Route plan scheduling for automated guided vehicles at container terminals

B. van Vuuren

TU Delft

Route plan scheduling for automated guided vehicles at container terminals

by

B. van Vuuren

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday September 28, 2017 at 14:00.

Student number:	1364693	
Project duration:	February 6, 2017 – September 28, 2017	
Thesis committee:	Dr. M.T.J. Spaan,	TU Delft
	Dr. M. M. de Weerdt,	TU Delft
	Dr. F. A. Kuipers,	TU Delft
	Ir. G. Bast,	TBA Group, Delft

This thesis is confidential and cannot be made public until September 28, 2018.

Preface

After receiving 24 years of education, the time has come to finish this chapter of my life by completing this master's thesis. It is needless to say that I would never have reached this point without the help of many others. Firstly, I would like to thank my parents who made it possible for me to receive education, and for who commitment was more important than results. Furthermore I would like to thank my roommates that made my time as a student unforgettable.
Concerning my thesis project, I would like to thank TBA for providing an interesting assignment and an environment where fun and hard work were both highly appreciated. The colleagues of the Simulations group were always ready to help me when I got stuck or wanted to discuss some new ideas. Especially my supervisors Arjen and Gijsbert, who spend a lot of time on helping me to successfully finish this project. I am very happy that Matthijs was willing to be my daily supervisor, as he proved to be an enthusiastic supervisor that showed great involvement in this project.
However, my greatest gratitude goes to my wife Karin, who encouraged me to start this master course, and supported me from begin to end. I would not have made it without you!

B. van Vuuren
Delft, September 2017

Abstract

Heavy competition between container terminals and the increasing volume of world-wide container shipment force container terminals to continuously increase productivity and reduce costs. An important condition for optimal operation of container terminals is that Quay Cranes (QCs) maximize their productive time, which requires a minimal lateness of Automated Guided Vehicles (AGVs) that bring and pick up containers at the QCs. In this research, the AGV driving strategy is investigated in order to reduce the time QCs are waiting for an AGV.

A route plan scheduling module for AGVs is developed, in which the central system plans the driving positions of AGVs in advance, and determines in what order AGVs should cross intersections. For priority determination, a first-come first-served strategy is compared with an advanced priority strategy that gives priority to AGVs that are late or have an important job, like bringing a container to a QC. Furthermore, the impact of AGVs' greediness for crossing intersections is investigated by comparing a greedy strategy with a non-greedy strategy.

By testing the priority and greediness strategies in a state-of-the-art simulation environment that was extended with our route plan scheduling module, it is found that advanced priorities reduce QC waiting time and improve QC productivity compared to the first-come first-served priority strategy. The non-greedy strategy gives a small decrease of the AGV driving delay compared to the greedy strategy. To investigate the full potential of using advanced priorities in a system with route plan scheduling, the route plans should better predict the AGV driving behavior, for which we suggest to use a more detailed AGV planning strategy. Finally, we discuss that AGV driving with reduced inter-vehicle distance is a promising approach to improve terminal performance.

Contents

1

Introduction

This chapter starts with an introduction to automated container terminals in section 1.1, followed by a motivation for the use of simulations in section 1.2. Subsequently, section 1.3 describes the objective of this thesis, which leads to the research questions in section 1.4. The contribution of this research to the field of container terminal optimization, is stated in section 1.5, and finally the outline of this thesis is provided in section 1.6.

1.1. Automated container terminals

In the highly competitive world of container transportation, the sizes of the largest vessels are continually increasing, to handle the increasing amount of container moves and reduce the costs. The size of container vessels is expressed in the number of standard sized containers, Twenty-feet Equivalent Units (TEU), they can transport. Although TEU is the default unit for vessel sizes, most vessels and terminal equipment are set to handle forty-feet containers, which have the size of exactly two TEUs in a row. Twenty-feet containers are still used intensively, but are mostly handled in pairs by the same equipment that handles the forty-feet containers. Nowadays, the largest vessels have capacities over 21.000 Twenty feet Equivalent Units [2] and on major terminals, these vessels generate up to 10.000 container moves per vessel call, but even these vessels should be handled in only 2 to 3 days [9].

1.1.1. Quay Cranes

As schematically shown in figure 1.1, vessel discharging is performed by giant Quay Cranes (QC) that grab a container from the vessel and drop it at a transfer point at the back of the QC. From this transfer point, the container is moved to a transfer point at the stacking area by some type of horizontal transportation vehicle. Different types of equipment can be used for this transport, however, a single terminal usually has only one type of equipment to perform these horizontal moves. For fully automated container terminals, which we investigate in this research, a commonly used type of vehicle is the Automated Guided Vehicle (AGV), which drives without any human intervention.

The speed of vessel's loading and discharging, heavily depends on the QCs' productivity. Moreover, QCs are generally the most costly part of equipment on the terminal, and their idle times should therefore be minimized [18]. A good performance of the QCs is vital for a terminal to increase the throughput and decrease the costs, causing terminals to invest millions of euros to improve this performance.

