then results in larger reductions for higher speeds. In terms of liters this means that 5,000 up to 30,000 liters can be saved every two weeks (from slowest to fastest speed).

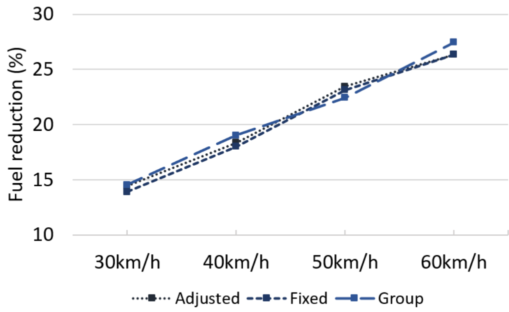
Looking at the number of trucks in Figure 12c, it can be seen that the number decreases for higher speeds. In general you need more trucks for low speeds in order to be able to meet the deadline requirements. Faster speeds could also choose these higher number of trucks, as higher number of trucks often have lower fuel consumptions. However, this is cost inefficient and therefore one chooses lower number of trucks in exchange for a slightly higher fuel consumption. Consequently, the payback period is often higher for lower speeds as those are defined by higher number of trucks and lower fuel reductions. Here, we see the payback period as the number of years until the platooning model costs less than the base model. We assume that the trucks are bought at the beginning of the time horizon and that this difference in costs is made up over the years by the reduction in fuel consumption.

After all, we see in Figure 12a that the cost reduction on a term of 10 years is higher for higher speeds relative to the cost for the base model with the same speed. This can save from 1.5 million for 30 km/h up to almost 9 million euros for the highest speed. Now, one might conclude that it might be better to maintain a higher speed, however, looking at Figure 12e, we see that the cost at the end of the ten years are significantly lower for lower speeds and can save around 15 million euros if the speed is reduced from 60km/h to 30 km/h. That is why, it can be concluded that in general cost and fuel reductions can be obtained whenever platooning is applied while maintaining the same speed. Nevertheless, it is more beneficial to reduce the speed as this results in much lower costs and fuel consumption. In any case, the benefits of platooning are favorable within around five years latest.

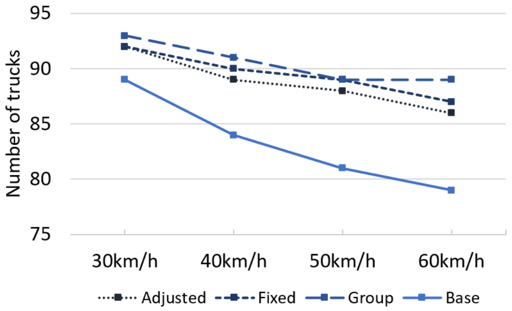
Finally, regarding the three approaches, the adjusted approach appears to be favorable in terms of costs, number of trucks, and payback time. In terms of fuel reduction the group approach seems to generate higher fuel reductions for most speeds, but at the expense of more trucks. Do note that all these results are based on the idea that a constant speed is maintained and is therefore an overestimation in terms of costs as the average speed is likely to be lower. The exact values that were found for the models for the different speeds are presented in the tables in Appendix D.1.



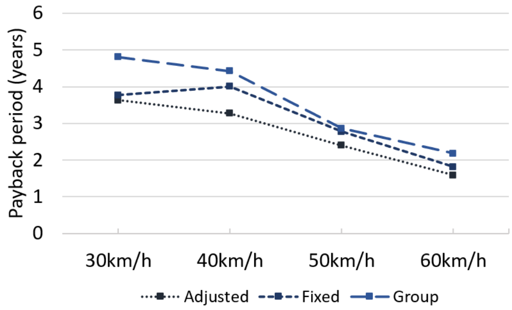
(a) The percentage reduction of the total cost after ten years relative to the base case for every given speed and approach



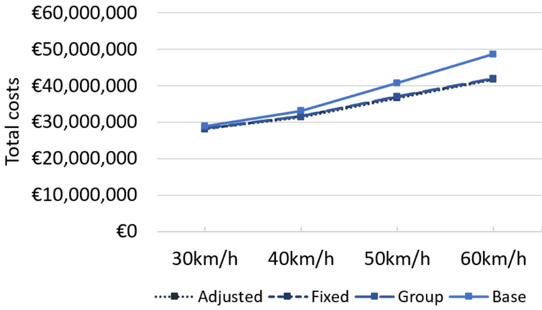
(b) The percentage reduction in fuel consumption every two weeks relative to the base case for every speed and approach



(c) Most cost efficient number of trucks for every speed and approach



(d) Payback time relative to the base case for each speed and approach



(e) The total cost in euros after ten years for each speed and approach

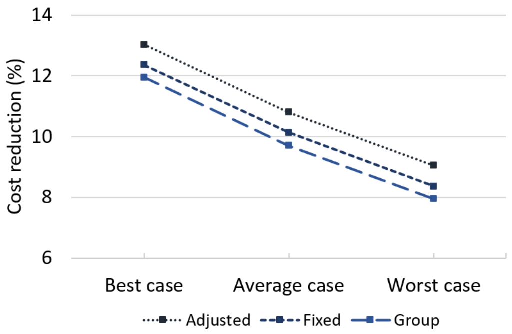
Figure 12: Computational results for the three online approaches and the four speeds, under the assumption of average truck costs

6.4.4 Cost sensitivity analysis

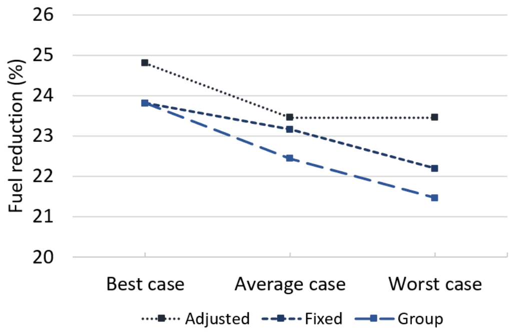
In the previous section it was already stated that the cost of autonomous trucks are still unknown or at least not publicly available yet. Therefore we perform a sensitivity analysis similar to the previous section visualized by Figure 13, but this time for different costs while assuming a speed of 50 kilometers per hour. Three different situations are tested, among which the best case of 150,000 euros, average case of 200,000 euros, and the worst case scenario of 250,000 euros for an autonomous truck.

According to Figure 13b, it can be concluded that less fuel is saved when the costs of the trucks increase. This is caused by the fact that less trucks are preferred to compensate for the rising costs at the expense of a slightly higher fuel consumption. It is, however, not possible to fully make up for the higher truck costs, because of which the reduction in the total costs declines as given in Figure 13a. This results in a maximum cost reduction of five million euros in the best case scenario and minimally about 3.5 million in the worst case scenario. Note that these results are based on the base model with 81 trucks as found in Section 6.4.2. Subsequently, it is still possible to save 19,000 up to 22,000 liters every two weeks. In addition, it takes more time until it becomes more efficient to use platooning when the costs increase, but still within a reasonable period of about three years.

