Erasmus University Rotterdam

Erasmus School of Economics

MSc Econometrics and Management Science

Operations Research & Quantitative Logistics

Master's Thesis

Traffic Control for Crossing Automated Guided Vehicles on Container Terminals

Author:
Yu Fee Chan
(431320)

Supervisor:
Dr. P.C. Bouman

Second Assessor:
R.S.H. Willemsen

April 30, 2022

Abstract

The integration of automation on containers terminals has lead to several benefits in terms of efficiency and safety. Specifically, on many container terminals the horizontal transportation has been fully automated with the use of Automated Guided Vehicles (AGVs). This thesis investigates the possibility to reduce the delays that AGVs experience from crossing each other, which happens frequently on a congested terminal. A proof of concept is laid out in which a centralized strategy is used that acts like a traffic guard to control the way that AGV cross each other, using information of the AGVs. By incorporating this strategy in a simulation model, the performance is compared to the current situation. It is found that a strategy as such is capable of shortening the completion time of the transportation tasks done by AGVs and thereby increasing the productivity of the quay cranes.

Contents

1 Introduction

With the ongoing increase in demand in the supply chain, it is essential to make the operations that involve container transportation as efficient as possible. In the full process of transporting containers from origin to destination, the handling on container terminals plays a vital role. These are facilities where containers are brought to for transshipment between different transport vehicles or vessels for onward transportation.

Many different operations are involved on container terminals and are carried out by various types of equipment, depending on the requirements and the layout of the terminal. At the waterside, the movement of containers between the deep sea vessels and the land is done by quay cranes. On the land, vehicles are used to move the containers between the quay cranes and the container yard where the containers are stored temporarily for transshipment (horizontal transportation). Some vehicles have the ability to store and stack the containers themselves, while other ones leave the task of yard handling to stacking cranes.

The developments in technology have made it possible to make the operations on container terminals more (cost-)efficient by means of automation at the stacking yard and in the horizontal transportation. The operations with automation are also less prone to human errors, which makes the system more predictable, stable, and safe. While it has brought many benefits, there are still ways to further improve the planning, scheduling, and control management of the automated equipment.

The introduction of automation in horizontal transportation has led to the use of Automated Guided Vehicles (AGV). These are unmanned vehicles with usually an electric or diesel-electric drive unit that travel by following a guided path. Regular ones are not able to load and unload by themselves and therefore require a quay crane or stacking crane to do so. Lift-AGVs, however, are able to place and take containers on racks located at the yard, such that they are decoupled from the stacking cranes.

The fleet of (lift-)AGVs is often controlled centrally, meaning that a central operator handles the dispatching, routing, and traffic control of the AGVs. While the vehicles are driving, it is possible that they come into conflict with each other, meaning that they require to pass or cross the same part of the road at the same time. To resolve such conflicts, they typically work with a first-come-first-served (FCFS) claiming system that is handled centrally. A claim for a part of the path is given to the vehicle that requests it first, provided that the trajectory of the claim is free of other claimed areas. When a claim is rejected, the corresponding vehicle must wait until its desired claim is no longer blocked by other claims.

A FCFS approach does not take certain factors into account, such as the urgency of the transportation requests, or other vehicles that may be affected and delayed by the actions of this rule. Some transportation requests may, for example, be more urgent in terms of their due time, such that giving it precedence for claiming is more beneficial. This is especially the case when the performance of quay cranes is reduced due to their waiting time for interchanges with AGVs. Furthermore, it is possible that a certain decision causes more conflicts in the system at a later stage, which could have been prevented by a different claiming order.

In this thesis, the real-time traffic management problem of AGVs on a container terminal is considered to investigate the relation between the completion time of AGV transport tasks and the quay crane productivity. Instead of using a FCFS approach, a conflict resolution model is used that controls conflicting AGVs by predetermining the order in which they should cross each other. The goal of implementing this method is to minimize the delay of AGVs at the destination of their tasks. Furthermore, it aims to improve on the FCFS approach in terms of the performance of quay cranes. In order for it to be useful in practice, the model needs to be capable of determining a solution within a few seconds, as decisions have to be made in real-time. The proposed method is evaluated using a detailed discrete event simulation model provided by TBA Group, Timesquare.

The remainder of this thesis is organized as follows. First, an overview is given in Section 2 of related work in traffic control in AGV systems. Then, more details are given in Section 3 regarding the operations at a typical container terminal and the problem that is considered in this research. In Section 4, a method is proposed for the AGV traffic coordination in the considered type of container terminal. Section 5 provides more details of the simulation setup and the experiments that are done, followed by the results thereof. Lastly, concluding remarks are given in Section 6.

2 Literature Review

The full process on container terminals consists of several steps. Ideally, all of the steps are considered jointly to find an optimal strategy. However, this would increase the complexity of the problem significantly, which makes such an approach not feasible in terms of computation time. These steps are therefore often carried out separately.

One of these steps is the traffic planning and path coordination of AGVs in the horizontal transportation. This can be interpreted as (dynamic) path planning, where decisions have to be taken regarding vehicle speed and traffic control, while the vehicles drive on a predetermined route. This is essential in order to avoid collisions and to maintain a stable traffic flow.

Generally, the approaches that are studied for this problem can be categorized into centralized and decentralized planning techniques. While centralized control is mostly used in current AGV systems, a trend in interest is seen towards decentralized control systems where AGVs can make decisions autonomously (De Ryck et al., 2020). In the remainder of this section, an overview is given of some studies related to traffic control in industrial AGV systems and vehicle coordination. For an extensive summary of AGV-related control algorithms, the reader is referred to the survey by De Ryck et al. (2020).

2.1 Centralized Planning

In centralized control systems, a control center is responsible for the traffic planning of the vehicles to avoid collisions and deadlocks. It can make use of global information of the environment regarding the AGV positions, their destination, and their current status. While all the information can be used, in most cases the size of the search space increases significantly with the number of AGVs. This makes it infeasible to find an optimal solution in a reasonable amount of time for real-time decision making. The following studies therefore take on a heuristic approach to solve a traffic planning problem in a centralized setting.

In the method by Ferrari et al. (1998), the planning of robots is divided into two subproblems. First, a route is determined for each robot that is free of collisions with stationary objects. Then, a schedule is computed for each route, after which they are adjusted to avoid collisions by stopping robots or by choosing detours.

A similar concept is used by Bae et al. (2008) for AGVs in an automated container terminal in a setting with 3 quay cranes and up to 24 AGVs. After the routes are created, they are divided into a series of so-called occupation areas. For each area, the entering and leaving time is computed for the corresponding vehicles, such that areas that overlap physically do not overlap in time. This scheduling is done in the order of vehicle dispatch, meaning that a route that is created earlier always has a higher priority. The authors show that this approach is able to prevent collisions and deadlocks. However, this approach is not dynamic as the planning per route is fixed once it is done.

In another approach to deal with the complexity in AGV coordination, Digani et al. (2019) choose to first partition a predefined roadmap in a warehouse into segments. The warehouse is also divided into sectors, where each sector contains at most one area where paths on the roadmap intersect. Then, the coordination is done within each sector when an agent enters or leaves an intersection area. The authors formulate it as a quadratic optimization problem where a velocity has to be determined for each segment that the vehicles traverse, such that the total traveling time is minimized. They show that the proposed method outperforms a decentralized negotiation strategy (Digani et al., 2014) in terms of average waiting time at intersections and average clearing time. However, the interaction between sectors was not taken into account. This becomes more relevant in settings where intersection areas are close to each other, which is common for AGVs on a container terminal with a dense layout.

The separation of routing and traffic coordination is also seen in the method by Zhong et al. (2020), where the routing is first done by using a Dijkstra-depth-first-search algorithm. When routes overlap, the conflict is resolved with a FCFS priority scheme, where the AGV that claims first will accelerate, while the other one slows down until it can claim again. It was shown that the waiting times of the AGVs could be reduced with this method.

Another priority-based traffic planning method is introduced by Van Vuuren (2017). Instead of using a FCFS priority rule, conflicting AGVs are first assigned a priority score that is based on the due time of their task and the type of task it is carrying out. Precedence is then given to the AGV with the highest score. In this method, the conflicts that occur are analyzed and solved individually. Compared to a FCFS priority rule, it was shown that the performance of quay cranes can be improved with a priority-based rule for AGVs.

Conflict resolution is also relevant in railway systems, as most parts of the infrastructure are shared among trains. When rerouting is not possible, a solution always includes delaying one or more involved trains, which is usually also the case for vehicles in AGV systems. Van Thielen et al. (2019) introduce a reordering heuristic for the resolution of conflicts at sections where trains require the same track. When resolving a conflict, the impact of the decision is evaluated over a fixed duration in terms of total expected delay. In the calculation, the trains that are involved indirectly are also considered to account for the impact on the other trains in the network.

2.2 Decentralized Planning

In a decentralized setting, the actions of the AGVs are not decided by a control center. Instead, an AGV makes decisions autonomously by using local information and communication with other AGVs or sensing devices in their proximity. While this makes the motion coordination problem much more scalable, it is often considered to be suboptimal as only local information is used. However, Cardarelli et al. (2017) show that the cooperation of AGVs is possible when the AGVs can individually transmit information to a cloud system and retrieve data from other AGVs. This can make decentralized control more robust and efficient. While the implementation of such methods is still in its infancy, methods in this category without cloud information sharing also show to work well.

To control the traffic of AGVs on a container terminal, the method by Evers and Koppers (1996) makes use of an abstraction of traffic lights. These limit the number of AGVs in each area. Basic priority rules are considered to prevent AGVs from colliding and getting into deadlocks within the areas.

When the order in which vehicles pass is known, it is useful to consider the way they approach and leave the intersection in terms of their velocity to optimize energy consumption. Makarem and Gillet (2012) introduce a decentralized navigation scheme to regulate the speed of vehicles that approach an intersection. Vehicles communicate their location, velocity, and path, such that their expected arrival at the intersection is used to compute the desired velocity profiles. Within this computation, a higher priority is given to heavier vehicles indirectly, as they require more time and energy to brake and accelerate. They show that their method is able to improve energy consumption and motion smoothness when compared to other centralized and decentralized methods, and a situation with traffic lights.

In a hybrid coordination procedure by Digani et al. (2014), conflicts at intersections in sectors are managed locally by combining a negotiation mechanism with a resource allocation strategy. As in Evers and Koppers (1996), the number of AGVs in a sector is set to be limited. The AGVs inside a sector share information regarding their pre-assigned priority level, their request to cross, and the status of the request, and use this in the coordination procedure. By using simulation, the authors show that their method is able to achieve conflict-free driving in a setting with up to 20 AGVs.

3 Problem Description

The automation at container terminals has brought many benefits to the efficiency of the operations. However, there is still room for improvement in the traffic control of horizontal transportation. This section gives a description of the layout of a typical container terminal and the operations that take place. It also gives a more detailed description of the horizontal transportation and the issues that may occur here. Then, a centralized planning strategy is summarized that is currently in use in the software of TBA Group. Some aspects in this strategy are pointed out that are of interest for the research in this thesis. Lastly, the research direction of this thesis is given, along with the research questions that it aims to answer.

3.1 Container Terminal Layout

The type of container terminal that is considered in this research is an LAVG-RMG terminal. This means that lift-AGVs (LAGV) are used in the horizontal transportation and Rail-Mounted Gantries (RMG) to stack containers at stacking modules on the yard. In the remainder of this thesis, a lift-AGVs will be referred to as AGVs for the sake of brevity.

The layout of the terminal that is considered, is commonly seen in practice. Here, the waterside can generally be divided into three sections: quay, apron, and storage yard. A schematic view from the side of such a container terminal is illustrated in Figure 1.