A well functioning AGV system that minimizes the QCs' waiting times by providing loaded or unloaded AGVs in time, is necessary to achieve the high performance goals of QCs. Ideally, there is an AGV at the transfer point, whenever the QC wants to pick up or drop a container. It is a far from trivial task to achieve this, as the containers need to be loaded in a predefined sequence, the area where AGVs drive, can be crowded and the number of AGVs is limited. In real operations, AGVs are regularly hindered by other AGVs, which sometimes leads to QC idle time. If we could decrease the delays of AGVs, especially of AGVs that are already late, the QCs productivity could increase, which means a terminal productivity increase.

1.1.2. Automated Guided Vehicles

In the remainder of this thesis, a special type of AGVs will be assumed, namely Lift-AGVs, which can autonomously pick-up or drop a container at a rack before the stacks. At the QC transfer points however, Lift-

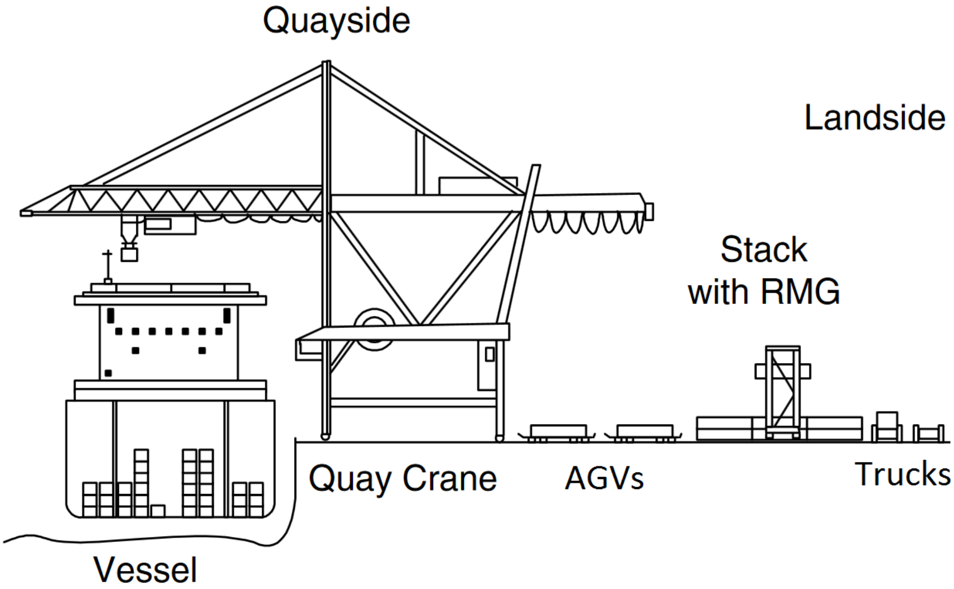


Figure 1.1: Schematic side view of an Automated Container Terminal, based on an image of Steenken et al [16].

AGVs, need to be loaded and unloaded by the QC (figure 1.2). For these AGVs, the properties are not directly taken from one type of real-world AGVs, but they are all realistic and based on average properties of real AGVs. The AGVs in this research have a length of 15 meters, a width of 3 meters and the load platform has a height between 2.4 and 2.7 meters. The AGV can handle a single forty-feet container or two twenty-feet containers, and in the remainder of this thesis, the performance will be expressed in moves, so the difference between moving a single forty-feet container and moving two 20-feet containers, is ignored. Furthermore, the AGVs are bidirectinal and can drive at most $6m/s$ on straight roads, or $3m/s$ in curves.

Positioning is done by communication with transponders that are placed in the roads, and is accurate to 3 cm. The driving is coordinated by the central system, which decides where the vehicles should drive, and that constantly sends new destination positions to the AGVs. The vehicles' autonomy is limited to deciding what speed can be driven in order to be able to halt at the last destination it received from the central system. The central system uses a conservative claiming strategy to guarantee collision free driving of the AGVs, in which vehicles receive exclusive access to certain parts of the terminal. The driving destinations that are sent by the central system to the AGVs, are always inside these exclusively granted areas, which ensures that AGVs do never collide. In practice, this results in an inter-vehicle distance which is at least as large the braking distance of the following vehicle, which is approximately 30 meters for AGVs that are driving on top speed.

The claiming strategy leads in practice to a First Come, First Served priority strategy for areas that are part of multiple AGV routes as the central system grants access to a certain area if it is not currently occupied by another vehicle. Granted requests are never revoked by the central system, which means that the AGVs' driving strategy can be classified as a greedy strategy because vehicles are minimizing their own driving time without taking other vehicles into account.

1.1.3. Stacking area

Containers that arrive by ship, are stored in the yard, which consists of a number of container stack modules. Most real world terminals are rectangular and have the stacking modules positioned parallel or perpendicular to the quay, as schematically displayed in figure 1.3. To limit the scope of our research, only one type of terminal layout is investigated. The layout with a perpendicular stack is chosen because it covers more of the possible aspects of real world terminals with its land side transfer points at the stacks where trucks can bring or receive containers for transportation from and to the hinterland, it would however be interesting to also investigate the performance on a layout with parallel layout, to see how the simulated strategies work out on a parallel layout. On every stack module, several yard cranes handle the container moves related to that stack, namely serving the trucks at the land side transfer points, handling the racks at the sea side transfer points and performing internal container movements to support the handling of the transfer points.