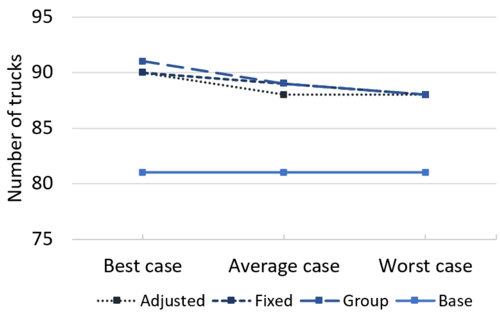
Again the adjusted approach stands out in terms of the cost and fuel reduction, by using low number of trucks, and lower payback periods. Finally, it can be concluded that it is still profitable and appealing in terms of sustainability when the costs for autonomous trucks increase. The exact values that were found are presented in the tables in Appendix D.2.



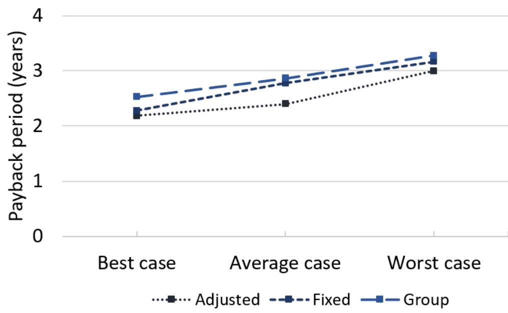
(a) The percentage reduction of the total cost after ten years relative to the base case for every cost and approach



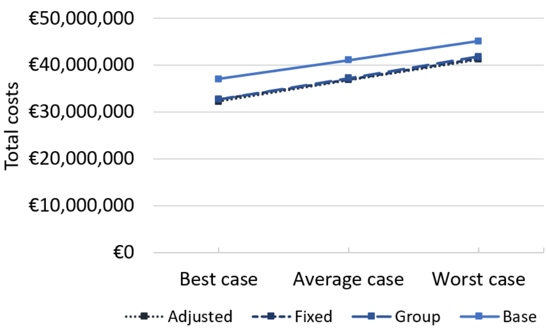
(b) The percentage reduction in fuel consumption for every two weeks relative to the base case for every cost and approach



(c) Most cost efficient number of trucks for every cost and approach



(d) Payback time relative to the base case for each cost and approach



(e) The total cost in euros after ten years for each cost and approach

Figure 13: Computational results for the three online approaches and the base case without platooning for the different costs, under the assumption of a speed of 50km/h

6.5 Online solution approaches in TIMESquare

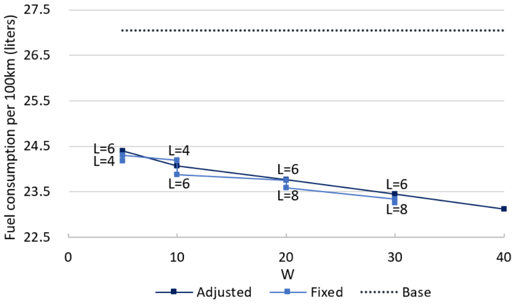
This section reviews the results generated by the TIMESquare simulations. First, we have to highlight that one run, which simulates three weeks (of which one week warm up period), takes about three days to finish. Thus, it takes a significant amount of time to get results, given that we want ten replications of each parameter set for reliable results. That is why it is impossible to run as many parameter sets as in Section 6.4 given the time limits set for this thesis.

From Section 6.4.3 and 6.4.4 it was concluded that it is cost efficient to choose up to around 95 trucks. Given this observation, we use the results of Section 6.4.1 in order to determine the parameters with the highest expected potential. For the adjusted approach we see a range from 5 up to around 40 minutes, whereas for the fixed approach the waiting time ranges between 5 and 20 minutes. For these parameters, we determine the values for the length in the fixed approach accordingly, which ranges from around 4 to 8. Consequently, we have decided to test for the adjusted approach $W \in \{5, 10, 20, 30, 40\}$ (where $W = PT$) and for the fixed approach we test the parameter combinations $(W, L) \in \{(5, 4), (5, 6), (10, 4), (10, 6), (20, 6), (20, 8), (30, 6), (30, 8)\}$.

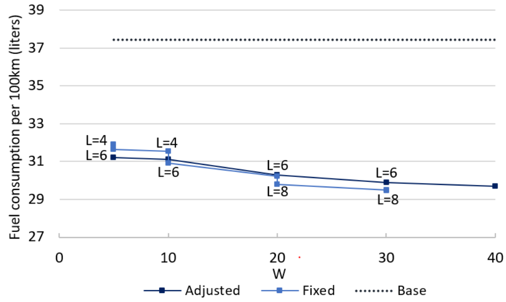
The tests are done for a simulation model which has solely implemented the platoon formation decisions, meaning that the container assignment strategy used by TBA is not changed into our own. This way, we can make a fair comparison between the base model and the platooning model such that efficiency is only generated by the platoons and not by a better assignment. In Section 6.5.2 we make a comparison between the model with and without platoon related assignment to determine the influence of the new container assignment strategy. By means of trial and error we determined for the given parameter set what the lowest number of trucks was that was necessary to ensure that less than 5% of the containers are delivered too late over the ten replications on average. Due to the time limits of this thesis, we increased the number of trucks with five trucks every time until the average of the ten replications met the deadline restriction. We decided to determine the lowest number of trucks as we do not think that it is likely that a higher number of trucks reduces the fuel consumption that much to still cover the extra costs of five more trucks (keeping the Java results in mind). This is therefore a cost related consideration. We present for each set the average results of ten replications for the minimum number of trucks that were necessary to satisfy the deadline constraint for the given parameters. In addition, we ran the base case in which trucks do not drive in platoons. We close this section by comparing the Java model with TIMESquare in Section 6.5.3

6.5.1 Computational results in TIMESquare

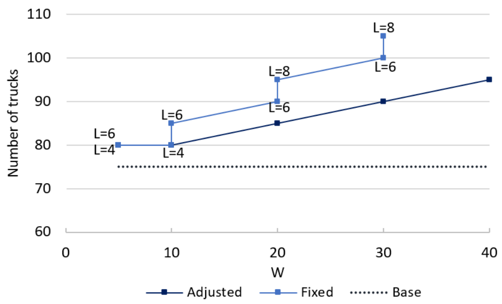
We start by analyzing the results that were generated by the TIMESquare simulation for the given parameters. Figure 14 presents the results for the experiments that were run in the TIMESquare software on the left side and the results that were generated by Java for the same parameters on the right side (which we review in Section 6.5.3). Figure 14a shows that the fuel consumption declines for larger waiting times and platoon lengths as expected. Here, the fixed approach results in a lower fuel consumption than the adjusted approach for the given waiting times. In total, one can save from 2.5 liters up to 4 liters every 100km compared to the base case, which is a reduction of 10 up to 16%. Both approaches seem to get benefits by reaching a lower average speed. This is likely to be caused by the increasing number of trucks which increases the congestion on the road. Extra benefits for the fixed approach are caused by the fact that the fixed approach is able to reach slightly longer platoons by waiting longer. Consequently, trucks drive more kilometers in a platoon than for the adjusted approach as can be seen in Figure 14k and 14m. Note that the trucks do not drive in a platoon on the terminal by definition, therefore we let the terminal distance out of both graphs. Thus, the distance not in platoon is the distance on the roads between the terminals on which trucks were not platoon followers. However, these benefits come at a cost, as Figure 14c shows that the fixed approach requires more and more trucks the higher the waiting time parameter. Both approaches need five extra trucks whenever the waiting time increases by 10 minutes, however, the fixed approach needs another five trucks whenever the platoon length parameter is increased by two trucks.