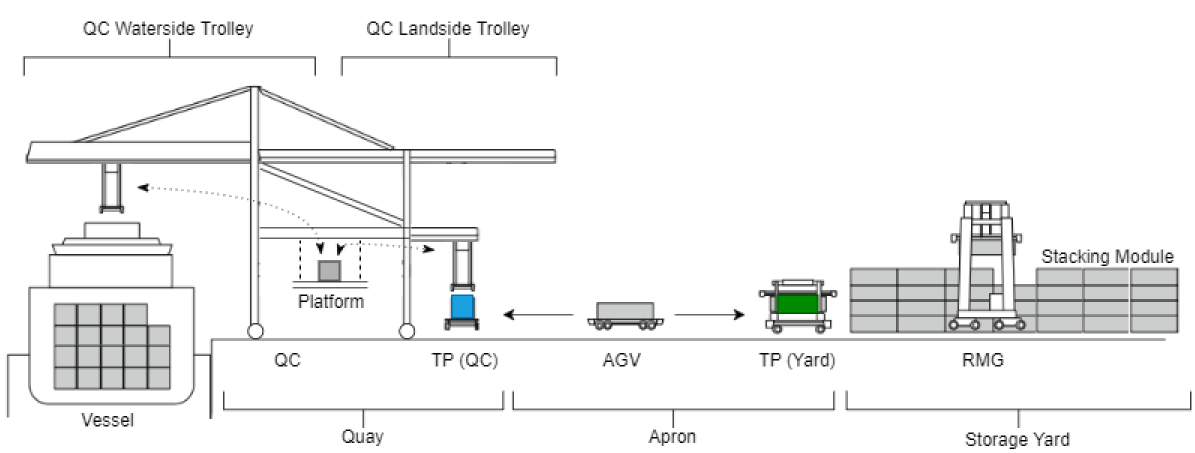
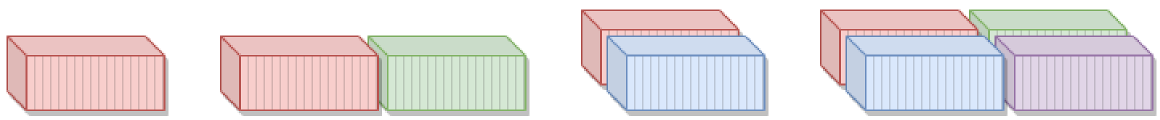


Figure 1: Schematic view of an LAGV-RMG container terminal from the side (adapted from Bae et al. (2011))

3.1.1 Quay

The quay is the area where vessels of various sizes berth to load or discharge, or both. Depending on the size of the vessel and of its cargo load, a number of quay cranes (QC) is allocated to handle the movement of containers between the vessel and the land. The operation of moving containers onto the vessel is called a load move, and moving containers out of the vessel is called a discharge move. Moves are further categorized as single, twin, tandem, and quad and are illustrated in Figure 2.



(a) Single (b) Twin (c) Tandem (d) Quad

Figure 2: Move types of a quay crane

To perform a move, the quay cranes make use of either a single trolley or double double trolley system. There are designated areas under the quay cranes on the quay, which are called transfer points (TP), where the transfer of containers takes place between the quay cranes and AGVs. Single trolley systems move containers directly between the vessel and the transfer points, whereas double trolley systems use indirect transfers with a platform between the trolleys. The waterside trolley moves containers between the vessel and the platform, and the landside trolley moves containers between the platform and AGVs at the transfer points.

3.1.2 Storage Yard

Transfer points are also located on the other side of the apron, in front of the storage yard. On this side, one or more RMGs handle the movement of containers between the transfer points per stacking module. Opposed to regular AGVs, the operations of lift-AGVs and the RMGs are decoupled. This means that the transfer points have racks where the lift-AGVs are able to drop containers for the RMGs to pick up, and where lift-AGVs can lift containers that have been placed on them by an RMG. On the opposite side of the yard, on the landside, containers are picked up or delivered by trucks or trains for transshipment.

3.1.3 Apron

The apron is an open area where the horizontal transportation between the quay and the yard takes place. Here, a fleet of AGVs follows guided paths to drive between the transfer points under the quay cranes and those at the yard. A schematic view from the top with a rough AGV path layout is illustrated in Figure 3.

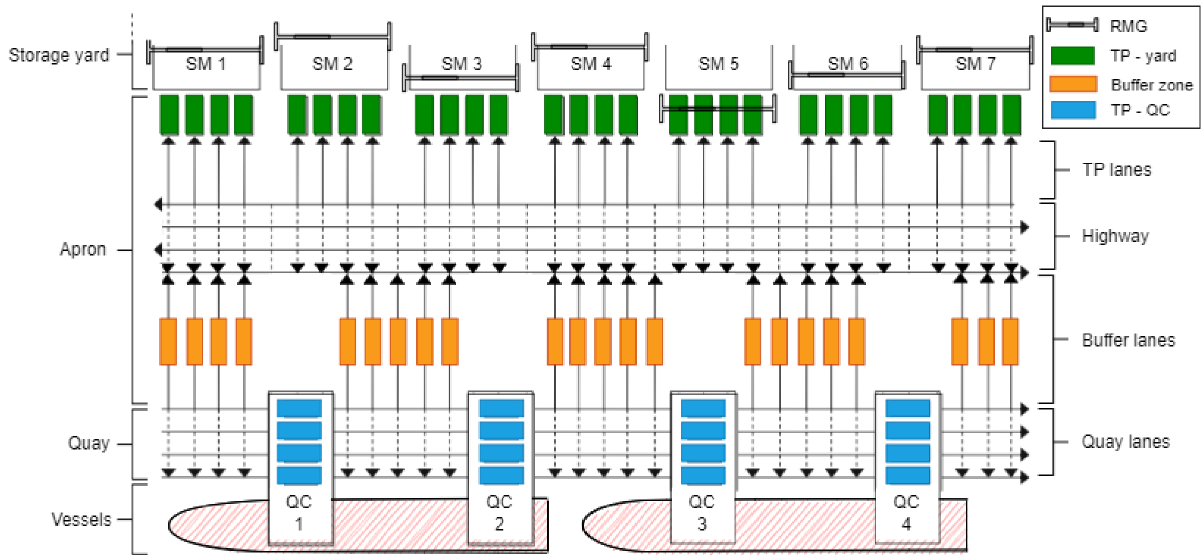


Figure 3: Common layout of an LAGV-RMG container terminal (adapted from Bae et al. (2011))

To travel along the length of the quay and the apron, there are two sets of "horizontal" lanes. The first set, the quay lanes, includes unidirectional lanes that are situated under the quay cranes, where transfer points are located on the lanes, right underneath the cranes. The other set, the highway, is located in front of the yard and also includes unidirectional lanes, but alternate in the driving direction. Transfer points at the yard are connected to the highway lanes with "vertical" TP lanes. From the quay side, the quay lanes are connected to the highway with "vertical" buffer lanes. On these lanes, the vehicles must wait at buffer zones before traveling towards transfer points at the quay or to the highway, until they receive a signal from the central control system to continue driving. Both sets of vertical lanes are bidirectional.

3.2 AGV Transport Tasks and Routing

The transport tasks that AGVs need to carry out depend on the operations that are being done by the quay cranes and consist of a pickup and a delivery sub-task. For a loading quay crane, there is a list of containers that are located in certain stacking modules at the storage yard. These containers are moved onto the racks by RMGs. Then, they must be picked up by an AGV, and delivered at one of the transfer points under the corresponding loading quay crane. These sub-tasks each have a due time that is based on the time at which the quay crane is expected to be available to perform the load-move.

For a discharging crane, the containers that it moves out of the vessel must be picked up by an arbitrary empty AGV and then brought to a rack at a specified stacking module. The due time of the pickup sub-task is based on the time at which the quay crane is expected to finish a discharge move. The due time of the delivery sub-task is based on the estimated driving time between the quay crane and the destination of the container. The way the origin and destination of containers in the yard are determined is outside the scope of this thesis.

Given the origin and destination of a transport task, a decision must be made for the allocation of the task to a vehicle, and the route that must be driven. In the literature, these steps are related to job dispatching, and vehicle routing, respectively. However, these are handled centrally by the central computer and are considered fixed in this research, such that the decisions cannot be altered afterwards once they are made. Since job dispatching and vehicle routing fall outside the scope of this research, they are only discussed briefly is this subsection.

3.2.1 Job Dispatching

Job dispatching involves the allocation of a transport task with a specified origin and destination to a vehicle in the system. Per task, the implemented job dispatching module computes a score for each vehicle that is based on the traveling distance and time from their current position to the starting position of the task. Vehicles that are idle are prioritized. Once the task has been allocated to a vehicle, a route is made for the selected vehicle to complete it.

3.2.2 AGV Routing

The way AGVs are routed depends on the organization of the highway. As mentioned earlier, the lanes on the highway are unidirectional and alternate in driving direction. The entering and leaving direction can also be decided per lane, such that the accessibility to each lane depends on the origin and destination of the AGVs.

An example of a highway infrastructure setting is given in Figure 4. In this example, the top two lanes (1, 2) are used by AGVs that travel between stacking modules. The two lanes in the middle (3, 4) are for AGVs that travel from the quay to the storage yard. Lastly, the bottom two lanes (5, 6) can be used by all AGVs that go to the quay.

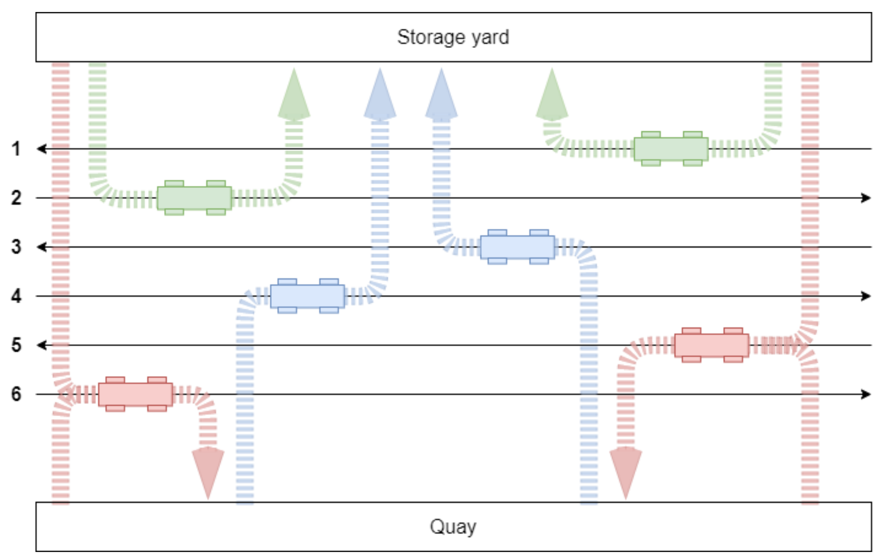


Figure 4: The layout of the highway

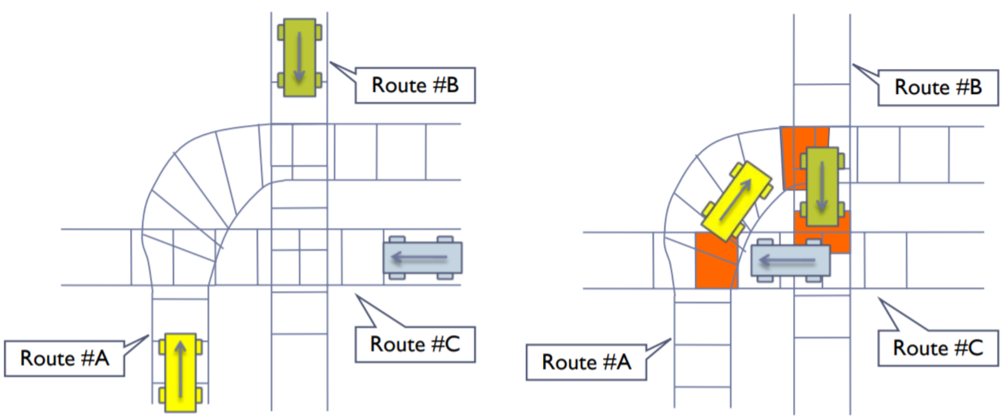
Given the origin and destination of a sub-task, and the infrastructure rules of the highway, routing of the AGVs is straightforward. Depending on the availability of the transfer points and the buffer zones, the shortest path is chosen.

3.3 AGV Driving

A route for an AGV consists of a series of actions, of which the main ones are driving straight, and driving a 90°-curve or an s-curve. Each driving action has a corresponding segment on the route. In order for an AGV to perform a driving action, it needs to send a request to claim the corresponding segment, which needs to be granted by the control center. When a segment has already been claimed by another vehicle, the request will be denied. Only when a vehicle passes its claimed segment, it becomes available again for other vehicles to claim. By default, requests to claim an available segment are always granted immediately.

The vehicles always aim to drive at the maximum speed that is allowed on their claimed segments. The total length of the claimed segments is limited, such that vehicles can only send claim requests for a certain distance ahead on their path. At any point, the speed of the vehicles is adjusted, so that they can come to a full stop before the end of their currently claimed segments. For this reason and since segments cannot be claimed by more than one vehicle, collisions are prevented.

The series of segments that correspond to curves are considered as a special case, as they must be claimed in its entirety. This prevents the vehicles from standing still on a curved section. The situation in Figure 5b is therefore not possible with this coordination strategy. This situation is called a deadlock in which vehicles block each other from moving forward in a closed chain. In the example in Figure 5, vehicle A is blocked by vehicle B, B by C, and C by A.