The most commonly used types of yard cranes are Rubber Tired Gantry cranes and Rail Mounted Gantry

Figure 1.2: Lift-AGVs at Quay Crane transfer points (left) and at stack transfer points (right) [1].

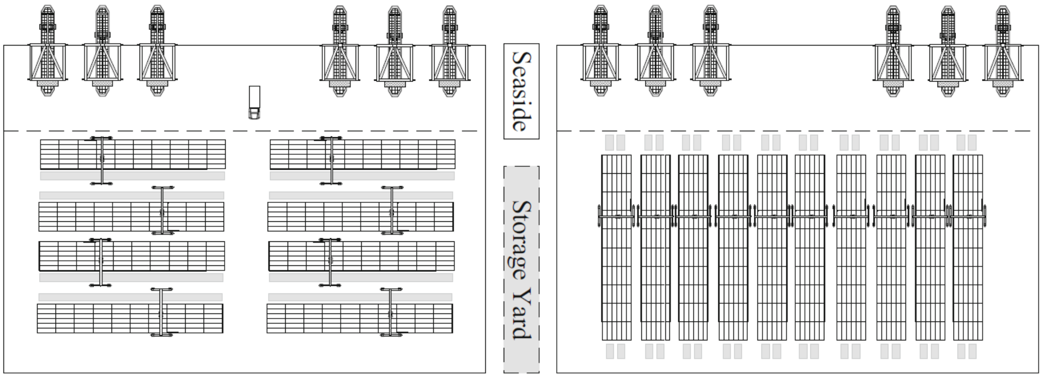


Figure 1.3: Parallel (left) and perpendicular (right) layout of container yard [20].

cranes, with as most important operational difference that Rail Mounted Gantry cranes have a fixed stack module to serve, while Rubber Tired Gantry cranes can change between different stack modules. For this thesis, the type of stack crane is not very relevant, and hence, we choose the Rail Mounted Gantry as this is less flexible which makes its behavior easier to compute and analyze.

1.2. Simulations

For new terminals, usually multiple layout and equipment scenarios are considered. To create substantiated predictions of future terminal productivity in different work load scenarios, independent experts are consulted to make productivity estimates using complex terminal simulations. The simulation outcomes are used as important guidelines for the several hundreds of millions of euros that are invested in the terminal layout and the equipment. Due to the great financial impact of the simulation results, the simulation consultancy has become a professional industry with very detailed simulations that provide generally quite reliable results.

For this research, the state of the art TIMESquare simulation model, which is developed by the TBA group, a major consultancy company in the field of container terminal optimization. The model has proven to be an appropriate tool to evaluate the performance impact of newly developed strategies, and is therefore a suitable tool for this research. More details about the TIMESquare model will be provided in section 2.1.

1.3. Objective

The goal of this research is to contribute to the continuous development of equipment strategies that improve terminal performance. An important Key Performance Indicator of automated container terminals, is the QC productivity, and therefore, our objective is to increase this productivity. To obtain a higher productivity, we minimize the time that QCs are waiting for AGVs, by improving the AGV driving flow.

The scope of this thesis is limited to AGVs' behavior, so the operation of other terminal equipment is only used as input for our model. The intelligence of our solution is implemented in the central AGV manager that continuously updates to what point an AGV should drive. This module gets the routes of the AGVs as input and its sole task is to schedule the drives along the received routes in such a way that the total AGV flow is optimized.

The AGVs themselves have limited computational power and only calculate at what speed they should drive to be able to halt at the last route point that was received from the central AGV manager. The desired solutions should work for a fixed layout and be suitable for real world terminals, meaning that the system should be scalable up to 100 AGVs, guaranteed collision-free and in practice deadlock-free.

1.4. Research questions

The goal of this thesis is to minimize the time QCs are waiting for AGVs on Automated Container Terminals. This brings us to the following research question:

Main research question. *To what extent can the QCs' waiting times be decreased by improving the AGVs' driving behavior?*

To answer this research question, we should first analyze how much time QCs are waiting for for an AGV, and whether this is different for loading and discharging QCs. To put these values in more context, it is also valuable to know how much time AGVs are waiting, leading us to our first sub question:

Sub research question 1. *How much time are AGVs waiting during their drive and how much time are QCs waiting for an AGV?*

With this sub question answered, we can compare new strategies to the original situation. We know that the use of AGVs' priorities has the potential to increase the QC's productivity, and it is therefore worth investigating to what extent priorities can decrease the QC waiting times in a model with AGV drive scheduling. This leads to the following sub question:

Sub research question 2. *To what extent can the QCs' waiting times be decreased by using AGV priorities?*

Priority determination can be applied in different ways to drive AGVs, ranging from greedy to defensive. The choice of greediness can impact the performance of AGVs due to the interaction with routes that change or routes that are added at later times. This brings us to our third sub research question:

Sub research question 3. *What is the effect of AGVs' greediness on the QCs' waiting times?*

By answering these 3 sub research questions, we can formulate a balanced answer to our main research question.