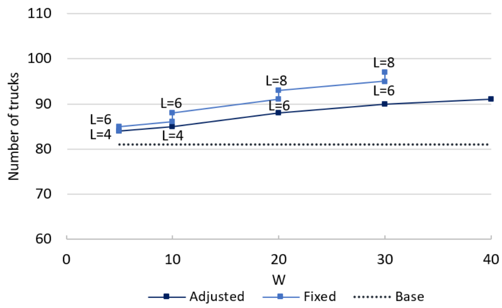
(a) Fuel consumption in liters per 100 km for the approaches in TIMESquare



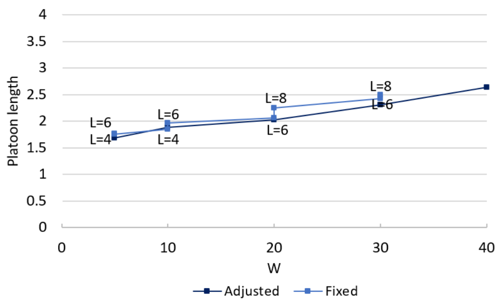
(b) Fuel consumption in liters per 100 km for the approaches in Java



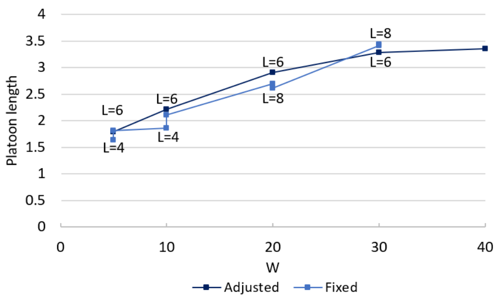
(c) Number of trucks required in TIMESquare



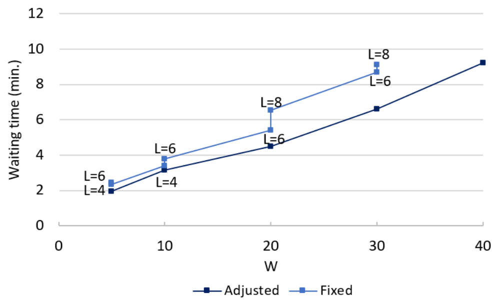
(d) Number of trucks required in Java



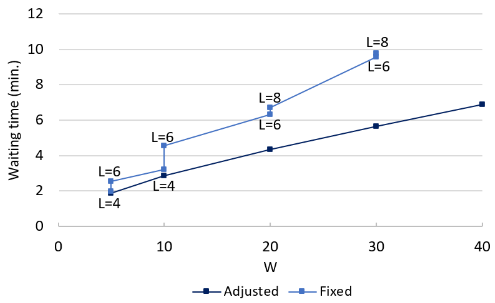
(e) Average platoon length in TIMESquare



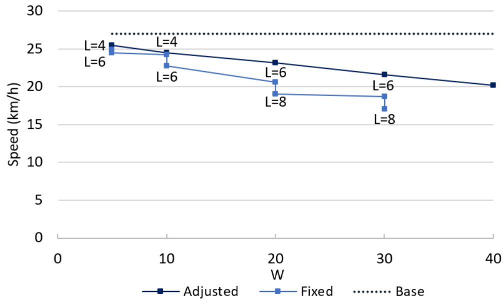
(f) Average platoon length in Java



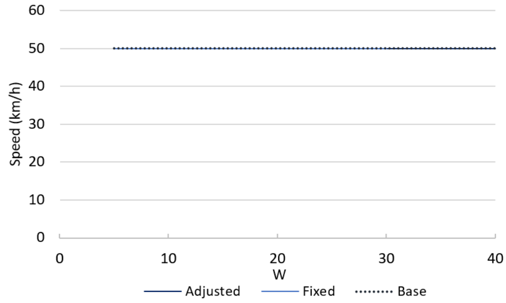
(g) Average waiting time in TIMESquare



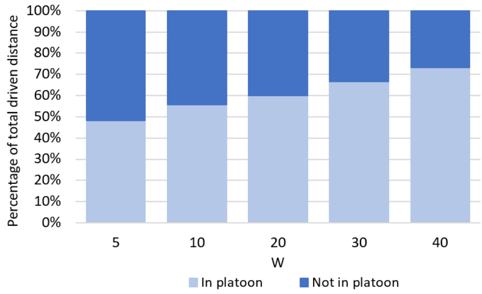
(h) Average waiting time in Java



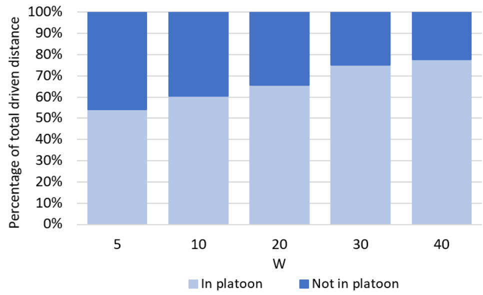
(i) Average speed in TIMESquare



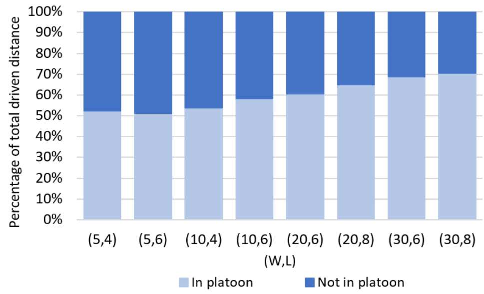
(j) Average speed in Java



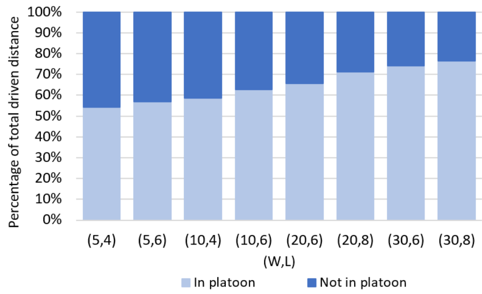
(k) Platooning distribution for the adjusted approach in TIMESquare (without terminal distance)



(l) Platooning distribution for the adjusted approach in Java



(m) Platooning distribution for the fixed approach in TIMESquare (without terminal distance)