(a) (b)

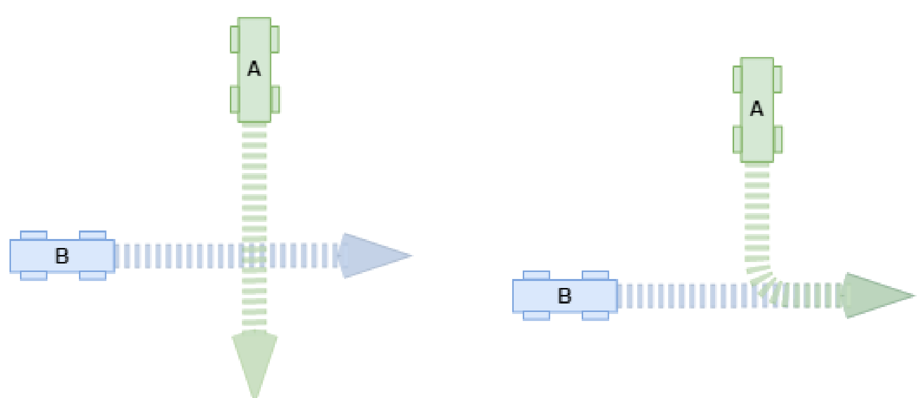
Figure 5: Deadlock (Image taken from Bae et al. (2008))

3.4 AGV Conflicts

For the AGVs, the lanes are shared among all of them and horizontal lanes intersect with vertical ones. It is therefore possible for planned routes to have segments that overlap or intersect, which is defined as a *spatial conflict*. Only when the corresponding vehicles are expected to cross these segments at approximately the same time, the segments form a conflict in terms of time as well. In the remainder of this thesis, a spatial conflict will be referred to as a *conflict*, where the aspect of the estimated crossing time of the vehicles is ignored. The corresponding route segments are referred to as the *conflict area*. When routes overlap or intersect where the corresponding segments have already been claimed by an AGV, it will not be registered as a conflict, as claimed segments cannot be unclaimed before crossing them.

On the terminal layout that is used for this research, conflicts only occur on the highway. Routes from the buffer zones to transfer points on the quay are short, such that the routes can be claimed in their entirety with a single claim request. Since claim requests never occur at the same time, a conflict (in the way that it is defined here) will never take place on the quay.

The types of conflicts are illustrated in Figure 6. In Figure 6a, the routes of the vehicles intersect, whereas in Figure 6b, the routes join paths and therefore have overlapping route segments.



(a) Intersecting routes (b) Joining routes

Figure 6: Conflict types

When a conflict is detected, a decision must be made to resolve it. More specifically, an order must be chosen in which the involved vehicles traverse the conflict area. Depending on when the vehicles are expected to arrive at the conflict area, the chosen order may cause delays for vehicles that must wait to traverse it.

This problem may seem trivial when looking at the conflicts individually, but it becomes more complex when multiple conflicts are considered with the same vehicles. The decision at one conflict influences subsequent conflicts where the same vehicles are involved. This may cause a certain secondary delay of the vehicles in the network (Bae et al., 2008; Van Thielen et al., 2019), which could potentially be lower with a different decision.

The decision of the traversing order of vehicles at conflict areas depends on the coordination strategy that is implemented. In any strategy, a deadlock should be prevented.

3.5 AGV Traffic Control

The AGV traffic control strategy that is currently used in the software of TBA Group is centralized. Within this strategy, vehicles continuously request to claim the next segments on their route while driving. The current control center works with a *first-claim-first-served* (FCFS) policy, meaning that the vehicle that requests a claim for a free segment first, is granted access on it immediately.

While this central control system performs without failure, in some situations the landside trolley of the quay cranes happens to spend a considerable amount of time waiting at the side of the transfer point. This waiting time occurs due to AGVs that arrive at the transfer points beyond the due time of their transport task. Ideally, every AGV arrives before the due time of its transport task to prevent cranes from waiting. This increases the quay crane productivity, which is often used as a key performance indicator.

The arrival time of AGVs at transfer points is affected by the traffic condition and the traffic control strategy. A congested highway leads to more conflicts and may lead to more delays per vehicle. A FCFS policy does not take late transport orders into account. It is therefore possible for AGVs with such orders to be delayed further if they do not receive precedence when facing conflicts.

3.6 Research Direction

The main goal of this research is to analyze the relation between quay crane productivity and the lateness of transport orders that are carried out by AGVs. To do so, a method is introduced that aims to minimize the lateness that is caused by conflicting AGVs. That leads to the following research questions:

- *Is it possible to decrease the lateness of AGV transport orders by using a traffic control strategy that is not based on a first-claim-first-served policy?*
- *How does the lateness of AGV transport orders affect the waiting time and productivity of quay cranes?*

4 Methodology

In this section a centralized coordination strategy is introduced for the conflict resolution of AGVs on a container terminal. The approach is applied in a highly developed discrete event simulation model of TBA Group. In the implementation, some components within the model are changed and modules are added. The details thereof are given in this section.

First, the modifications and additions to the simulation model are clarified. Then, details are given for a module that monitors conflicts between AGVs in the system. Afterwards, the approach for resolving conflicts is elaborated, which includes a mathematical formulation for AGV scheduling at conflict areas. Lastly, a feature is introduced that incorporates priority levels into the approach.

4.1 Simulation Model Modifications and Additions

As groundwork for the introduced conflict resolution strategy, some components in the simulation model are added or changed. These are elaborated in this subsection.

4.1.1 Route segmentation

As explained in Section 3, a route is divided into several segments. By default, these segments are up to 18 meters long, which is the length of an AGV. To define a conflict area in a finer way, these segments are set to have a maximum length of 2 meters. An exception is given for segments that belong to curved sections, as these segments cannot be claimed in parts.

4.1.2 Trailing AGVs

When AGVs are driving on the highway, it is possible for them to drive behind one another. Consequently, these AGVs can come into a conflict with another AGV together. The AGVs that are driving in succession must pass common conflict areas in the order that they are driving, as overtaking is not possible.

To ensure that this is not violated in the conflict resolution module, a module is added that keeps track of overlapping segments and another that uses this module to register any leading and trailing AGVs. A flowchart of how leading and trailing AGVs are registered is given in Figure 7. Here, `Lead_trail` is a square matrix with binary values with a size equal to the number of active AGVs. The value of an element `Lead_trail[AGV i, AGV j]` is `TRUE` when `i` is driving in front of `j`, and `FALSE` otherwise. The default values are set to `FALSE`. Furthermore, the values for `Lead_trail[AGV i, AGV j]` and `Lead_trail[AGV j, AGV i]` are set to `FALSE` for all AGVs `j` when an AGV `i` reaches the end of its route.

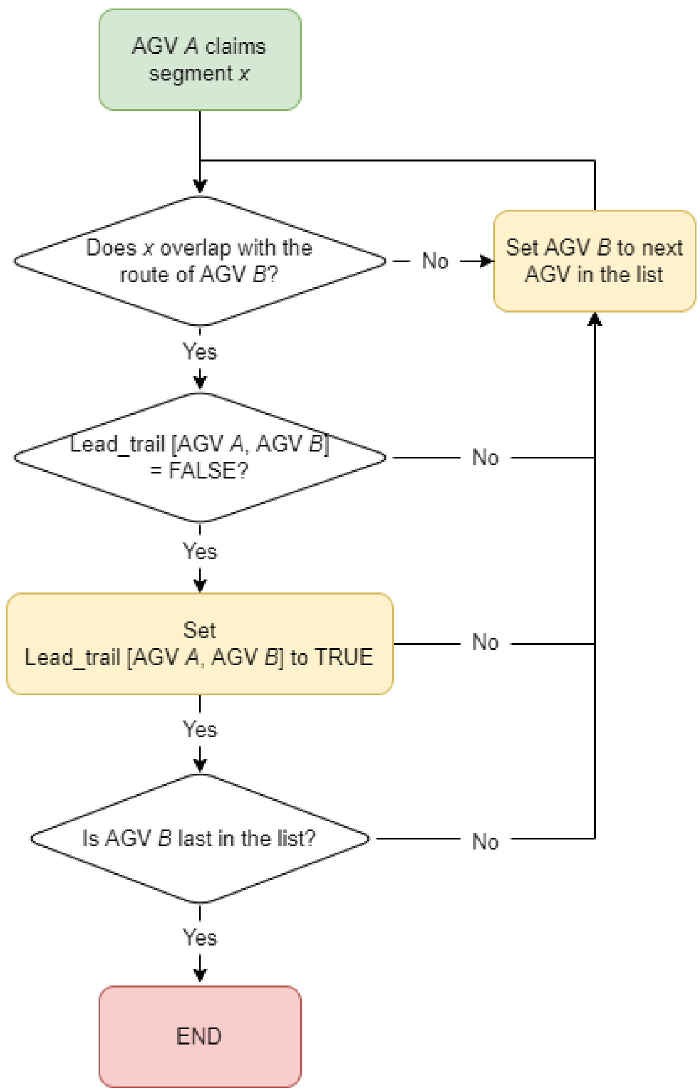


Figure 7: Flowchart for registering leading and trailing AGVs

4.1.3 Time Estimation

To coordinate conflicting vehicles during the simulation, it is of essence to have an accurate estimation of the time that the vehicles reach and leave the conflict areas. To do so, a time estimation module is added that gives an approximation of the time a vehicle enters and leaves each segment of its route. This approximation is based on the average speed that a vehicle with a medium load passes through each section of the route (curves, before and after curves, and straight sections). It is assumed that the vehicle can accelerate to the maximum speed that is allowed on the sections without interruptions. For vehicles with a higher load, the maximum speed is set to be lower.

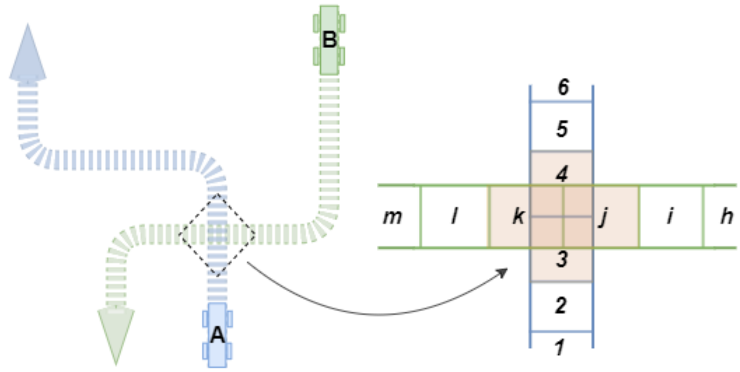
While a vehicle is driving, the estimated times of the remaining segments are updated dynamically. Once a vehicle passes through a segment, the difference is taken between the actual time and the estimated time of leaving the segment and applied to the estimated times of all subsequent path segments.

4.2 Monitoring Conflicts

In order to perform efficient conflict resolution that is not based on trivial rules, conflicts must be monitored continuously. To do so, a conflict registration module is built in the simulation model that keeps track of upcoming conflicts in real time.

4.2.1 Conflict Detection and Registration

During the simulation, when a new route is assigned to a vehicle, the simulation model verifies whether it has conflicts with other existing routes individually. The details of the conflicts per involved vehicle, such as the route segments in the conflict area and the expected times of entering and leaving the area, are determined by an added conflict registration module. An example of a registered conflict in two intersecting routes is illustrated in Figure 8. In this example, when the route of vehicle B is added, a conflict is detected with vehicle A and given the ID *"Diamond"*. Per vehicle, their conflicting path segments are registered as the conflict area, along with the estimated time the corresponding vehicle will enter and leave the area.



(a) Segments that form a conflict area (highlighted)

Conflict Table

Conflict	Involved Vehicles
Diamond	A, B

Vehicle Tables

Conflicts	Segments	Est. Time In	Est. Time Out
Diamond	3, 4	5	6

Conflicts	Segments	Est. Time In	Est. Time Out
Diamond	j, k	10	11

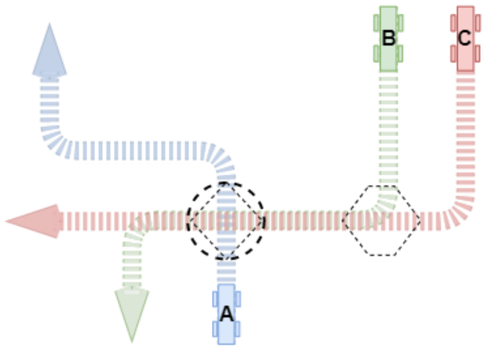
(b) Conflict registration

Figure 8: Example of a registered conflict in intersecting routes

When two routes have an overlapping section on the same lane, only the initial part of the shared section is registered as a conflict area, as in the dotted hexagon in Figure 9a.