1.5. Contributions

The research in this thesis aims to contribute to the field of research in container terminal automation, but has also added value for other situations with autonomous driving vehicles in controlled environments like warehouses and factories.

A first contribution is that we applied is to apply route plan scheduling for autonomous vehicles in a realistic industrial environment while respecting the real world constraints. In other research the scenarios are often very small and the safety is only guaranteed if vehicles act exactly as expected. In this research however, a scenario with 40 AGVs is performed in which real-world safety guarantees are provided.

Furthermore, our system is highly resilient against deviations from the AGV driving plans, which is also vital for real world systems, and not investigated before. Next, the prior determination of AGV driving order based on an advanced priority determination strategy, is a contribution to the research field. Finally, within the developed route plan scheduling strategy, the AGV greediness is investigated.

1.6. Outline of this thesis

In the next chapters, we first give a more detailed and formal description of our problem statement in chapter 2, followed by an overview of related work of others that was already done by others in chapter 3. Subsequently, we describe the proposed methods to answer these research questions in chapter 4, which lead to the results and corresponding analysis in chapter 5. Finally, in chapter 6, we formulate answers to our research questions and state our suggestions for application of this work and further research that could be done based on this thesis.

2

Problem Description

Before formulating the research question in a formal way, we first give an overview of the TIMESquare model that is used for the simulations. Next, we describe the limitations of the current system in the constraints section 2.2, followed by the requirements on the used solutions in section 2.3. Finally, we define the problem formally and also formulate the main research question as a formal minimization problem in section 2.4.

2.1. TIMESquare simulation model

The scope of our research is limited to the optimization of Automated Guided Vehicle (AGV) behavior in a simulation environment, but by using the professional simulation model TIMESquare, that is also used in industry to perform simulations for fully automated container terminals, the simulation results can be considered as a good prediction of results in real terminals. To the best knowledge of the author, there are currently no two terminals in the world that have the exact same layout but a lot of terminals however, do have similar structures, and the model that is used, combines properties of several real-world terminals and can be seen as a realistic terminal. An overview of the terminal used in this study, is shown in figure 2.1.
Along the quay, there are eight Quay Cranes (QCs) located. These cranes can perform at most 40 moves per hour, under perfect circumstances, i.e. AGVs that serve the cranes are always in time to bring or receive a container. In practice however, this performance is often not achieved.

2.1.1. Terminal layout

In the used layout, there is no train connection to the terminal, and the only inland transportation goes by trucks, which can bring or receive a container at the land side transfer points of the stacks. The whole terminal handles 100 gate moves per hour. These gate moves itself are not relevant for the research, however, if we

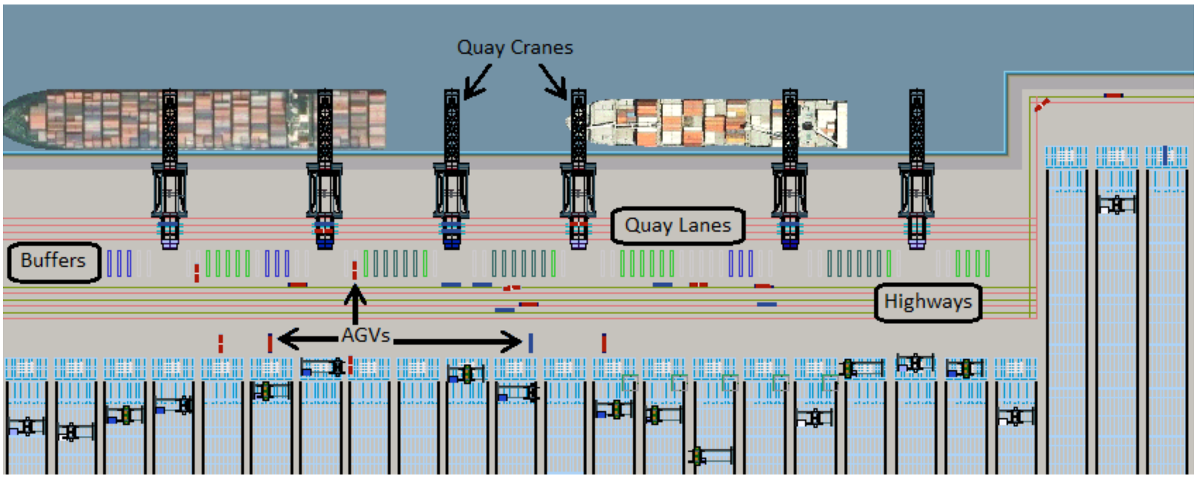
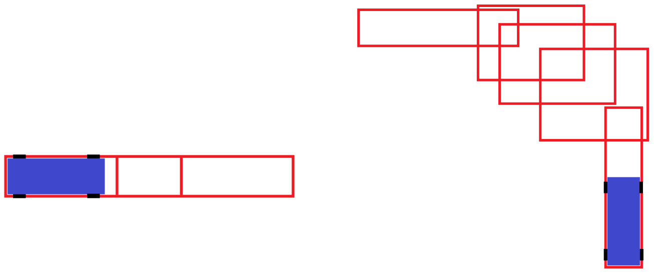


Figure 2.1: Screenshot of a part of the used TIMESquare simulation model



(a) Two overlapping straight AGV claims

(b) A curve claim consists of multiple rectangular areas

Figure 2.2: Schematic view of straight claims and a curve claim.