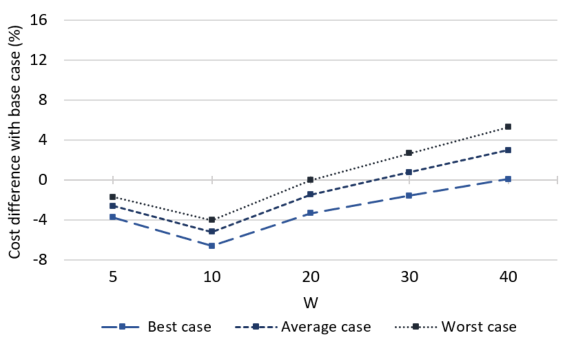


(n) Platooning distribution for the fixed approach in Java

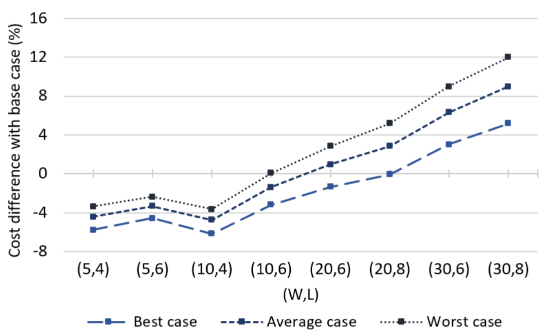
Figure 14: Computational results of the simulation runs in TIMESquare(left) compared with the results generated by the simulations (right) in Java

This higher number of trucks might influence the decision of a company on which approach should be applied. The consequence of this is that the total costs increase significantly, which cannot fully be caught by the fuel reduction generated by platooning. Therefore we perform a cost analysis similar to Section 6.4.4, but this time mainly focus on the influence of the truck costs on the total costs in order to show how attractive it remains for companies in terms of costs. Here, we again assume the best cost of 150,000 euros, the average cost of 200,000 euros, and the worst case of 250,000 euros. Figures 15a and 15b show the percentage difference in costs for the adjusted and fixed approach respectively relative to the base case with 75 trucks. Here, the influence of the number of trucks on the total costs becomes clear as platooning remains profitable for the adjusted approach up to and including 20 minutes waiting time and 85 trucks, whereas the fixed approach is only profitable for a waiting times of 10 minutes and 85 trucks and lower (where the number of trucks follow from Figure 14c). Although the platooning benefits (i.e. cost reduction due to lower fuel consumption) keep increasing for both models for higher waiting times, it appears that both are most cost efficient at 10 minutes. In this case it is possible to save two million euros every ten years and save three liters every 100 kilometers for the adjusted approach. And as Figure **??** already showed that more fuel can be saved for longer waiting times, this can be reached in exchange for higher costs.

It then depends on the company whether the costs or fuel reduction is more important. The figures show that for the adjusted approach under the worst case scenario the company can play break-even for a waiting time of 20 minutes and 85 trucks, which can save 3.28 liters every 100 kilometers. The fixed approach can only save 3.18 liter per 100 kilometers in its break-even point. In the best case the company can save 3.93 liters in its break-even point for the adjusted approach by increasing the waiting time to 40 minutes, whereas the fixed approach can only increase the reduction up to 3.46 liters by increasing the waiting time up to 20 minutes and the platoon length to 8. After all, the costs of the adjusted model seem to increase much slower due to lower number of trucks and therefore creates more flexibility for a company to choose how much they want to pay for a reduction in their fuel consumption.



(a) Percentage cost reduction relative to the base case for the adjusted approach



(b) Percentage cost reduction relative to the base case for the fixed approach

Figure 15: The percentage reduction of the total cost after ten years relative to the base case with 75 trucks for each truck cost

6.5.2 Analysis on the assignment strategy

As already stated, in the previously described models we did not adjust the container assignment strategy yet in which we assign a container to a truck based on the waiting time of the corresponding platoon. We have simulated two test cases in order to show the influence of this assignment on the platoon formation process and therewith the results. We have run the adjusted model with assignment adjustments for the case with a waiting time of 10 and 40 minutes, as presented by Table 4 and 5 respectively.

For a smaller waiting time of 10 minutes, the assignment does not seem to have significant influence. The assignment even seems to have a slightly negative influence as the fuel consumption increases together with the empty distance. Due to the short waiting time, trucks are often not able to reach the platoon before it departs as the handling times at the terminal are between 30 and 50 minutes. Consequently, it might occur that trucks are too often not able to catch the platoon and therefore took a different container than the one that would have been better.

Table 4: Computational results for the adjusted approach where $W = 10$ for the model with and without platoon related assignment

Assign	**Trucks (#)**	**Handl. time(h)**	**Fuel (l/ 100km)**	**Empty dist. (km/box)**	**Platoon length**	**Waiting (min.)**	**Speed (km/h)**	**Occup. rate(%)**
Yes	80	22.54	24.11	2,240	1.86	3.19	24.68	97.60
No	80	24.01	24.07	2,020	1.88	3.15	24.52	97.90

Now considering Table 5, the height of the waiting time indeed does appear to have an influence on the benefits of the platooning assignment. We first regard the bottom two lines of the table, which present the results for the model with and without platoon related assignment. The fuel consumption is relatively similar even though the platoon length is slightly longer for shorter waiting times. This seems to imply that longer platoons within short time can be created due to a more effective assignment. The assignment also seems to be effective in terms of empty distance and that is also why this analysis is done separately. Consequently the handling time decreases and the occupancy rate decreases. Further research even shows that the model can be solved for five less trucks under the same assumptions. This does come at the cost of a slightly higher fuel consumption as the model is not able to reach such long platoon lengths as before. Nevertheless, this fuel consumption is still much lower than the previous lowest fuel consumption of the adjusted approach, which is found for the model with $W = 30$ and 90 trucks, which resulted in 23.46 liters per 100 kilometers as shown in Figure 14a. Reducing the number of trucks for the previous model where $W = 10$ was not possible.

Table 5: Computational results for the adjusted approach where $W = 40$ for the model with and without platoon related assignment

Assign	**Trucks (#)**	**Handl. time(h)**	**Fuel (l/ 100km)**	**Empty dist. (km/box)**	**Platoon length**	**Waiting (min.)**	**Speed (km/h)**	**Occup. rate(%)**
Yes	90	21.47	23.28	1,904	2.5	8.01	20.45	98.20
Yes	95	19.17	23.10	2,039	2.74	9.09	20.11	96.80
No	95	21.27	23.12	2,137	2.64	9.21	20.17	97.60

After all, the assignment might not create much fuel reduction compared to the model without assignment adjustments but it does reduce the delay that is caused by platooning because of which less trucks are necessary. Further, it does come with possible fuel benefits compared to models with lower parameter values. Figure 16 shows the difference in cost reduction for the model with platooning assignment and 90 trucks, and the model without assignment with 95 trucks. We see that the model with $W = 40$ is now still below the break-even point for both the best and average case for only 90 trucks. In terms of costs, it means that 1 million up to 1.5 million euros can be saved. In addition, the model takes lower values than the model where $W = 30$ and also 90 trucks are used, as presented in Figure 15a. About half a million euros can be saved compared to that model due to the lower fuel consumption. After all, this assignment might make it for companies more interesting to invest in platooning as it lowers the costs significantly.