4.2.2 Conflict Merging

Since conflicts are registered sequentially per two vehicles, it may occur that a new conflict is registered where the path segments in the conflict area of one AGV overlap with those of a different AGV in another conflict area. In this case, the conflict areas should be merged in order to prevent inconsistencies in the assigned traversing orders. An example of two conflicts being merged is given in Figure 9. Here, the segments of vehicle B in conflict *Diamond* overlap with segments of vehicle C in conflict *Circle*. Because of this overlap, the conflicts *Diamond* and *Circle* are merged into one conflict, that is given the ID “X” in this example.



(a) Conflicts that are present

Conflict	Involved Vehicles
Diamond	A, B
Circle	A, C
Hexagon	B, C

Conflict	Involved Vehicles
X	A, B, C
Hexagon	B, C

(b) Conflict registration and merging

Figure 9: Example of merging conflicts

4.3 Conflict Resolution

Given the conflicts that are present during the simulation, the task of the conflict resolution module is to find an efficient order in which the AGVs traverse the conflict areas. The method that is introduced in this subsection for this task is inspired by various methods that are discussed in Section 2.

4.3.1 AGV Scheduling at Conflict Areas

The proposed approach involves path scheduling for AGVs and is inspired from the method used by Bae et al. (2008). Instead of performing AGV scheduling for every part of a path, this research limits it to the scheduling at path segments that form a conflict area. A decision for one conflict may lead to a chain reaction in the system, causing more delays for the AGVs. This effect is called *delay propagation* (Van Thielen et al., 2019) and is taken into account while resolving conflicts. The measure that is therefore used is the total estimated delay of all conflicting AGVs that follows from the results of the scheduling algorithm.

To set up a mathematical formulation, the following notation is used:

Sets Given all registered conflicts and the involved AGVs, sets are defined as follows:

- $\mathcal{V}$: set of all AGVs v that have at least one conflict,
- $\mathcal{C}$: set of all conflicts c,
- c_v^*: the first conflict that AGV v will cross,

- $\mathcal{C}_v \subseteq \mathcal{C}$: set of conflicts where AGV v is involved,
- $\mathcal{F} \subseteq \mathcal{C}$: set of conflicts where the driving order is fixed due to trailing AGVs,
- $\mathcal{J} \subseteq \mathcal{C}$: set of conflicts where at least two AGVs join paths or where they join paths at an earlier conflict, with a common subsequent conflict area,
- $\mathcal{H} \subseteq \mathcal{J}$: set of conflicts where at least two of the involved AGVs join paths before the conflict area,
- $\mathcal{V}_c \subseteq \mathcal{V}$: set of AGVs that are involved in conflict c,
- $\bar{\mathcal{V}}_c \subseteq \mathcal{V}_c$: set of AGVs that will join paths at conflict c, or at another conflict before c,
- $\mathcal{U}_c \subseteq \mathcal{V}_c$: set of AGVs that already drive in front of at least one AGV on the shared path before conflict c,
- $\mathcal{W}_{v,c} \subseteq \mathcal{V}_c$: set of AGVs that already drive behind AGV v on its shared path before conflict c.

Parameters When the conflict resolution module is called, some calculations are done regarding the driving time at certain parts of the track and the priority level of each AGV. These are defined as parameters:

- t_v^*: the estimated minimum time that is needed for AGV v to reach its first conflict area,
- p_c: the estimated minimum time that is needed to pass conflict c,
- $t_{c',c}$: the estimated minimum time that is required to arrive at conflict c that follows after passing the previous conflict c',
- e_{v,l_v}: the estimated earliest possible time for AGV v to reach its last conflict l_v,
- δ: the minimum driving time that is required between a leading and a trailing AGV.

Since the duration of crossing a conflict, p_c, is not necessarily equal for every AGV, the average is taken of the estimated duration of each AGV that is in conflict c. The value of δ is set to 3 seconds.

Auxiliary variable The following auxiliary variables are defined for AGVs that drive in succession after an unresolved conflict as, for example, AGVs B and C in Figure 9, and for every subsequent common conflict that is not their last.

- $\forall c \in \mathcal{J}$:

$$b_{v,w,c} = \begin{cases} 1, & \text{if AGV } v \text{ passes potential conflict area } c \text{ before AGV } w, \\ 0, & \text{otherwise.} \end{cases}$$

In the example in Figure 9, this variable is defined for AGVs B and C at conflict "*Hexagon*", as they join paths after this conflict, while they have a common subsequent conflict, X. Since conflict X is their last common conflict, this variable is not defined for these AGVs at this conflict.

Decision variable

- $s_{v,c}$: the time at which AGV v starts to pass the conflict area of conflict c.

Mathematical formulation Using the notation that is described above, the AGV scheduling problem at conflict areas can be formulated as the following mixed integer programming problem:

$$\min \quad \sum_{v \in V} s_{v,l_v} - e_{v,l_v}, \tag{1}$$

$$\text{s.t.} \quad s_{v,c} \geq t^*_v, \qquad \forall v \in \mathcal{V}, \quad c = c^*_v, \tag{2}$$

$$s_{v,c} \geq s_{v,c'} + p_{c'} + t_{c',c}, \qquad \forall v \in \mathcal{V}, \quad c \in \mathcal{C}_v \setminus c^*_v, \tag{3}$$

$$|s_{v,c} - s_{w,c}| \geq p_c, \qquad \forall c \in \mathcal{C}, \quad v, w \in \mathcal{V}_c, \quad v \neq w, \tag{4}$$

$$s_{w,c} \geq s_{u,c} + \delta, \qquad \forall c \in \mathcal{F}, \quad u \in \mathcal{U}_c, \quad w \in \mathcal{W}_{u,c}, \tag{5}$$

$$s_{w,c} \geq s_{v,c} - M(1 - b_{v,w,c}), \qquad \forall c \in \mathcal{J}, \quad v, w \in \bar{\mathcal{V}}_c, \quad v \neq w, \tag{6}$$

$$b_{v,w,c} + b_{w,v,c} = 1, \qquad \forall c \in \mathcal{J}, \quad v, w \in \bar{\mathcal{V}}_c, \quad v \neq w, \tag{7}$$

$$s_{w,c} \geq s_{v,c} + \delta - M(1 - b_{v,w,c'}), \qquad \forall c \in \mathcal{H}, \quad v, w \in \bar{\mathcal{V}}_c, \quad v \neq w, \tag{8}$$

$$s_{v,c} \in \mathbb{R}^+, \qquad \forall v \in \mathcal{V}, \quad c \in \mathcal{C}_v, \tag{9}$$

$$b_{v,w,c} \in \{0, 1\}, \qquad \forall c \in \mathcal{J}, \quad v, w \in \bar{\mathcal{V}}_c, \quad v \neq w. \tag{10}$$

The objection function (1) is the total estimated delay, where the delay of an AGV is computed as the difference between the estimated time of reaching the last conflict area without interruptions and with the interruptions of the conflicts. Constraint (2) takes the driving time into account from the starting point of each AGV to their first conflict area. Constraint (3) ensures that the earliest time to pass the next conflict area cannot be earlier than the time of passing the previous one plus the driving time between the conflict areas. Constraint (4) ensures that two AGVs do not enter a conflict area at the same time. To ensure the driving orders of leading and trailing AGVs are consistent in consecutive conflicts, Constraint (5) is added. Constraints (6) and (7) are added to ensure that the auxiliary variable is defined properly whenever AGVs drive behind each other at conflict areas where the order is initially undecided. Since overtaking should not be possible, Constraint (8) ensures that an AGV that drives behind another AGV enters a shared conflict area after the AGV in front of it has passed it. Lastly, Constraints (9) and (10) state the domain of the decision variable and the auxiliary variable, respectively.

4.4 Priority Weights

Quay cranes lose productivity when they must wait for an AGV to arrive to either load or discharge. Since this should be avoided, AGVs that are delayed should be given priority when they face conflicts. AGVs that are ahead on schedule can afford to be delayed, as arriving too early means that they need to wait to be handled by the quay crane. To take this into account in the traffic control, the objective function (1) is changed such that the estimated acquired delay per AGV is scaled with a weight factor ω_v that is based on a priority level of the AGV:

$$\sum_{v \in V} \omega_v (s_{v,l_v} - e_{v,l_v}), \tag{11}$$

The calculation of the priority weight for an AGV is based on the formula used in Van Vuuren (2017). In this formula, two factors are taken into account and multiplied with each other to compute a final score: an urgency level of the AGV and a score based on the type of task it is carrying out:

$$\omega_v = u_v \cdot \tau_v \tag{12}$$

To compute the urgency level of an AGV v, its slack time is determined first as in Equation (13), which is computed as the difference between the due time of its current task, d_v, and the expected time of arrival at the destination of the task, α_v. Then, the urgency level is computed as in Equation (14). This level increases when the amount of slack decreases.

$$\sigma_v = d_v - \alpha_v \tag{13}$$

$$u_v = \begin{cases} 1, & \text{if } \sigma_v < 6.67, \\ \frac{20}{10+\sigma_v} - 0.2, & \text{if } 6.67 \leq \sigma_v < 89.5, \\ 0.001, & \text{otherwise.} \end{cases} \tag{14}$$

Depending on the type of task an AGV is carrying out, the urgency is scaled further. These type of tasks are described in Section 3.2. Additionally, an AGV can go to the battery station to swap its battery whenever that is needed. In Equation (15), different scores are assigned to an AGV depending on its task. An AGV that is performing a task for a loading quay crane receives a higher score than one that is performing tasks for a discharging quay crane. This is done as the order in which containers are loaded into the vessel needs to be according to a pre-specified sequence. Therefore, the AGVs that deliver these containers must also arrive at the quay crane in this given order. For a discharging quay crane, AGVs that are scheduled to pick up the discharged containers can arrive in an arbitrary sequence.

$$\tau_v = \begin{cases} 1, & \text{if } v \text{ is driving to QC for delivery,} \\ 0.8, & \text{if } v \text{ is driving to stack for pickup,} \\ 0.6, & \text{if } v \text{ is driving to QC for pickup,} \\ 0.01, & \text{if } v \text{ driving to stack for delivery,} \\ 0.01, & \text{if } v \text{ is driving to a battery station.} \end{cases} \tag{15}$$

The relation between the slack and the priority weight is shown in Figure 10, where the weight is plotted against the amount of slack for the different types of tasks.

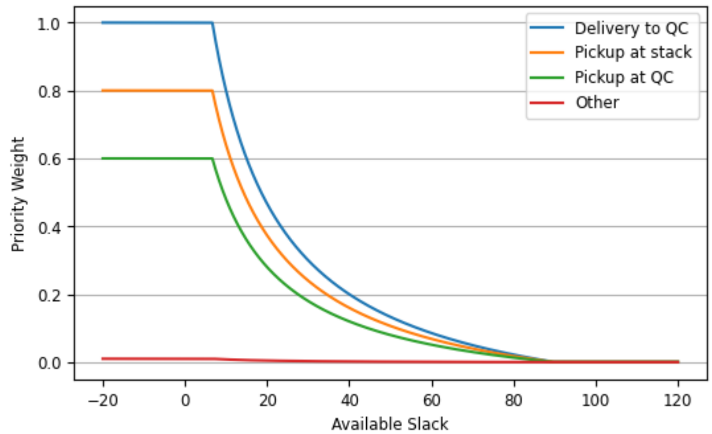


Figure 10: Priority weight plotted against the available slack of AGVs with different tasks

5 Simulation Setup and Results

To evaluate the coordination strategy for traffic control that is introduced in the previous section, several experiments are conducted within a simulation environment. In doing so, the coordination strategy that is currently in use by TBA Group is used as a benchmark. This section gives a brief description of how the simulation experiments are set up, and of the different scenarios that are used within these experiments. Then, the results are presented per scenario.

5.1 Experiment Settings

The simulation software that is used in this thesis is provided by TBA Group and is able to model operations on a container terminal in great detail. It uses discrete event simulation, meaning that is keeps track of a list of all upcoming events, such as vehicle or crane movements, along with the time that they occur. Stochasticity is incorporated in for example the actions of quay cranes, such as the move types and operation speed, and the containers for discharging and loading, which can vary in various properties. To acquire comparable and reliable results that are not heavily influenced by stochasticity, it is necessary to have a sufficient amount of simulation running hours per scenario, divided in different replication runs.

An experiment consists of a number of replication runs, using a fixed scenario over the these runs. For the stochasticity, a different seed is used at the start of each replication run. A scenario describes the settings on the container terminal in terms of the quay cranes, AGVs, and the used AGV coordination strategy. Given a scenario, a single replication run amounts to 8 hours of simulation time. The first hour of each replication is regarded as a warm-up period to initialize the processes, such that the corresponding data is discarded. With twenty replications for a single scenario, an experiment will have 140 simulated hours.