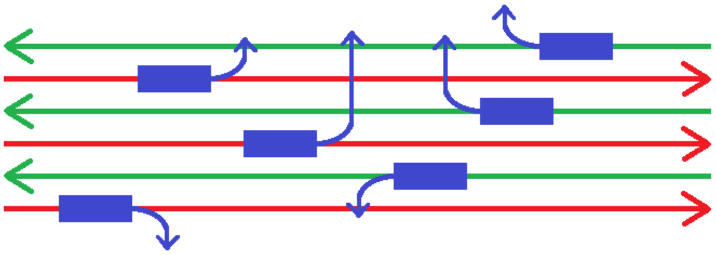


Figure 2.3: schematic view of driving directions of AGVs, the top two lanes are used for drives from a certain place in the Quay Crane area to another position in the Quay Crane area. The middle two lanes are used for drives from the Stack area to the Quay Crane area, and the bottom two lanes are used for all drives to the Stack area.

would remove the gate moves from the model, the two yard cranes that are situated in every stack module, would cooperate to fulfill all the AGV requests faster than in reality. This removes a lot of delay that does occur in real terminals, making it harder to see how our solution handles AGVs that are delayed.

The chosen layout contains 28 stack modules that are perpendicular to the quay. In every module, 2 yard cranes handle both the sea and land side requests. On the layout, there are a number of fixed, unidirectional lanes defined, which are drawn as green and red lines on the map in figure 2.1. There is a highway parallel to the quay that consists of six lanes, which is used for transport between the stacking area and the QCs. When an AGV drives from the stack to a QC, it first drives to a buffer close to the its destination QC. Every QC has two transfer points which lay on the Quay lanes. When there is a free transfer point under a QC, the next AGV in order gets permission to drive to this transfer point. When an AGV has delivered or received its container at the transfer point, it directly drives to an empty buffer while waiting for a next destination.

We choose to use a layout of average complexity. It would be preferable to investigate a simple, medium and complex model to see if and how the solutions perform for different complexities, but due to the limited duration of this research, we choose to only investigate one terminal of intermediate complexity. The aim of this study is to explore the profitability of using sequential route scheduling in a general case, and that can be done fairly well with a single layout.

The used AGVs have lifting capacity and can grab or drop a container at a rack on the sea side of the stack without help of the Stacking Crane. At the transfer point with the QCs however, the AGV should be loaded or unloaded by the QC as there are no racks situated there.

The driving of AGVs is coordinated by the central system that is responsible for the collision- and deadlock prevention of AGVs. The driving cycle of an AGV starts with the receipt of a route from the central system. After an AGV has received a route, it starts asking the central system for exclusive access to the first part of its route. Based on the access to parts of the track that were already granted to other AGVs, the central system decides whether the AGV will receive its requested exclusive access. As long as the AGV's request is not granted, the AGV waits for a message of the central system has been left by the AGV that occupied it.

When an AGV's request is granted, it starts to drive to the end of the granted area. Meanwhile, it requests access to its subsequent route segment, and when this access is also granted, the AGV updates its endpoint. To prevent unnecessary accelerating and decelerating, AGVs send their requests early enough to be able to continue driving on their maximum speed if the traffic situation permits.

2.1.2. Driving of Automated Guided Vehicles

The AGVs area requests in TIMESquare are called claims and they appear in two forms, rectangular claims for straight parts of the route (figure 2.2a) and curve claims for curves in the AGV route (figure 2.2b). The rectangular claims are wide enough to cover the AGV and lengthwise they overlap with previous and following claims. A curve claim is a set of overlapping rectangular claims that together fully cover the area that is used by the AGV to drive the curve. The separte rectangular claims of a curve claims however, cannot individually be granted by the central system.

To prevent AGV deadlocks, the tracks on the terminal have a fixed driving direction and the destination of an AGV determines what lane a vehicle takes. As shown in figure 2.3, the terminal that is used for this study, has six high way lanes of which four are used by vehicles that move towards the Quay Cranes and two for AGVs that drive towards the stack. This strategy prevents most of the deadlocks on the high way as two vehicles that both drive on the highway, cannot mutually block each other.

Additionally, the claiming area of a vehicle around a curve, is marked as a no-stop claim. This claim consists of the curve claim together with a short straight claim before and after the curve. The AGV requests access to this (large) no-stop claim at once, to make sure it will never have to stand still while it is driving a curve. Also for high way inserting of vehicles, they use a no-stop claim that ensures that an inserting only gets permission to insert when it can directly drive until it is fully on its desired lane. In practice, these rules are enough to prevent deadlocks on the highways.