Nevertheless, further research is required to determine the point from which applying the new assignment strategy is profitable, as the previous results show potential benefits.

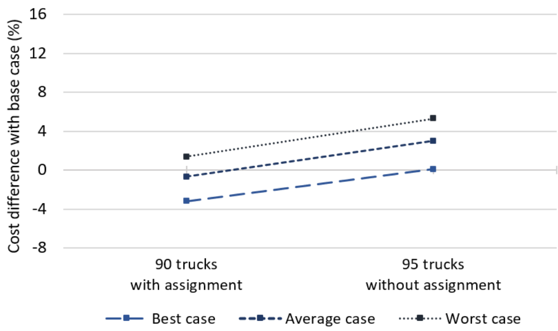


Figure 16: The percentage reduction of the total cost after ten years for the adjusted approach where $W = 40$ relative to the base case with 75 trucks for each truck cost

6.5.3 Comparison with the Java model

We can now compare the TIMESquare results with the Java simulation results presented on the right side of Figure 14. A first glance shows that the curves of both models tend to take similar shapes and therewith similar behavior. The fuel consumption per 100 kilometers for the platooning approaches differ from six up to eight liters between the two models and the base cases differ around ten liters, in which the Java simulation takes the higher values. The percentage reduction of the fuel consumption for the platooning approaches relative to the base case is larger for the Java simulations, which is partly caused by the higher speed that is maintained and by the longer platoons that are formed.

Remarkable is that the adjusted approach generates longer platoons than the fixed approach in the Java simulations, whereas it is the other way around in TIMESquare. A possible explanation for this is the real time approach that is applied in TIMESquare. In Java we use constant driving and terminal handling times measured in minutes, because of which the chances are higher that trucks arrive at the boarding point at the exact same time. We check first which trucks arrive at the given timestamp and if this triggers a departure at the current timestamp then all those trucks are considered to be in the platoon. TIMESquare on the other hand runs real time because of which it might occur more often that a truck that was just a few seconds behind was not allowed to join the platoon anymore. The variation in speeds and handling times lowers the chance that several trucks arrive at the same time even more. Consequently, the platoons are smaller as more trucks will be just too late to catch the platoon. The smaller platoons result in a smaller platooning ratio for the TIMESquare models as displayed in Figure 14k and 14m, than for the Java simulations as given in Figure 14l and 14n.

The only result not discussed yet is the number of trucks that is used. In general the numbers do not seem to differ too much between the models. It is however difficult to make a conclusion on this aspect as we have only checked the number of required trucks in steps of five trucks in TIMESquare, whereas this is checked in Java by increasing the number with one truck at a time. It is therefore possible and likely that the results are much closer to each other.

After all, it must be highlighted that is not completely fair to compare both models with each other. We review the assumptions and present Table 6 where some results based on the average of all models that were tested in Java and TIMESquare are given, in order to substantiate this. One of the assumptions in the Java model was that we do not take the driving on the terminal into account in the calculation of the total fuel consumption. Though, the TIMESquare simulations show that about 24% of the total distance was driven on a terminal area. The loaded distance is, on the other hand, relatively similar to each other. As we have determined the travel distance ourselves, they might differ from the distances in TIMESquare and it might be possible that we already caught a part of the terminal in our calculation.

The empty distance in Java results to be significantly lower, but the cause of this is hard to define. A part of the distance can be lost due to disregarding the terminal distance, but might also be caused by a different implementation of the assignment strategies. It is difficult to compare the two simulation models as they are built up differently in many ways. TIMESquare is a near real time event based simulation model that has been improved for many years and is therefore likely to perform different. The assignment process is much more complex than the one in Java, which already influences the results. In addition, TIMESquare takes delays into account caused by turns, junctions, or congestion, whereas we assume constant driving and terminal handling times. This also influences the speed, as can be seen in Figure 14i and 14j. The average speed is almost half the speed that is maintained in Java due to all these occurrences and speed restrictions on the terminals, because of which the fuel consumption is higher but also the trucks are much faster at their destinations. Moreover, we do not take terminal capacity into account because of which as many trucks can visit the terminal as we want at once. All these assumptions and differences affect the decisions that are made in the simulation models and therewith affect the results of both models. This results already from the fact that both models need different number of trucks to satisfy the deadline constraint. After all, we think that it might not be accurate enough to see the Java model as a replacement for the TIMESquare model, but it can be used to pre-process the parameters for the TIMESquare model to save a significant amount of simulation time. It took Java namely only four minutes, two hours, and twelve hours to find the optimal parameters for the group, adjusted, and fixed approach respectively.

Table 6: Average computational results of the simulation runs in TIMESquare and Java

Model	**Avg. total dist.(km)**	**Avg. loaded dist.(km)**	**Avg. empty dist.(km)**	**Avg. terminal dist.(km)**	**Avg. time at terminal(min)**
Java	225,060.52	190,406.53	34,654.26	0	34.17
TIMESquare	250,518.50	191,647.18	58,871.32	61,035.40	35.25

7 Conclusion

The aim of this research was to determine the potential of platooning in a port area. We approached the problem by means of an online and an offline solution approach, in which the solutions of the online approach could be tested against the offline solution. The offline approach was modeled as a Pickup and Delivery Platooning Problem with Time Windows and defined as a mathematical model accordingly. However, this problem is considered to be NP-hard and could therefore not be solved within reasonable time for a reasonable number of orders.

The online approach took shape as an event based simulation model which exists of a container assignment, truck assignment, and a platoon formation process. We defined three different approaches to facilitate platoon formation at terminals. Two approaches were based on an ad hoc assignment, which means that a container is assigned immediately whenever a truck finishes its current assignment or whenever a container arrives at a facility. Here, the fixed approach forms platoons by defining a maximum platoon length and waiting time, whereas the adjusted approach reviews and adjusts the departure time whenever a truck joins a platoon. The third approach assumes that trucks drive in a fixed platoon formation for the entire planning horizon. Consequently, batches are assigned to a group of trucks instead of per truck. Different parameter sets were tested for the approaches in order to determine the most fuel efficient parameter set for a given number of trucks. The waiting time parameter resulted to be quite reliable and predictable, showing an increasing slope for larger number of trucks. The platoon length can be seen as reliable for the group approach, but shows more fluctuation for the fixed approach as it is correlated with the waiting time parameter.

Regarding the behavior of the approaches themselves, the ad hoc approaches provide similar results and are relatively robust towards variability in the number of trucks. As expected, the model chooses parameter sets in such a way that longer platoons can be formed when the number of trucks increases. The group approach on the other hand starts of with comparable results but generates worse solutions for larger number of trucks, whereas the fuel consumption of the other two approaches stagnates. Comparing the fuel consumption with the base case without platooning, it is concluded that the adjusted approach reaches the highest fuel reduction from 17% up to almost 24%. The fixed approach can reach almost the same reduction but scores on average 1% lower, after which the group approach performs even another percent worse.