5.2 Scenarios

As stated in Section 3, the type of terminal that is considered is an LAVG-RMG terminal. A snapshot of the terminal in the simulation is shown in Figure 11. Across the experiments, the number of stacking modules, and all the lanes are fixed. Each stacking module is operated by two RMGs. There are six parallel highway lanes below the stacking modules with an alternating driving direction and four parallel non-alternating unidirectional (left to right) quay lanes underneath the quay cranes. The infrastructure rules that are used for the highway are as described in Section 3.2.2 and Figure 4. On both sides of the stacking yard, there are battery charging stations for the AGVs.

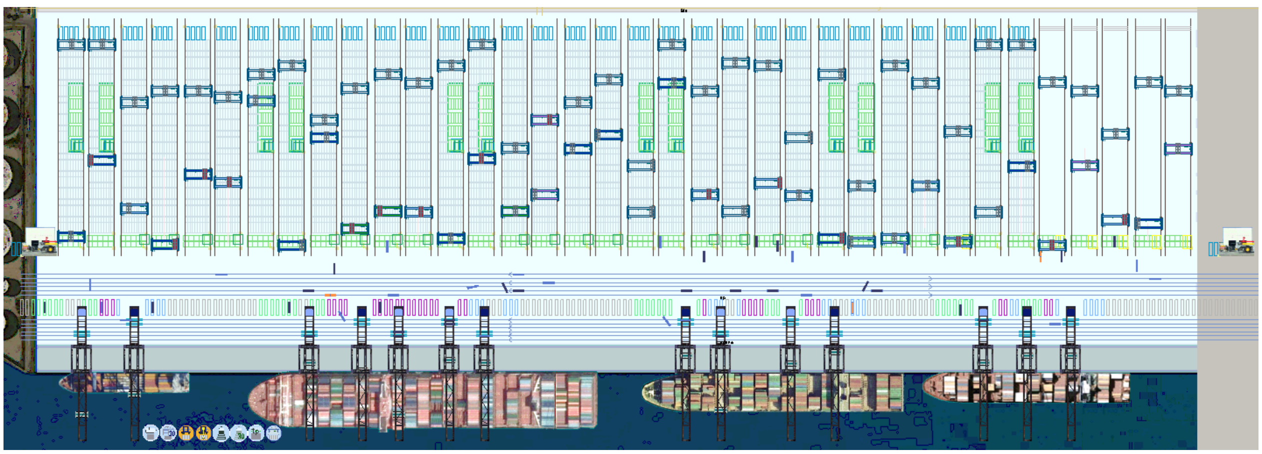


Figure 11: Snapshot of the container terminal in the simulation model

The settings that vary across the experiments include the selection of quay cranes, the fleet size of AGVs, and the conflict resolution strategy. In total there are fourteen quay cranes, which are numbered from 1 to 14 from left to right in Figure 11. The main scenarios are described in Table 1.

Table 1: The main scenarios for the simulation experiments

Name	Number of QCs	Selected QCs	Number of AGVs	AVG/QC ratio
4QC-20AGV	4	4-7	20	5
10QC-40AGV	10	2-11	40	4
14QC-50AGV	14	1-14	50	3.57

This table shows which quay cranes are set to be active and the number of active AGVs per main scenario. Additionally, the ratio of the number of AVGs per quay crane is given.

Within the main scenarios, sub-scenarios are defined which describe the conflict resolution method that is used. The names of these sub-scenarios are given in Table 2 and are used throughout this section for the sake of brevity.

Table 2: The sub-scenarios for the simulation experiments

Name	Conflict resolution
FCFS	First-claim-first-served (benchmark)
VS-NW	AGV scheduling - no weights
VS-PW	AGV scheduling - with priority weights

5.3 Connection with the Conflict Resolution Module

The computations of AGV scheduling in the conflict resolution module are executed in Python on an external hardware with a Intel Core i7-4800MQ CPU at 2.70GHz, with 16GB of RAM. In this thesis, a commercial optimization software package, CPLEX (IBM, 2020) is used.

The simulation model and the conflict resolution module communicate through a shared database. During the simulation, the module is triggered whenever a new route is assigned to an AGV that creates new conflicts. The simulation model is then paused until a solution is returned. First, given the conflicts that are present at that time and the position of all involved AGVs, an instance is created in the simulation model that is sent to the conflict resolution module. In doing so, per AGV the list of conflicts is sorted according to the estimated time the AGV enters them. This is necessary for the construction of Constraint (3) and (8).

When the instance is created, it is saved in the database, which triggers the conflict resolution module to read the instance and to build and solve a optimization model accordingly for the AGV scheduling problem at conflict areas (1)-(10). From the solution, the order in which the AGVs traverse each conflict area is derived and sent to the database, such that it can be read by the simulation model.

When the traversing order has been read, the claiming behavior of the AGVs is adjusted accordingly. When an AGV sends a request to claim a segment that is part of a conflict area, it is only granted when the AGV is the next to traverse the conflict area according to the determined order. Once the request has been granted, the AGV is removed from the conflict.

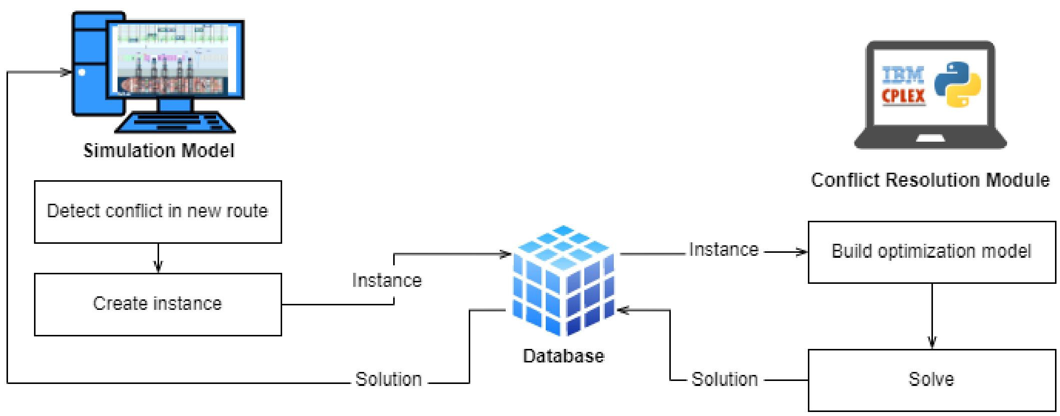


Figure 12: Communication between the simulation model and the conflict resolution module

5.4 Results

In this subsection, the results are shown per main scenario of the simulation experiments. Per main scenario, the situation in which a first-claim-first-come rule is applied is used as a benchmark to evaluate the introduced conflict resolution strategies. In order to make a fair comparison of the average quay crane productivity level in terms of number of containers moved per hour, the distribution of quay crane move types should be similar across the sub-scenarios. Figure 19, 20, and 21 in Appendix A.1 show this distribution across all replications, where no significant differences can be seen. Furthermore, confidence intervals are constructed with a 95% confidence level using the Student's t-distribution to account for the uncertainty when comparing the average quay crane productivity levels per experiment:

$$CI = \hat{q} \pm\ t^*\ \frac{s}{\sqrt{n}}, \tag{16}$$

where:

- $\hat{q}$ = the sample mean of the quay crane net productivity (containers moved per hour) in the experiment,
- t^* = the critical value of the Student's t-distribution,
- s = the sample standard deviation,
- n = the number of hours in the experiment.

5.4.1 General Results

To perform the experiments, several CPUs are run in parallel to save time. The average run time of one replication run for each main- and sub-scenario is given in Table 3. It is shown that the run time increases greatly when the proposed traffic control strategy is in use. When comparing the run time for the benchmark scenario in 4QC-20AGV and 10QC-40AGV, it can be argued that the increase in run time is attributed to a finer segmentation of the routes. This feature increases the number of computations in specific functions significantly that were already present in the original simulation model. Evidently, a scenario with more AGVs and active quay cranes lead to a longer run time as well.

Table 3: Average run time of one replication for each sub-scenario per main scenario

Main scenario	Sub-scenario		
	FCFS	VS-NW	VS-PW
4QC-20AGV	4hr 49min*	5hr 25min	5hr 31min
10QC-40AGV	1hr 49min**	18hr 53min	18hr 45min
14QC-50AGV	2hr 39min**	18hr 13min***	18hr 17min***

This table shows the average run time of one replication for each of the sub-scenarios in the main scenarios. *In these runs, the routes were segmented into parts of 2 meters. **In these runs, the route segments were not further divided. ***These replications were run on a newer and faster version of the simulation model software.

Before the simulation experiments were run, preliminary results had shown that the conflict resolution module could fail to build an optimization model when inconsistencies were found in the given instances. More details are given in Appendix A.1. When building the optimization model had failed, the simulation model continued with the first-claim-first-served rule until the module was able to provide a solution from following instances. The average number of times that the module was triggered per hour in the simulation is shown in Table 4, along with the number of times it succeeded and failed to return a solution. With more vehicles, the module is triggered more often, where the increase does not seem to be exponential. However, the module often fails to build a model properly in a scenario with more AGVs, as it is more likely to have a situation as in Figure 22.

Table 4: Statistics of the conflict resolution module

Main scenario	Average #triggers/hour	Average #failed/hour	Average #solved/hour	Average % solved/hour
4QC-20AGV	311	4	307	98.8
10QC-40AGV	736	115	621	84.4
14QC-50AGV	965	362	603	62.5

This table shows statistics per main scenario regarding the conflict resolution module during one hour in the simulation model.

For the times that the module was able to build the optimization model and to return a solution, the average real time that was needed to do so is given in Table 5. Furthermore, some statistics are given regarding the size of the instances in Table 6. It is shown that the instances increase in size for busier scenarios, while the computation time remains reasonable.

Table 5: Statistics of solving the AGV scheduling problem within the conflict resolution module

Main scenario	Average model build-time (s)	Average model solve-time (s)	Average total duration (s)
4QC-20AGV	0.016	0.040	0.056
10QC-40AGV	0.048	0.092	0.140
14QC-50AGV	0.079	0.100	0.179

This table shows statistics per main scenario regarding the average real time that was required for the conflict resolution module to return a solution in the simulation model.

Table 6: Statistics of instances given to the conflict resolution module

Main scenario	#Unique AGVs	#Conflicts	#AGVs per conflict	#Conflicts per unique AGV
4QC-20AGV	5.2	4.0	2.5	1.9
10QC-40AGV	12.9	12.8	2.7	2.7
14QC-50AGV	17.8	18.6	2.9	3.0

This table shows average values per main scenario regarding the size of the instances that were created for the conflict resolution module.

5.4.2 4QC - 20AGV

The first scenario that is considered is rather small and makes use four quay cranes and twenty lift-AGVs. Figure 13 shows the distribution of the driving status of AGVs, averaged over all active AGVs. On average, scheduling AGVs at conflict areas leads to AGVs spending more time standing still to wait for a claim request to be granted. Moreover, the AGVs within those sub-scenarios have to brake and accelerate more often, such that they drive with a constant speed less often. A reason for this occurrence could be that AGVs with a lower priority have to wait at conflict areas more often and for a longer time. This loss in driving constancy for these AGVs is traded off for a gain for AGVs with a higher priority, where the loss weighs more in the trade-off.

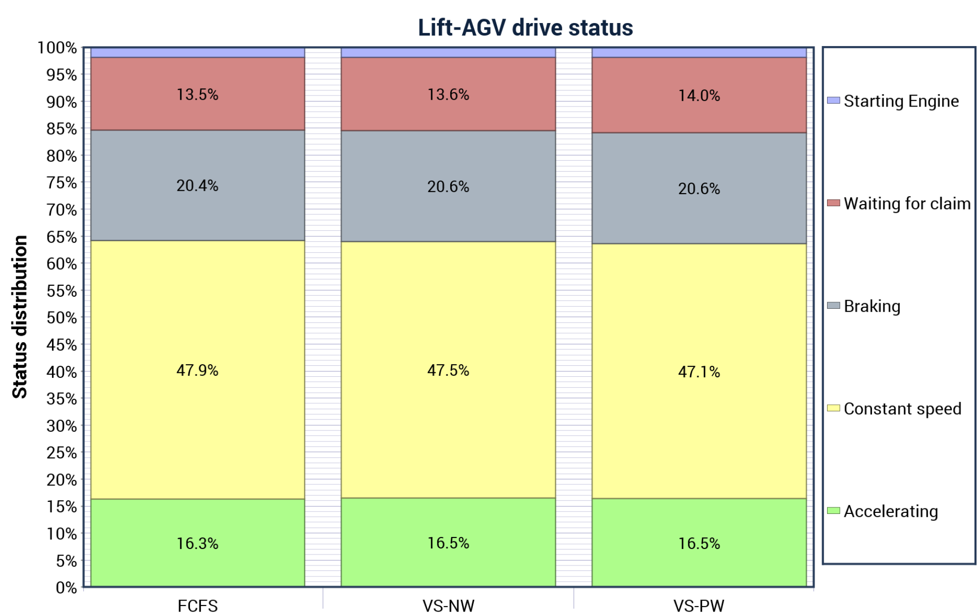


Figure 13: Average AGV driving status distribution (4QC-20AGV)

Table 7 shows the lateness per AGV transport task type that is averaged across all the replication runs in the experiment. Compared to the benchmark method, AGV scheduling without weights does not seem to improve the lateness. The only improvement is seen in the value at the 95^{th} percentile, which is only a few seconds less than that of the benchmark. Although minor, the method with priority weights does show an improvement for the tasks with higher priority that are associated with quay cranes performing loading operations. This improvement is traded off for slightly later arrivals of the remaining tasks that are associated with discharging quay cranes.