2.2. Constraints

The hardware of the assumed equipment, the configuration of the Terminal Operating System and the limitations of the TIMESquare model, add constraints to our research. Furthermore, the solution space is reduced even further, by adding constraints on the proposed solutions. The mentioned constraint types can be classified as the following constraint types: physical constraints, configuration constraints, model constraints and solution constraints.

Physical constraints include all constraints that cannot be bypassed with changes to the Terminal Operating System. Most of these constraints are related to AGVs, which are homogeneous and bidirectional, have no hardware to directly communicate with each other, can pick up or drop a container autonomously at the stack transfer point, but cannot drop or pick up a container at QC transfer points. Furthermore, their maximum speed is 6 meters per second and they are capable of driving to a certain position on the terminal while adjusting their real position using the transponders that are placed on the area where the AGVs drive. Other physical constraints are the terminal dimensions and the stochastic distribution of job execution times of QCs and yard cranes.

A next set of constraints are the configuration constraints, that are caused by the limitations of the used Terminal Operating System. The routes AGVs can take, are limited to fixed, virtual, unidirectional lanes. The route an AGV has to drive, is only known shortly before its expected start time and the AGV itself only gets next coordinates it should drive to. The last configuration constraints are that AGVs should drive at most 3 meters per second in curves and at most 1 meter per second when crabbing.

Subsequently, the model constraints express what limitations are introduced by using the TIMESquare simulation model. For this study, the model constraints are that time is discretized in heterogeneous slots of at most 1000 ms and that AGV breakdowns are not taken into account.

Finally, we introduce some solution constraints to limit the solution space. The focus of this thesis is on the driving of AGVs and therefore, the behavior of other equipment should not be taken into account and is considered as an input for the model. Furthermore, the allocation of jobs to AGVs, i.e.: drive from A to B, is a whole new research area, which we do not address in this thesis. Also, containers generally have doors at only one side, making them unidirectional, and the direction of the doors is stored in the system. On vessels and trucks, the door should always be at the rear side, and as there is no crane on the terminal that can rotate containers, the correct orientation of containers should be preserved by the AGVs, forcing them to make a special move to turn the AGV sometimes.

2.3. Requirements

The aim of this study is to develop strategies for AGV behavior that could be implemented in real terminals and therefore we state a number of requirements on the solutions to guarantee that the solutions do not only have a theoretical value. This section only describes the requirements on our additions to the system. First of all, the researched solutions should be fully collision free, furthermore, deadlocks should in practice

(a) The routes of AGV A and AGV B overlap in the segment between X and Y.

(b) The routes of AGV A and AGV B intersect in X and in Y

Figure 2.4: Schematic overview of AGVs with overlapping routes and routes that cross twice.

only occur very rarely, at most once a week. Another important requirement for real-world value is that the solution should be scalable up to 100 AGVs, which we assume to be achieved when a single computer can run the simulation in real time. Finally, the system should remain collision free in case of message delay or loss between AGVs and the central system, unexpected AGV braking and speed deviations of the AGVs of at most 20 percent.

2.4. Problem definitions

During a simulation run, the Terminal Operating System is responsible for the supply of containers and empty AGVs to the QCs. Therefore, it sends jobs to the AGVs that consist of a driving route, a task (e.g. Drive to stack X and grab container Y) and a due time. This due time is especially important for AGVs that are driving towards a QC and for AGVs that are driving empty to the stack to grab a container, as they are indirectly also driving towards a QC. Next, the received route is split up into segments that are stored in the claim list.

The claims in this list are either straight claims or curve claims which are schematically drawn in figure 2.2. A straight claim covers a single rectangular area and a curve claim is the union of multiple rectangular areas. The claims in the claims schedule together cover the full route of the AGV during its drive and consecutive claims in the schedule have an overlap that is large enough to cover an AGV. As a consequence, an AGV is always fully covered by at least one of the segments in the claim schedule. For all claims of an AGV, a list of overlapping claims of other vehicles is maintained, which describes the possible conflicts between this AGV and other AGVs. For all overlapping pairs of claims, a desired order of drive should be computed, which is handled by the scheduling module that is developed by the author.

Priority decisions of a pair of claims can influence priority decisions for other pairs of claims as displayed in figure 2.4. If, for example, two AGVs drive partly the same route (see figure 2.4a), then the vehicle that is allowed to cross point X first, should maintain a higher priority than the other AGV until point Y, as vehicles cannot overtake each other. However, the straightforward solution to calculate the order of priority only once for every pair of vehicles, leads to large delays in certain situations as shown in figure 2.4b. Giving priority to one AGV for both intersections X and Y, leads to unnecessary waiting for the other AGV, as the optimal strategy here would be to give priority to AGV A at intersection X and to give AGV B priority at intersection Y. Therefore, a more sophisticated strategy is applied that can distinguish situations like in figure 2.4a and in figure 2.4b.

Based on the priority decisions, the list of claims that do overlap with a certain claim, is divided into a list of claims that will be granted before this one and a list of claims that will be granted after this claim. A vehicle's request for a claim is only granted if all segments from the before-list are released. If the access to an AGV's claim request is granted, then the AGV has exclusive access to that area with immediate effect until the AGV releases it.