To extend this analysis, a sensitivity analysis is performed on the speed and costs of the trucks. The results show that a higher percentage reduction in fuel and costs can be reached for higher speeds. Nevertheless, it remains more efficient in terms of total costs and fuel consumption to choose lower speeds. Next to this, higher truck costs do potentially decrease the cost and fuel reduction, but platooning remains favorable over time anyways. In both analyses the adjusted approach appears to be the best approach.

On the grounds of these results, it is decided to only further evaluate the two ad hoc approaches. These two approaches were implemented in the more extended simulation software of TBA, called TIMESquare, which simulates the container exchange in a near real time manner. There appears to be a trade-off between the adjusted and the fixed approach in terms of fuel reduction and number of trucks. The fixed approach results in slightly lower fuel consumption than the adjusted approach, but at the cost of more trucks. Consequently, it is for companies more interesting to apply the adjusted approach as it is cost wise more flexible and therewith gives the company a greater opportunity to choose how much it wants to pay in exchange for fuel reduction. These costs could even be further lowered by adjusting the container assignment strategy into one that stimulates platooning, as it reduces the required number of trucks in return for a slightly higher fuel consumption for larger values of W. Finally, it is concluded that it is not completely fair to compare the TIMESquare with the Java model due to the assumptions that were made in the Java model and the difference in the build-up of both the simulation models. Nevertheless, it does appear useful to use it to pre-process the parameters in order to save time.

After all, it can be concluded that, even though the fixed approach might use slightly less fuel, the highest potential for platooning can be reached by means of the adjusted approach. About 2.5 to 4 liters can be saved every 100 kilometers, which is a reduction of 10 to 16%, and the cost can be decreased up to two million euros every ten years. It is finally up to the company that wants to use platooning to make the trade of between fuel consumption and costs.

8 Discussion

To conclude this research, we discuss the strength and weaknesses of this research, and provide some recommendations for future research subsequently.

To summarize, we aimed to tackle the platooning problem by means of three different types of models. For a start, an offline solution approach was constructed by means of a Mixed Integer Program for the so called Pickup and Delivery Platooning Problem with Time Windows. As the Vehicle Routing Problem (VRP) is already known to be NP-hard, this variation of this problem is NP-hard as well. Consequently, this problem could not be solved within reasonable time and for reasonable number of containers. It could be interesting to investigate potential relaxations and approaches such as decomposition algorithms in order to solve the problem in a near optimal way. In this research the choice was explicitly made not to do this, as we were mainly interested in what the potential of platooning would be in case we had all information. Consequently, approximating this by other approaches would mean that it loses its purpose for comparing it with the simulations. At the same time, having all the information is a limitation as well as the arrival time of the containers is in real life often unknown or unreliable.

That is why, our event based simulation model, the second model, finds its strength in the fact that it simulates a situation which is closer to reality as the arrival times of containers are unknown. We tested three different approaches among which two were based on an ad hoc approach and one was a group approach. In general, it could be recommended to look deeper into an efficient build up of the simulation. It might for example happen that several trucks are finished at the same time but we generate separate events for this, even though it might be more beneficial to create one single event. This way, you can check from a container point of view which truck is closest to pick it up, instead of checking from a truck point of view which container is closest.

Finally, the simulation approaches were also implemented in TBAs software TIMESquare which mimics reality near real time. The strength of this simulation is that trucks accelerate and decelerate at arrival or departure or due to congestion, because of which driving times are more plausible than the basic simulation. Next to this, the handling times at the terminals are simulated in a more realistic way and the fuel consumption for driving at the terminals is taken into account. In addition, there is a maximum on the number of trucks that can visit a certain terminal in order to take congestion into account, whereas the previous two models do not assume a maximum because of which more trucks arrive than there is capacity.

There are also still some opportunities for improving the approaches that were implemented in the previous two models. For the ad hoc approach we determined the container assignment based on the waiting times of a platoon. Different assignment strategies could be defined and tested, for example predictions can be made on whether the truck would be able to reach the platoon in time or an assignment strategy can be defined in which the container choice is based on the longest platoon in order to speed up the platoon formation process. Another interesting area is the opportunity to change assignments. Currently, we do not do anything to stimulate platooning when a truck is assigned to a different pickup location. It could be valuable to allow for a change in assignments whenever the truck arrives at the other terminal and it appears that it is more beneficial to pickup a different container to stimulate platoon formation. Finally, it is advised to perform a more in-depth parameter tuning in which parameter sets are tested that have not been reviewed in this research. Also the speed that is used in Java could be tuned, so that the results get more reliable compared to the TIMESquare results.

In addition, all the models were built on several assumptions during this research. The first two assumed constant travel and handling times, infinite capacity and zero costs for driving at the terminal, no acceleration and deceleration measures within the fuel formula, infinite amount of fuel, and a constant air drag reduction factor. The models can be made more realistic by adapting these assumptions and therefore it is advised to perform a more extensive research on which assumptions can be dropped easily without reducing the efficiency of the models too much. The last model only operates under the last three assumptions. Because of the infinite fuel there is no time calculated for refueling, however in reality this might influence the number of trucks necessary to meet the deadlines. Therefore it is recommended to consider refueling in order to determine a better approximation on the number of trucks. Another assumption is the constant speed, which results in constant driving times and a constant fuel consumption between two facilities. Our exact model and Java simulation therefore generate an overestimation of the fuel consumption. The extended simulation model is more accurate, but still does not take the consequences of braking and accelerating into account. A more extensive fuel formula that does consider this could improve the accuracy of the results. Finally, we assume a constant air drag reduction factor. However, this also depends on the speed and the weight of the trucks. In case an adaptable reduction factor is used in which those things are considered as well, it might be interesting to place the trucks of a platoon in order of weight so that the reduction factor is maximized.

To conclude, we have built simulation models which can answer the main question of this research. Nevertheless, there is still much opportunity for improvement of these models, because of which the potential of platooning can be determined even more accurate.

References

Adler, A., Miculescu, D., & Karaman, S. (2020). Optimal policies for platooning and ride sharing in autonomy-enabled transportation. In *Algorithmic Foundations of Robotics XII* (pp. 848–863). Springer.

Alam, A. (2011). *Fuel-efficient distributed control for heavy duty vehicle platooning* (Unpublished doctoral dissertation). KTH Royal Institute of Technology.

Bhoopalam, A. K., Agatz, N., & Zuidwijk, R. (2018). Planning of truck platoons: A literature review and directions for future research. *Transportation Research Part B: Methodological*, *107*, 212–228.

Bonnet, C., & Fritz, H. (2000). *Fuel consumption reduction in a platoon: Experimental results with two electronically coupled trucks at close spacing* (Tech. Rep.). SAE Technical Paper.