Table 7: Average lateness and the 95^{th} percentile, given per AGV transport task type (4QC-20AGV)

	Task type			
Sub-scenario	Delivery - QC	Pickup - Stack	Pickup - QC	Delivery - Stack
FCFS	3.26 (7.60)	1.97 (5.36)	-1.43 (0.46)	-2.81 (-0.38)
VS-NW	3.31 (7.54)	1.96 (5.41)	-1.35 (0.50)	-2.65 (-0.02)
VS-PW	3.21 (7.15)	1.87 (5.32)	-1.24 (0.71)	-2.57 (-0.05)

This table shows for the experiments in main scenario 4QC-20AGV the average value and between parentheses the 95^{th} percentile of the lateness of every completed AGV transport task over the whole experiment. The values are given in minutes.

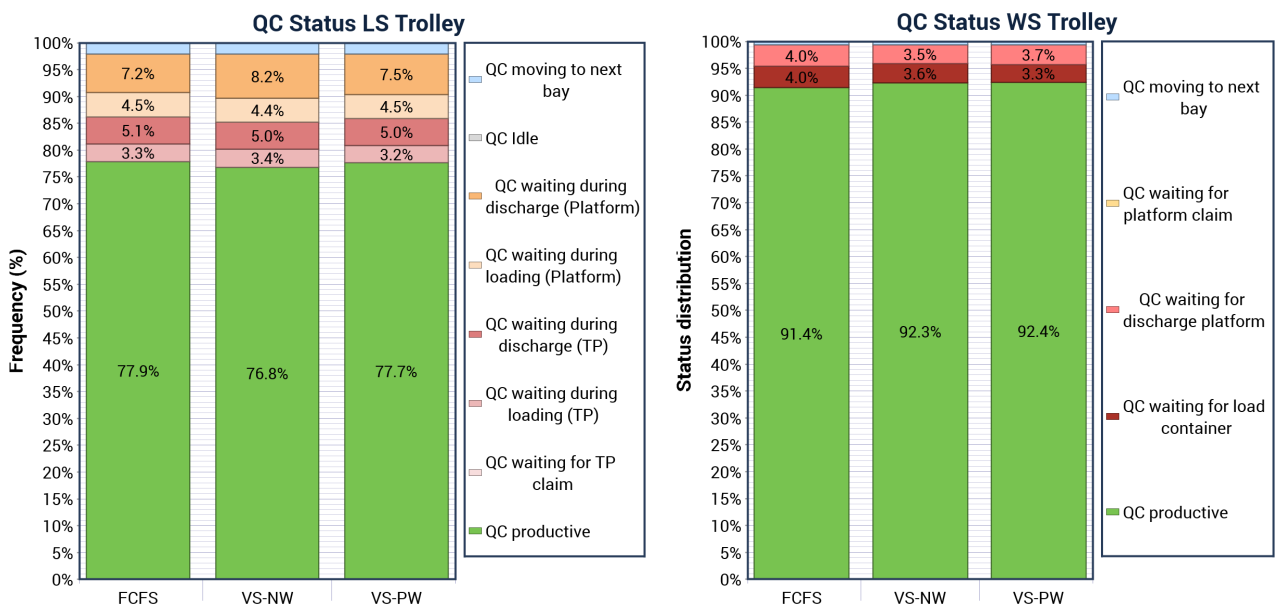
Table 8 shows the quay crane productivity in terms of the number of containers moved per hour, averaged over all replications for each sub-scenario. The values seem to show a minor difference between the sub-scenarios, which is not significant according to the computed confidence intervals.

Table 8: Average quay crane net productivity (4QC-20AGV)

Sub-scenario	Average QC productivity (containers/hour)	Diff. with benchmark	95% Confidence interval (half-width)
FCFS	47.18	-	0.72
VS-NW	46.63	-0.55	0.74
VS-PW	46.97	-0.21	0.72

This table shows for the experiments in main scenario 4QC-20AGV (1) the average net productivity in terms of the number of containers moved per hour of all active quay cranes during the whole experiment, (2) the difference in average productivity with that of the benchmark sub-scenario, and (3) the half-width of the corresponding 95% confidence interval ($t^* \frac{s}{\sqrt{n}}$ in Equation (16)).

To further analyze the productivity of the quay cranes, the status distribution of the landside trolley and the waterside trolley are shown in Figure 14a and 14b, respectively. The slight decrease in the percentage of time that a landside trolley is waiting at the side of the transfer point during loading could be a consequence of having slightly earlier arrivals of laden AGVs at quay cranes. However, since the difference is minor, this cannot be said with certainty. In Figure 14b, it can be seen that the waterside trolleys have a productivity rate that is near maximum. This means that a better performance of the landside trolleys, and hence earlier arrivals of AGVs at transfer points, cannot lead to a significant increase in the performance of waterside trolleys and therefore in the overall quay crane productivity.



(a) (b)

Figure 14: Quay crane status distribution per trolley (4QC-20AGV)

5.4.3 10QC - 40AGV

To create a scenario where there is space for quay cranes to gain in productivity, more vehicles and quay cranes are deployed. The next main scenario makes use of ten quay cranes and fourty AGVs, such that the ratio of the number of AGVs per quay crane is lower than in the previous scenario. With more AGVs in the system, there is also more congestion taking place. This can be derived from the driving status distribution of AGVs, which is shown in Figure 15. Compared to the previous scenario, AGVs spend more time waiting for a claim to be granted. They also spend less time driving a constant speed, whereas the percentage of braking and accelerating stay more or less the same. These two observations give an indication that an AGV stands still during a task for a longer time compared to the previous scenario, creating a situation with more congestion.

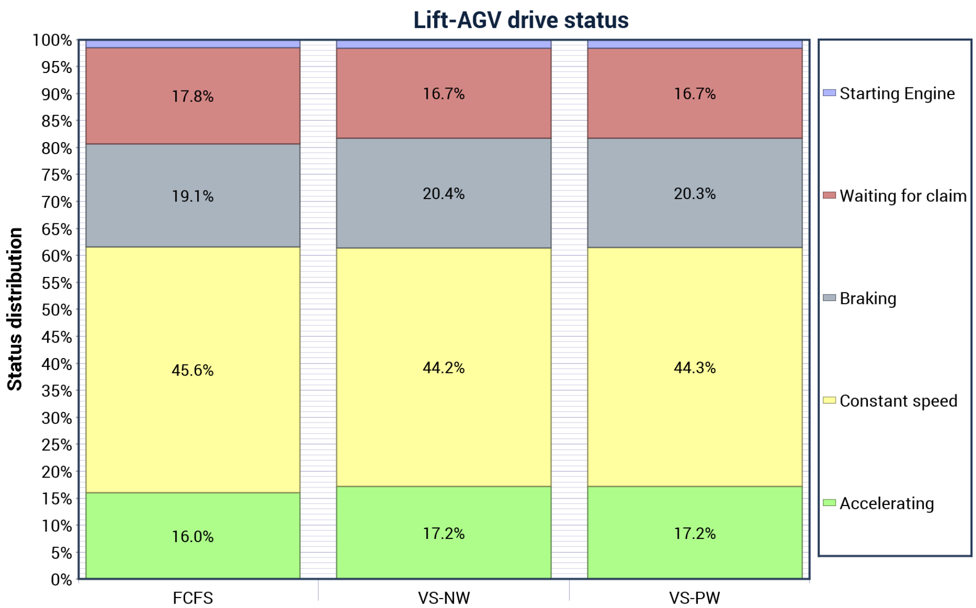


Figure 15: AGV driving status distribution (10QC-40AGV)

Figure 15 also shows that on average, AGVs spend less time standing still in the experiments where an AGV scheduling method is used, compared to the experiment of the benchmark. While this is an improvement, the driving speed seems to be more irregular with more decelerating and accelerating, rather than driving with a constant speed. This could be an indication that there are more situations where AGVs have to slow down more often to let other AGVs pass, while they do not always come to a full stop.

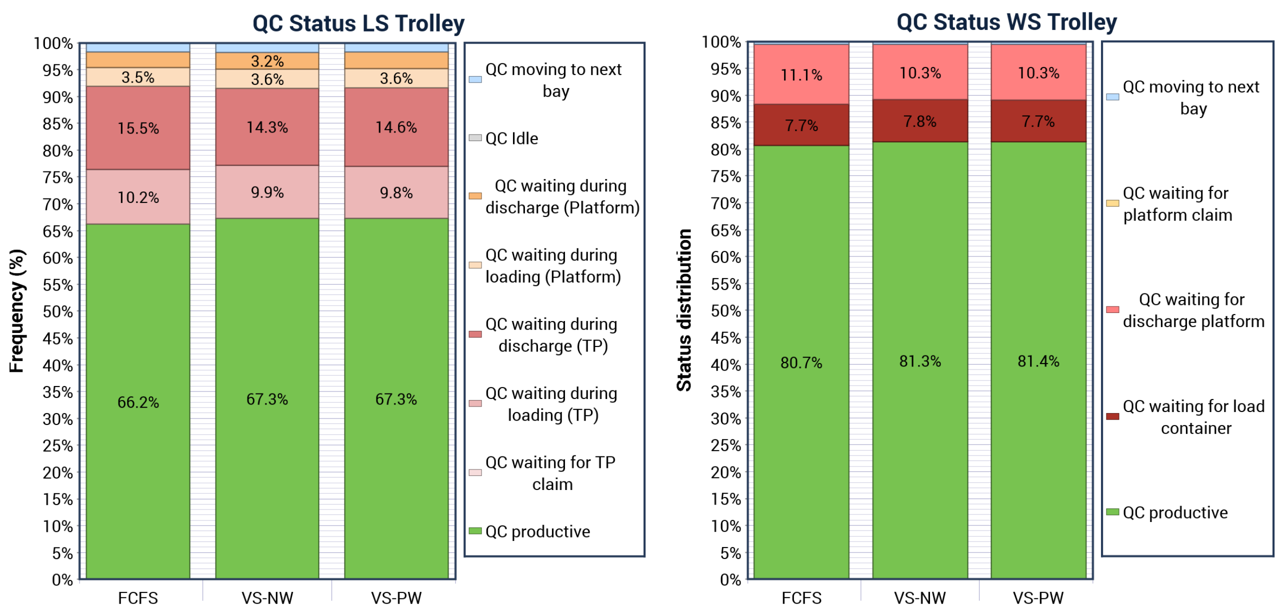
An improvement is also seen in the average lateness of AGVs (Table 9). It is shown that by using AGV scheduling for conflict resolution, all the tasks are completed earlier on average. Even without priority weights, the gain is the highest for the task with the highest priority. With the method that uses priority weights, an AGV that delivers containers to quay cranes arrives on average around 24 seconds earlier than in the benchmark scenario.

Table 9: Average lateness and the 95th percentile, given per AGV transport task type (10QC-40AGV)

	Task type			
Sub-scenario	Delivery - QC	Pickup - Stack	Pickup - QC	Delivery - Stack
FCFS	3.80 (8.83)	2.65 (6.88)	0.52 (3.38)	-1.30 (1.75)
VS-NW	3.56 (8.67)	2.57 (7.01)	0.36 (3.16)	-1.44 (1.53)
VS-PW	3.40 (8.39)	2.47 (6.70)	0.40 (3.15)	-1.35 (1.61)

This table shows for the experiments in main scenario 10QC-40AGV the average value and between parentheses the 95th percentile of the lateness of every completed AGV transport task over the whole experiment. The values are given in minutes.

The larger difference in the average lateness between the sub-scenarios is also reflected in the frequency that the landside trolley waits at the side of the transfer point (Figure 16a).