The central system should also prevent deadlock situations, which we informally define as a situation in which a group of AGVs is mutually waiting for each other. Conditions for a deadlock are that all vehicles want to drive, but none of them is allowed to drive. For the simulations in this thesis, the system is considered deadlock-free if deadlocks occur less than once per 100 hours of simulation. This is also safe enough for a real world terminal as there is always a terminal operator available that can manually solve these rare deadlocks.

If N is the number QCs in production for a simulation of H hours, the research problem can be stated as the following minimization problem:

$$\min \sum_{i=1}^{N} \sum_{j=1}^{H} t_{QC_i,\text{wfa},j}$$

in which $t_{QC_i,\text{wfa},j}$ is the time fraction QC i is waiting for an AGV during hour j.
This minimization is subject to the following constraints:

- During the execution of the simulation, no collisions should occur.
- Deadlocks should occur at most once per 100 hours of simulation.

3

Related work

In the literature, several levels of Automated Guided Vehicle (AGV) driving are distinguished([5], [8]). There exists no widely used standard for this taxonomy, and there are several definitions for the terms 'routing' and 'scheduling'. In this paper, we use a taxonomy, based on the work of Co [4]. Co distinguishes three primary functions of the AGV System. The first function is dispatching, which consists of the assignment from jobs to AGVs (or sometimes assigning AGVs to jobs [17]). Routing consists of finding a route to execute that job. The last function is scheduling, in which a schedule is built to drive collision-free along that route. How this schedule is optimized, depends on the goals and strategy of the routing system.

3.1. Vehicle routing

Fazlollahtabar [8] divides the path finding part of routing into two types, static and dynamic. Static algorithms determine a whole route in advance without taking dynamically changing traffic density into account. Static algorithms are closely related to the Vehicle Routing Problem (VRP) which is intensively studied [6]. The VRP is a graph-theoretic problem. Eksioglu [6] defines the VRP as follows: Let $G = (V, E)$ be a complete graph where V is the set of vertices and E is the set of edges. Vertices correspond to customers, with the exception of $v_0 \in V$, which is the depot. Edges correspond to routes between customers and have a non-negative weight that represents the costs of moving from one customer to another. The goal of the VRP is to minimize the sum of the costs of a set of simple routes with the following restrictions:

1. every route visits the depot;
2. every customer is visited by exactly one circuit; and
3. the sum of the customers' demands of a route does not exceed the vehicle capacity.

In contrast to the static VRP situation, the problem under study assumes that jobs come up in real time and that AGVs in general directly get another job assigned after executing the current job. Therefore, the VRP is not applicable to our Automated Container Terminal model.

Research on the abstract idea of prioritized moving of multiple objects was done by Erdmann [7]. His idea was to consider the time-space path of a high prioritized object as a moving obstacle which has to be avoided by lower prioritized objects. The problem that was solved by Erdmann, relates to the problem of finding an optimal AGV driving strategy, as AGVs can be considered as objects with a certain priority that need to move through a certain space. There are however also several major differences that make the results from Erdmann not applicable to container terminals. Firstly, Erdmann assumes that all future drives are known at the beginning of the run, while in real-world container terminals, new jobs occur at different times. Furthermore, the system has no ability to handle deviations from the driving plans and finally, the strategy was applied only to very small examples with only a few moving objects.

3.2. Applications in Automated Container Terminals

A very important boundary condition for routing in a real operation environment like an Automated Container Terminal (ACT) is safety, i.e. the collision risk should be negligible. As the actual execution times of equipment regularly deviates heavily from the planned execution times [3], it is important that algorithms use real-time information of the system to adjust their planning. For safety guarantees however, most research assumes reliable communication between the AGVs and the central system or AGVs that are equipped with sensors to prevent collisions. In this thesis, we focus on ACTs without direct interaction between AGVs as this is still the case at most automated container terminals nowadays. Furthermore, although the communication between AGVs and the central system is in practice quite reliable, the system should also be safe when messages get lost. This restriction makes most of the available algorithms inappropriate.

3.2.1. Conflict-free routing of AGVs

A number of authorsaddressed the problem of conflict-free routing of AGVs. Krishnamurthy [12] proposed a method to optimize the makespan using column generation. They assumed a situation in which the job dispatching was already done. The collision-free routes of the AGVs are calculated consecutively, using the earlier calculated routes as inputs. Although this method gives no optimality guarantees, the authors present empirical results that are usually within a few percent of a lower bound that was found using linear programming. This method is not directly applicable to ACTs due to scalability issues and the lack of proper collision avoidance when AGVs deviate from their schedule.

Lee [13] presented a two stage collision-free driving technique for AGVs. In the first stage, the k shortest feasible paths between all pairs of stations are computed. These paths are stored in a table. This might be a time consuming job, but this is acceptable as this step is only executed once. During the execution of the algorithm, conflict-free routes are constructed in real-time using the table of stage 1. Link usage is stored in a link occupation table and used to prevent collisions with other AGVs. The authors show the real-time operation of the traffic control with an example. This example however, is extremely simple compared to the complexity of a real world ACT, using only 3 AGVs, small numbers of nodes and links, a maximum speed of $1 m/s$, and a fixed accelaration and decelleration rate of $1 m/s^2$. Furthermore, safety is only guaranteed if all vehicles drive as planned. Therefore, this approach is not applicable to real ACTs.