Boysen, N., Briskorn, D., & Schwerdfeger, S. (2018). The identical-path truck platooning problem. *Transportation Research Part B: Methodological*, *109*, 26–39.

Chaimowicz, L., Campos, M. F., & Kumar, V. (2002). Dynamic role assignment for cooperative robots. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)* (Vol. 1, pp. 293–298).

Cigolini, R., Perona, M., Portioli, A., & Zambelli, T. (2002). A new dynamic look-ahead scheduling procedure for batching machines. *Journal of scheduling*, *5*(2), 185–204.

Desrochers, M., Lenstra, J. K., Savelsbergh, M. W., & Soumis, F. (1987). *Vehicle routing with time windows: optimization and approximation* (Tech. Rep.). CWI. Department of Operations Research and System Theory [BS].

European Commission. (2006). *Emissions and fuel consumption from heavy duty vehicles.* COST 346. Retrieved from https://trimis.ec.europa.eu/project/emissions-and-fuel-consumption-heavy-duty-vehicles%C2%A0#tab-docs

European Commission. (2019). *Reducing CO2 emissions from heavy-duty vehicles.* Retrieved from https://ec.europa.eu/clima/policies/transport/vehicles/heavy_en

Huff, A. (2021). *Analysis: self-driving trucks will force consolidation of trucking market.* CCJ. Retrieved from https://www.ccjdigital.com/technology/article/15066309/why-the-first-movers-in-autonomous-trucking-have-the-edge0

Hussein, A. A., & Rakha, H. A. (2020). Vehicle platooning impact on drag coefficients and energy/fuel saving implications. *arXiv preprint arXiv:2001.00560*.

Larson, J., Kammer, C., Liang, K.-Y., & Johansson, K. H. (2013). Coordinated route optimization for heavy-duty vehicle platoons. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)* (pp. 1196–1202).

Larson, J., Munson, T., & Sokolov, V. (2016). Coordinated platoon routing in a metropolitan network. In *2016 Proceedings of the Seventh SIAM Workshop on Combinatorial Scientific Computing* (pp. 73–82).

Larsson, E., Sennton, G., & Larson, J. (2015). The vehicle platooning problem: Computational complexity and heuristics. *Transportation Research Part C: Emerging Technologies*, *60*, 258–277.

Liang, K.-Y. (2014). *Coordination and routing for fuel-efficient heavy-duty vehicle platoon formation* (Unpublished doctoral dissertation). KTH Royal Institute of Technology.

Liang, K.-Y., Mårtensson, J., & Johansson, K. H. (2013). When is it fuel efficient for a heavy duty vehicle to catch up with a platoon? *IFAC Proceedings Volumes*, *46*(21), 738–743.

Liang, K.-Y., Mårtensson, J., & Johansson, K. H. (2015). Heavy-duty vehicle platoon formation for fuel efficiency. *IEEE Transactions on Intelligent Transportation Systems*, *17*(4), 1051–1061.

Port of Rotterdam. (2021a). *Carbon emissions in Port of Rotterdam drop swifter than national average.* Retrieved from `https://www.portofrotterdam.com/en/news-and-press-releases/carbon-emissions-in-port-of-rotterdam-drop-swifter-than-national-average`

Port of Rotterdam. (2021b). *Carbon neutral in 3 steps.* Retrieved from `https://www.portofrotterdam.com/en/doing-business/port-of-the-future/energy-transition/carbon-neutral`

Shell. (2021). *Historisch prijzenoverzicht.* Retrieved from `https://www.shell.nl/consumenten/shell-fuels/historisch-prijzenoverzicht.html`

Van de Hoef, S. (2016). *Fuel-efficient centralized coordination of truck platooning* (Unpublished doctoral dissertation). KTH Royal Institute of Technology.

Verbruggen, F. (2021). Design of electrified powertrains for heavy-duty trucks. *Eindhoven University of Technology*.

Yang, J., Jaillet, P., & Mahmassani, H. S. (1999). On-line algorithms for truck fleet assignment and scheduling under real-time information. *Transportation Research Record*, *1667*(1), 107–113.

Zhang, G., Cai, X., & Wong, C. (2001). On-line algorithms for minimizing makespan on batch processing machines. *Naval Research Logistics (NRL)*, *48*(3), 241–258.

A Linearization

In Section 4, several constraints are presented in non-linear form. In order to be able to implement those in a solver, we now present the linearizations of these functions.

We introduce the linearization of constraints (16)-(18) from Section 4.3. Then constraints (16) are equivalent to linear inequalities (41) and (42). Constraints (17) are given by inequalities (43),(44), and (45). Finally, we define (47) and (48) in order to linearize constraints (18).

$$A_{jk} \geq D_{ik} + t_{ij} - M \cdot (1 - x_{ijk}) \qquad \forall (i,j) \in A; k \in K \tag{41}$$

$$A_{jk} \leq D_{ik} + t_{ij} + M \cdot (1 - x_{ijk}) \qquad \forall (i,j) \in A; k \in K \tag{42}$$

$$A_{jk} + z_{jk} \geq s_j \cdot \sum_{(i,j) \in A_{pickup}} x_{ijk} \qquad \forall j \in N^+; k \in K \tag{43}$$

$$M \cdot (1 - y_{jk}) - z_{jk} \geq -M \cdot \Big(1 - \sum_{(i,j) \in A_{pickup}} x_{ijk}\Big) \qquad \forall j \in N^+ k \in K \tag{44}$$

$$M \cdot y_{jk} - A_{jk} - z_{jk} + s_j \geq -M \cdot \Big(1 - \sum_{(i,j) \in A_{pickup}} x_{ijk}\Big) \qquad \forall j \in N^+; k \in K \tag{45}$$

$$y_{jk} \leq \sum_{(i,j) \in A_{pickup}} x_{ijk} \qquad \forall j \in N^+; k \in K \tag{46}$$

$$w_{jk} \geq D_{jk} - A_{jk} - l_j - z_{jk} - M \cdot (1 - x_{ijk}) \qquad \forall (i,j) \in A; k \in K \tag{47}$$

$$w_{jk} \leq D_{jk} - A_{jk} - l_j - z_{jk} + M \cdot (1 - x_{ijk}) \qquad \forall (i,j) \in A; k \in K \tag{48}$$

B Pseudocodes

Algorithm 1 Container assignment

```
if There exists a container with high priority then
    Assign the container that is closest to its deadline;
else if There are containers available for pickup at current location then
    Determine which platoon has less assigned trucks than the maximum length;
    Determine for which destination the corresponding platoon has the longest waiting time;
    Assign the container with that destination that is closest to its deadline;
else if There are containers available for pickup at other locations then
    Determine the pickup location that is closest to current location;
    Assign the container at that location that is closest to its deadline;
else
    Assign nothing;
end if
```