(a) (b)

Figure 16: Quay crane status distribution per trolley (10QC-40AGV)

Indeed, in a scenario with more congestion, a larger difference is seen between the used methods for conflict resolution. Waiting for a laden AGV at the transfer point during loading occurs less frequently with the method with priority weights. However, the method without priority weights gains more in terms of the frequency of waiting for an empty AGV at the transfer point during discharging. This may be explained by the fact that on average, an empty AGV arrives slightly earlier at the quay cranes with the method without weights.

The average net productivity of the quay cranes is shown in Table 10. Both sub-scenarios with the conflict resolution module seem to achieve a better performance than the benchmark, where only the performance in sub-scenario VS-NW shows a significant improvement. There is a difference in net productivity between sub-scenarios VS-NW and VS-PW, while the frequency of being productive is almost identical for the landside and waterside trolleys (Figure 16). This difference can be explained by the fact that there are relatively more twin moves used in sub-scenario VS-NW (Figure 20).

Table 10: Average quay crane net productivity (10QC-40AGV)

Sub-scenario	Average QC productivity (containers/hour)	Diff. with benchmark	95% Confidence interval (halfwidth)
FCFS	42.97	-	0.38
VS-NW	43.46	+0.50	0.36
VS-PW	43.32	+0.35	0.34

This table shows for the experiments in main scenario 10QC-40AGV (1) the average net productivity in terms of the number of containers moved per hour of all active quay cranes during the whole experiment, (2) the difference in average productivity with that of the benchmark sub-scenario, and (3) the half-width of the corresponding 95% confidence interval ($t^* \frac{s}{\sqrt{n}}$ in Equation (16)).

5.4.4 14QC - 50AGV

To evaluate the methods in an even larger environment, all fourteen quay cranes are set to be active and fifty AGVs are deployed. Compared to the previous scenario, there is again a relative increase in the time that AGVs spend waiting for claims (Figure 17), and a larger difference when comparing the values between themselves. Similar observations as in the previous scenario can be made regarding the driving behavior.

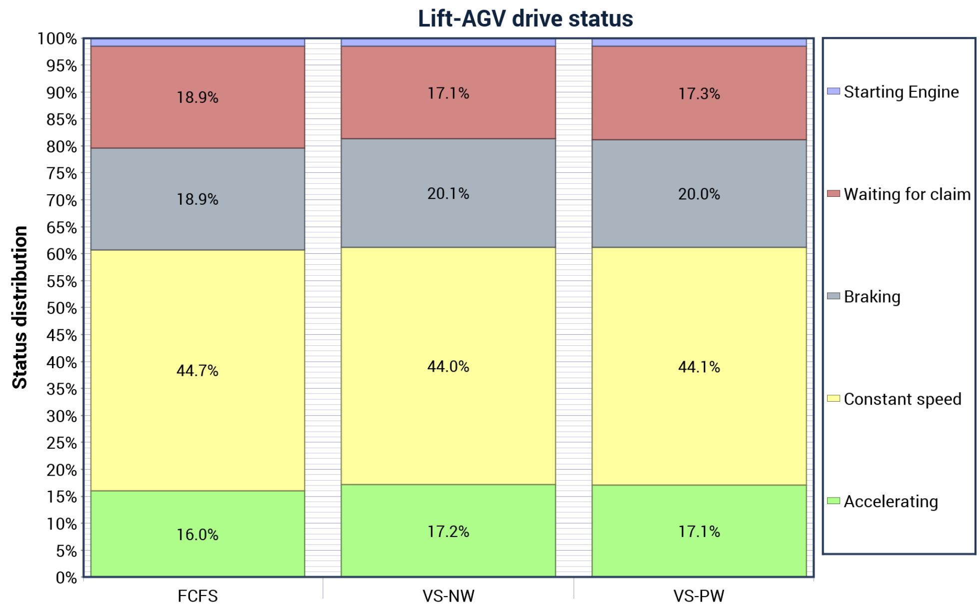


Figure 17: AGV driving status distribution (14QC-50AGV)

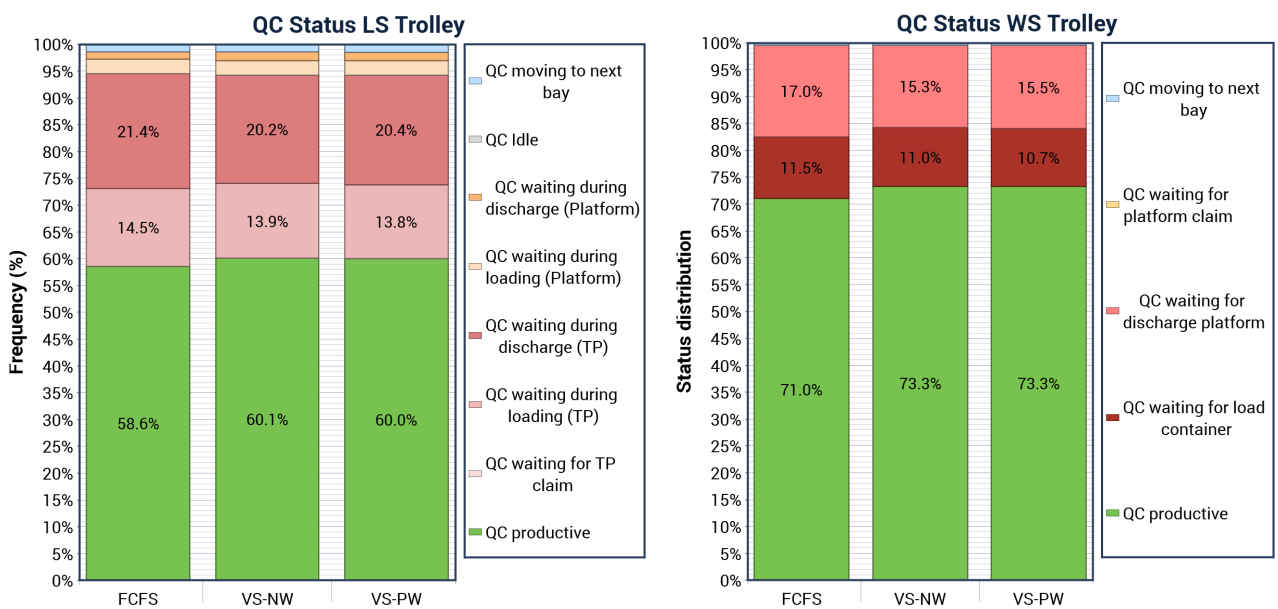
Regarding the average lateness of AGVs and the performance of the quay crane trolleys, similar observations are made as in the previous scenario. With a conflict resolution method that is not based on first-claim-first-served, it is shown in Table 11 that overall, the AGVs arrive earlier at the destination of their transport task. Interestingly, the average lateness of AGVs that deliver containers to a loading quay crane is *not* the lowest in the method that uses priority weights. Perhaps this scenario does not provide enough situations where AGVs with a higher priority are given precedence at conflict areas. It might be the case that the decision to let these AGVs go first does not lead to a lower total expected weighted delay that is used in the objective function in the AGV scheduling problem.

Table 11: Average lateness and the 95$^{\text{th}}$ percentile, given per AGV transport task type (14QC-50AGV)

	Task type			
Sub-scenario	Delivery - QC	Pickup - Stack	Pickup - QC	Delivery - Stack
FCFS	4.04 (9.81)	3.19 (8.38)	1.19 (4.23)	-0.66 (2.73)
VS-NW	3.70 (9.40)	3.07 (8.25)	1.06 (3.96)	-0.82 (2.38)
VS-PW	3.77 (9.56)	3.04 (8.13)	1.09 (3.94)	-0.75 (2.43)

This table shows for the experiments in main scenario 14QC-50AGV the average value and between parentheses the 95$^{\text{th}}$ percentile of the lateness of every completed AGV transport task over the whole experiment. The values are given in minutes.

With more quay cranes, but less AGVs per quay crane, Figure 18 shows that the waiting time of quay cranes has increased in this main scenario. This leaves more room for the introduced methods to show an improvement. The average productivity rate of both the landside (Figure 18a) and the waterside trolley (Figure 18b) has increased in comparison to the rate achieved in the benchmark scenario. However, there seems to be no clear difference in the output between the methods that use AGV scheduling at conflict areas.



(a) (b)

Figure 18: Quay crane status distribution per trolley (14QC-50AGV)

The average net productivity of the quay cranes (Table 12) also shows only a small difference between the introduced methods. Nonetheless, the difference that is shown in comparison to the value of the benchmark sub-scenario, is significant. With about 1.23 more containers moved in an hour per quay crane, around 17 more containers per hour can be moved as a collective by the quay cranes.

Table 12: Average quay crane net productivity for (14QC-50AGV)

Sub-scenario	Average QC productivity (containers/hour)	Diff. with benchmark	95% Confidence interval (halfwidth)
FCFS	37.03	-	0.34
VS-NW	38.23	+1.20	0.31
VS-PW	38.26	+1.23	0.28

This table shows for the experiments in main scenario 14QC-50AGV (1) the average net productivity in terms of the number of containers moved per hour of all active quay cranes during the whole experiment, (2) the difference in average productivity with that of the benchmark sub-scenario, and (3) the half-width of the corresponding 95% confidence interval ($t^* \frac{s}{\sqrt{n}}$ in Equation 16).

6 Conclusion

In this thesis, research has been conducted to investigate the relation between the lateness of AGVs in performing their transportation tasks and the productivity of quay cranes. In an attempt to reduce this lateness, a traffic control strategy has been proposed to resolve all conflicts between AGVs that occur in areas where AGVs cross each other. Whereas often in practice a first-come-first-serve rule is used, the proposed method overrides this rule and chooses an order in which AGVs must cross each other for every conflict. This choice is based on the minimization of the delay of all AGVs that is expected when the AGVs cross the conflicts in the given order. As an additional feature, the expected delays are weighted by a factor that is based on the type of task of the AGV, and the amount of slack the AGV has in regard to the due time of its task. The traffic control method is triggered whenever a new conflict is registered.

The proposed traffic control strategy has been evaluated using a discrete event simulation model, Timesquare, that had been provided by TBA Group. An environment has been chosen to mimic a typical LAGV-RMG container terminal. Various settings have been for the number of active quay cranes and the number of deployed AGVs to create situations where the bottleneck is shifted towards the AGVs and the landside of the quay cranes.

The first setting that was chosen showed that the quay cranes had a very high performance rate, with a low frequency of waiting for AGVs, even with the first-come-first-served rule,. The proposed strategy was not able to further improve the overall quay crane productivity. Moreover, the driving behavior of AGVs was slightly negatively impacted, as they were stopped more often.

The proposed traffic control strategy only showed an improvement in the results in settings in which there were relatively few AGVs per crane, such that they had a low performance due to a high frequency of waiting for AGVs. There seemed to be no significant difference when the estimated delays were weighted with a priority factor. Regarding the driving behavior of the AGVs, it seemed to be influenced in a way where they decelerate and accelerate more often, and halt to a complete stop less often. In these cases, AGVs were able to complete their transport task faster with the proposed traffic control strategy. With earlier completion times of AGV transport tasks, the quay cranes showed to wait less often for AGVs to deliver and to pick up containers. This lead to smoother operations between the landside trolley and the waterside trolley and a significantly higher average net productivity of the quay cranes. It should be noted that the improvement in performance was seen, despite the fact that the traffic control strategy did not operate flawlessly throughout the experiment.

To return to the research questions, this research confirms that the lateness of AGV transport orders can be reduced by using a traffic control strategy that is not based on a first-claim-first-served rule. Moreover, by reducing this lateness, this research shows that depending on level of congestion on the container terminal, the waiting time of quay cranes decreases, leading to a gain in in the quay crane productivity rate.

Discussion and Further Research Within the proposed method, several decisions were made for the setup of the AGV scheduling problem. Regarding the segmentation of the route, perhaps a different size could have been chosen to decrease the computational burden of detecting conflicts. In the scheduling problem, the estimations of the driving time between conflict areas and the crossing time of conflict areas were based on driving without interruptions. Ideally, the time for braking and acceleration should also be taken into account for AGVs that need to stop to let another AGV pass.

The decision was also made to resolve all conflicts whenever a new one is detected in a new route. When a solution is given, it will not change until a new conflict is detected. It is therefore more dependent on the accuracy of the driving time estimations that are used. Instead, a rolling horizon method could be used where conflicts are resolved whenever an AGV is expected to approach a conflict within a given time horizon.

Regarding the priority weights, it was shown that including them, in the way that they were computed, did not lead to a significant difference in the results. Due to a limited time availability and a long run time for the simulation experiments, it was not feasible to try different ways of computing the weights and to see the effects thereof. An opportunity for investigation may be to take the status of the quay cranes and the AGVs that queue up in the buffer zones into account.

As an extension to the proposed strategy, speed profiles could be assigned to AGVs when the solution for conflict resolution is known. For example, when an AGV is not the first to cross a conflict area, it could slow down in anticipation, instead of driving at maximum speed and braking fully. Then, this AGV does not have to accelerate from zero, which could potentially save energy.

When this traffic control strategy is adopted, it should be taken into account that there is a substantial amount of additional computation time to keep track of conflicts. However, resolving conflicts within this strategy proved to remain fast, where only about a tenth of a second was needed to do so in a system with up to fifty AGVs. Furthermore, it should be noted that the strategy may malfunction due to inconsistencies in merging conflicts. The errors may be specific to the way conflicts are represented in the simulation model, where it showed to be too time-consuming to set up rules that cover all special cases that occur for conflicting AGVs. Nevertheless, a different approach could be considered in the future for the registration and merging of conflicts.

It should also be mentioned that AGVs in this strategy could drive in a way with more braking and accelerating, which is less efficient regarding energy consumption. There is therefore a possible trade-off to be considered between a higher productivity rate, and a higher computational burden and less efficient driving of AGVs.

References

Bae, H. Y., Choe, R., Park, T., and Ryu, K. R. (2008). Travel time estimation and deadlock-free routing of an agv system. In *Dynamics in logistics*, pages 77–84. Springer.

Bae, H. Y., Choe, R., Park, T., and Ryu, K. R. (2011). Comparison of operations of agvs and alvs in an automated container terminal. *Journal of Intelligent Manufacturing*, 22(3):413–426.

Cardarelli, E., Digani, V., Sabattini, L., Secchi, C., and Fantuzzi, C. (2017). Cooperative cloud robotics architecture for the coordination of multi-agv systems in industrial warehouses. *Mechatronics*, 45:1–13.

De Ryck, M., Versteyhe, M., and Debrouwere, F. (2020). Automated guided vehicle systems, state-of-the-art control algorithms and techniques. *Journal of Manufacturing Systems*, 54:152–173.

Digani, V., Hsieh, M. A., Sabattini, L., and Secchi, C. (2019). Coordination of multiple agvs: a quadratic optimization method. *Autonomous Robots*, 43(3):539–555.

Digani, V., Sabattini, L., Secchi, C., and Fantuzzi, C. (2014). Hierarchical traffic control for partially decentralized coordination of multi agv systems in industrial environments. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6144–6149. IEEE.

Evers, J. J. and Koppers, S. A. (1996). Automated guided vehicle traffic control at a container terminal. *Transportation Research Part A: Policy and Practice*, 30(1):21–34.

Ferrari, C., Pagello, E., Ota, J., and Arai, T. (1998). Multirobot motion coordination in space and time. *Robotics and autonomous systems*, 25(3-4):219–229.

IBM (2020). IBM ILOG CPLEX Optimizer 20.1.0.0. `https://www.ibm.com/analytics/cplex-optimizer`.

Makarem, L. and Gillet, D. (2012). Fluent coordination of autonomous vehicles at intersections. In *2012 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 2557–2562. IEEE.

Van Thielen, S., Corman, F., and Vansteenwegen, P. (2019). Towards a conflict prevention strategy applicable for real-time railway traffic management. *Journal of Rail Transport Planning & Management*, 11:100139.

Van Vuuren, B. (2017). Route plan scheduling for automated guided vehicles at container terminals. Master's thesis, TU Delft.

Zhong, M., Yang, Y., Sun, S., Zhou, Y., Postolache, O., and Ge, Y.-E. (2020). Priority-based speed control strategy for automated guided vehicle path planning in automated container terminals. *Transactions of the Institute of Measurement and Control*, 42(16):3079–3090.

A Appendix

A.1 Results

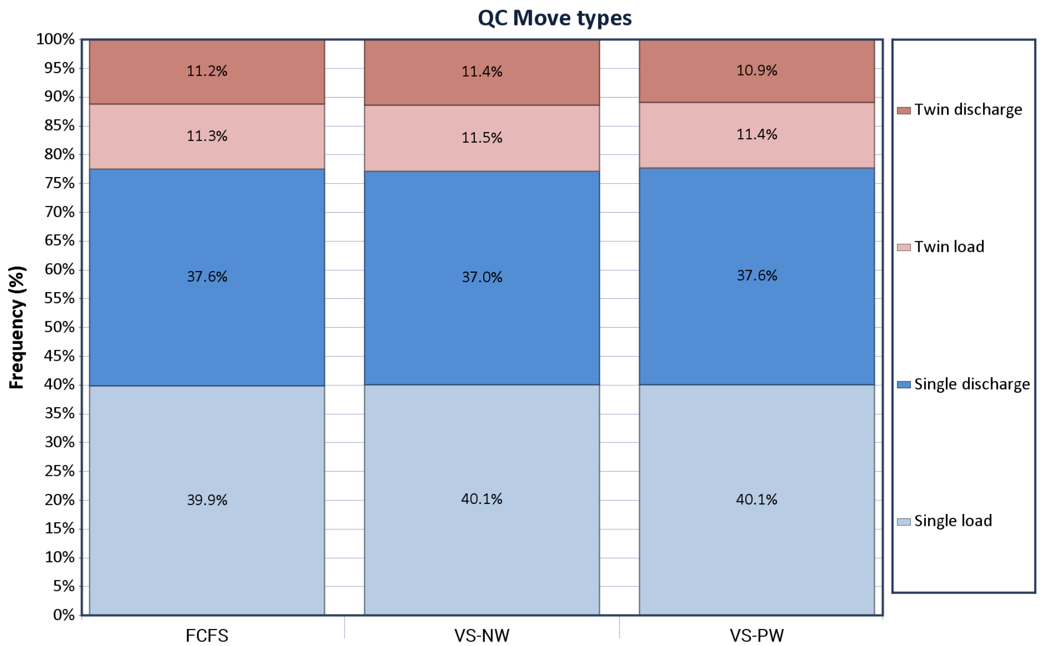


Figure 19: The distribution of move types of QCs across all replications in main scenario 4QC-20AGV

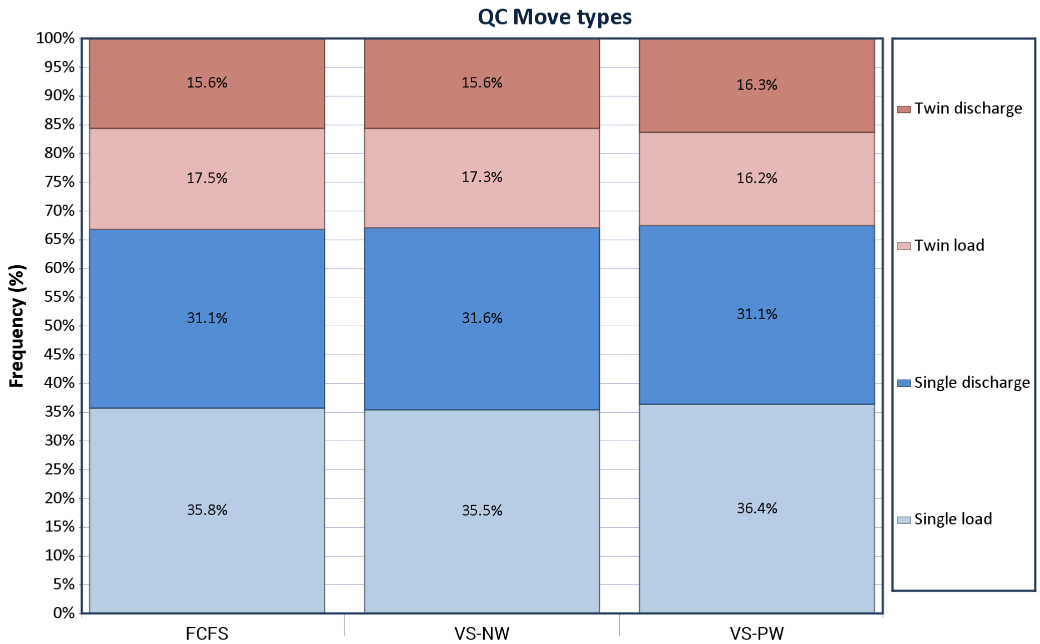


Figure 20: The distribution of move types of QCs across all replications in main scenario 10QC-40AGV

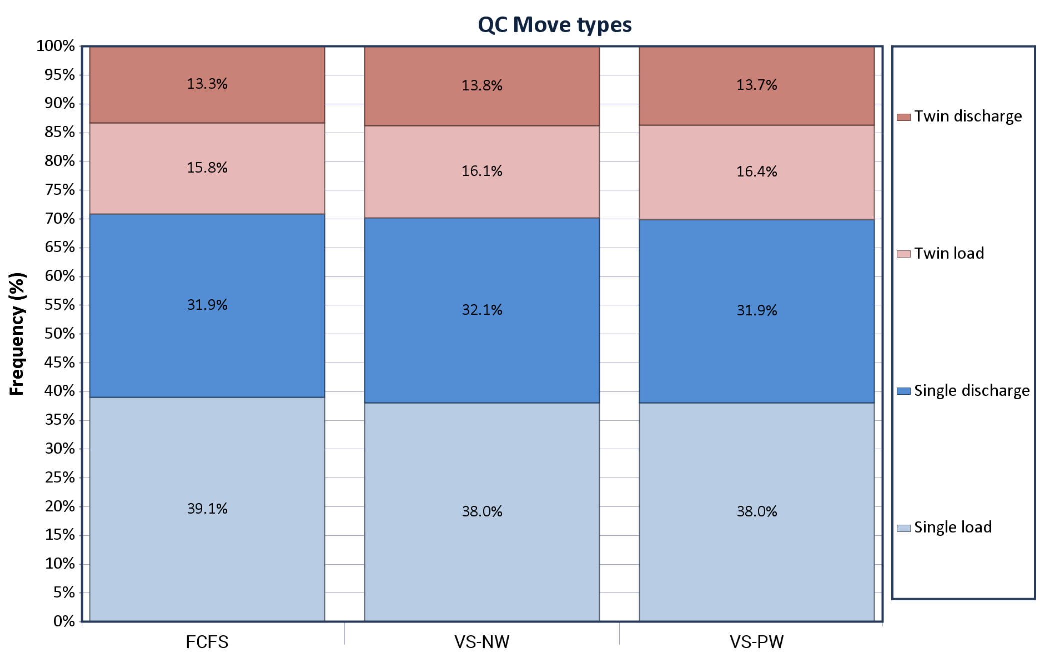
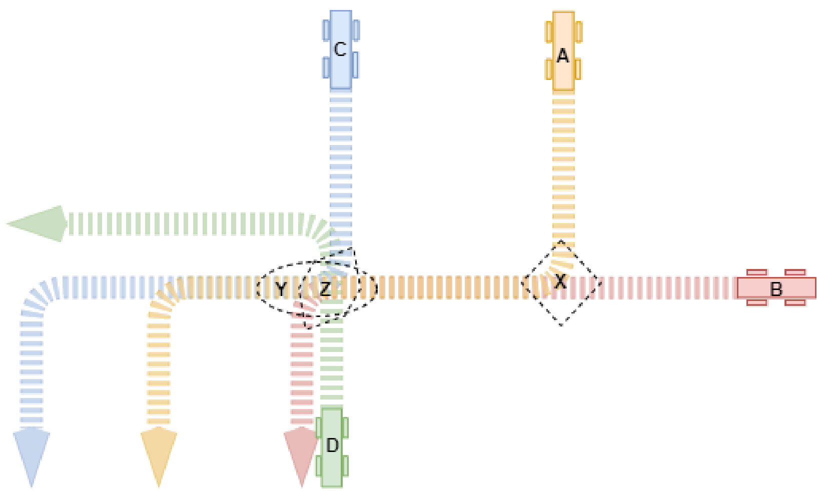


Figure 21: The distribution of move types of QCs across all replications in main scenario 14QC-50AGV

Inconsistencies in Instances

The way an optimization model is built is dependent on the correctness of the given instance and therefore on a correct conflict registration.

Especially when conflicts areas were clustered, it sometimes occurred that the conflict merging module was not able to merge conflicts correctly. It could therefore occur that, for example as in Figure 22, AGV A must cross conflicts X, Y, and Z in this order, while AGV B is registered only in X and Z, and while AGV B joins AGV A at X. Constructing the optimization model would therefore fail in this example when Constraint (8) is added, as the variable $b_{A,B,Z'} = b_{A,B,Y}$ is not defined.



(a) Conflicts that are present

Conflict	Involved Vehicles
X	A, B
Y	A, C
Z	A, B, C, D

(b) Registered conflicts

Figure 22: Example of an inconsistent instance