Algorithm 2 Fixed platoon formation

```
Input: origin of platoon, destination of platoon, and container c
platoonLength += 1;
if platoonLength = 1 then
    Schedule Depart event at time Tcur+maxWaitingTime for platoon at origin to destination;
else if platoonLength = maxPlatoonLength then
    platoonLength = 0;
    for Every truck in the platoon from origin to destination do
        if truck is at the pickup location of c then
            Schedule Ready for truck at destination at time Tcur+drivingTime+loadTime;
        else
            Schedule Join Platoon for truck at destination for the platoon into the direction of the delivery location of its assignment at time Tcur+drivingTime+loadTime;
        end if
    end for
else if Truck must leave earlier than scheduled Depart in order to arrive on time then
    Schedule Depart at the maximum time that the truck must leave;
end if
```

Algorithm 3 Adjusted platoon formation

```
Input: origin of platoon, destination of platoon, and container c
if platoonLeader = True then
    Tprev=Tcur, constant=0, factor=0;
end if
constant += Tcur - Tprev, factor += 1;
Tdep = (PT+constant)/factor + Tprev;
if platoonLength=maxPlatoonLength then
    Reschedule Depart immediately;
else if The truck must leave earlier than Tdep to be on time then
    Reschedule Depart at latest time the truck may depart;
else
    Reschedule Depart at Tdep;
end if
```

C Parameters

Table 7: Fuel model parameters

Symbol	Description	Value	Unit
A_a	Vehicle frontal area	10.26	m^2
c_D	Air drag coefficient	0.6	-
c_e	Engine combustion efficiency	40	g/l
c_r	Rolling resistance coefficient	$7{\cdot}10^{-3}$	-
η_f	Final drive efficiency	0.97	-
η_g	Gear efficiency	0.95	-
g	Gravitational constant	9.81	m/s^2
i_f	Final drive ratio	3.08	-
i_g	Highest gear ratio	1	-
J_e	Engine inertia	3.5	kg/m^2
J_w	Wheel ratio	65.8	kg/m^2
k_E	Energy conversion factor	-	-
m	Vehicle mass (tractor+trailer+load)	5,400+7,500+L_s	kg
ϕ	Air drag reduction factor	0.68	-
ρ_a	Air density	1.29	kg/m^3
ρ_d	Energy density of diesel	45	MJ/kg
r_w	Wheel radius	0.495	m
T	Time driven a certain speed	-	min
θ	Slope of the road	0	-

Table 8: Distances in kilometers between the facilities

	APMT2	**Area 21**	**BSC**	**Customs**	**Distripark**	**ECT**	**Euromax**	**DR Depots**	**RPH**	**RWG**
APMT2	0	3.9	5.0	3.7	4.1	2.6	15.7	4.7	3.1	4.6
Area21	3.9	0	3.9	0.4	2.5	2.5	15.8	2.2	0.8	6
BSC	5.0	3.9	0	5.0	5.8	2.6	19.2	5.6	4.4	9.6
Customs	3.7	0.4	5.0	0	2.1	2.2	15.5	2.0	0.6	6.2
Distripark	4.1	2.5	5.8	2.1	0	2.6	14.6	1.0	1.0	5.0
ECT	2.6	2.5	2.6	2.2	2.6	0	16.8	3.2	1.7	6.2
Euromax	15.7	15.8	19.2	15.5	14.6	16.8	0	14.5	15.1	9.9
DR Depts	4.7	2.2	5.6	2.0	1.0	3.2	14.5	0	1.5	5.0
RPH	3.1	0.8	4.4	0.6	1.0	1.7	15.1	1.5	0	5.5
RWG	4.6	6.0	9.6	6.2	5.0	6.2	9.9	5.0	5.5	0

D Results

D.1 Velocity analysis

Table 9: Computational results of the adjusted approach for the velocity analysis under the assumption that a truck costs €200,000

	Total costs	**Fuel (liters)**	**Number of trucks**	**Payback period (weeks)**
30 km/h	€28,178,910.94	34,611.53	92	189.05
40 km/h	€31,364,595.99	48,010.60	89	170.19
50 km/h	€36,658,933.03	67,457.29	88	124.67
60 km/h	€41,759,079.87	86,924.54	86	82.79

Table 10: Computational results of the fixed approach for the velocity analysis under the assumption that a truck costs €200,000

	Total costs	**Fuel (liters)**	**Number of trucks**	**Payback period (weeks)**
30 km/h	€28,236,457.33	34,815.21	92	195.88
40 km/h	€31,623,553.91	48,219.28	90	208.25
50 km/h	€36,932,648.90	67,718.20	89	144.30
60 km/h	€41,964,546.94	86,943.89	87	94.67

Table 11: Computational results of the group approach for the velocity analysis under the assumption that a truck costs €200,000

	Total costs	**Fuel (liters)**	**Number of trucks**	**Payback period (weeks)**
30 km/h	€28,364,880.34	34,561.87	93	249.94
40 km/h	€31,654,255.89	47,620.07	91	229.96
50 km/h	€37,110,537.78	68,347.82	89	148.89
60 km/h	€41,994,714.97	85,634.90	89	113.56

Table 12: Computational results of the base case for the velocity analysis under the assumption that a truck costs €200,000

	Total costs	**Fuel (liters)**	**Number of trucks**	**Payback period (weeks)**
30 km/h	€29,229,284.20	40,452.87	89	N/A
40 km/h	€33,419,989.71	58,824.88	84	N/A
50 km/h	€41,098,408.22	88,125.56	81	N/A
60 km/h	€49,152,699.56	118,048.72	79	N/A

D.2 Cost analysis

Table 13: Computational results of the adjusted approach for the cost analysis under the assumption that a speed of 50 km/h is maintained

	Total costs	**Fuel (liters)**	**Number of trucks**	**Payback period (weeks)**
Best case	€32,222,808.77	66,267.61	90	113.67
Average case	€36,658,933.03	67,457.29	88	124.67
Worst case	€41,058,933.03	67,547.29	88	155.83

Table 14: Computational results of the fixed approach for the cost analysis under the assumption that a speed of 50 km/h is maintained

	Total costs	**Fuel (liters)**	**Number of trucks**	**Payback period (weeks)**
Best case	€32,468,890.91	67,138.59	90	118.39
Average case	€36,932,648.9	67,718.20	89	144.30
Worst case	€41,372,121.13	68,565.79	88	164.67

Table 15: Computational results of the group approach for the cost analysis under the assumption that a speed of 50 km/h is maintained

	Total costs	**Fuel (liters)**	**Number of trucks**	**Payback period (weeks)**
Best case	€32,619,351.26	67,140.22	91	131.56
Average case	€37,110,537.78	68,347.82	89	148.89
Worst case	€41,554,460.88	69,211.16	88	170.29

Table 16: Computational results of the base case for the cost analysis under the assumption that a speed of 50 km/h is maintained

	Total costs	**Fuel (liters)**	**Number of trucks**	**Payback period (weeks)**
Best case	€37,048,408.22	88,125.56	81	N/A
Average case	€41,098,408.22	88,125.56	81	N/A
Worst case	€45,148,408.22	88,125.56	81	N/A