Oboth [15] investigated a method that solves the routing and the scheduling problem separately. The routing part was solved using a column generation method, comparable to Krishnamurthy [12]. The conflict-free routing was more dynamic than Lee's method [13], as jobs occurred randomly and vehicle speeds could vary. Similar to Lee, the routes were planned sequentially in order of increasing slack time. However, it does not provide a solution for situations where low-prioritized vehicles hinder higher prioritized vehicles. Furthermore, the system is not robust against unplanned braking of AGVs. Therefore, this approach is also not directly applicable.

3.2.2. Graph-based routing

A dynamic routing approach is proposed by Möhring [14]. In this model, the terminal layout is represented by a graph. As a preprocessing step, the physcial properties of the AGVs are used to determine for all edges what other edges cannot be used simultaneously. All edges maintain a time window to keep track of occupancy intervals. When an AGV is planned to drive along an edge, the related edges, as computed in the preprocessing step, are also blocked in that time interval. For new jobs, the chosen route depends on the availability of the edges on those routes. New driving schedules always respect the planned job occupancies of older jobs. This leads theoretically to conflict-free driving. In practice however, this approach does not provide safety guarantees in case of disturbances as it lacks a feedback system of the actual execution of driving plans.

Kalinovcic [10] applied a modified version of the banker's algorithm to route AGVs deadlock free. The routing network is modelled as a graph in which the edges represent track segments. These edges are used as resources in the banker's algorithm. The banker's algorithm is used to identify unsafe states that could lead to deadlock. To decrease the AGVs' waiting times, unsafe states are allowed under certain circumstances.

3.2.3. Track claiming

Van den Hof [19] investigated the idea of dynamic highway picking based on traffic density or job urgency. This approach led to a performance increase between 1 and 2 percent. Both the benchmark and the solution provide appropriate safety guarantees by letting AGVs claim the track segment they want to use. The central system grants permission to AGV's claim requests. Claim requests are only granted when the previous AGV that used that segment, has relinquished it. AGVs adjust their speed in such a way that it is always possible to halt before the end of the currently granted claim. Besides the safety requirement, also the other requirements, as stated in section 2.3, were met.

Another idea that was proposed by Van den Hof, is to use job priorities based on the available slack time of an order. This gave a performance improvement of 2.5% while meeting almost all our requirements. One important requirement however, was not met. The used implementation of the priority model, led on average to a deadlock situation every two hours, which is far too much. If we could use the idea's of priorities in our deadlock-free terminal layout, without validating the deadlock-free properties, this could lead to a significant productivity improvement.

3.2.4. Reduced inter-vehicle distance

The claiming approach from Van den Hof [19] is an effective and reliable collision avoidance strategy. The provided safety guarantees come with the cost of large inter-AGV distances. A way to let vehicles drive with shorter inter-vehicle distance, is platooning. Kavathekar defines a platoon as a group of linked vehicles that travel together and act like one unit [11]. Vehicles in the platoon obtain their driving instructions directly from the leading vehicle or from the infrastructure, for example by magnetic markers in the roadway.

Although platooning gives promising results, it is not applicable to our study as it needs communication techniques that are not possible with the available hardware. However, it is possible to decrease the inter-AGV distance by using the fact that driving AGVs have a breaking distance. It is therefore not necessary to be able to halt before the current position of the predecessor. It is enough to make sure that you brake fast enough to avoid a collision in case of an emergency brake of the predecessor. This method could decrease the inter-AGV distance, especially for driving at higher speeds.

3.3. Research directions

This thesis a number of new strategies is investigated to optimize the AGV driving on automated container terminals. Oboth showed that route scheduling could improve the efficiency of AGV driving, but the used strategy is not directly applicable to container terminals for two reasons. Firstly, Oboth's strategy is not scalable to the dimensions and number of AGVs that are found on real-world container terminals, and furthermore, it assumes a situation in which AGVs exactly drive as scheduled. Even small deviations of this schedule, could lead to collisions of AGVs. As the impact of colliding AGVs can be enormously, systems without collision avoidance guarantees, are simply not applicable to real terminals.

Oboth's route scheduling should thus be adapted to be more scalable and collision-free. To improve the scalability, the terminal is not modelled as a graph, which would be enormously because AGVs can make an very large number of curves, which should be added as edges to the graph. Instead, the terminal is modelled as a grid, without bookkeeping of the individual grid blocks. This gives the possibility to precisely define the occupation of the terminal by AGVs, but does not require a massive amount of memory. The bookkeeping, is done by cutting routes of an AGV into segments and registering all overlaps with segments of other AGVs, which is a technique that is based on the work of Möhring.

To guarantee collision-free driving of AGVs, we use the strategy of claiming exclusive access to segments, as described by Van den Hof in his research on AGV driving optimization. Determining priorities for AGV jobs, that is researched in the same work of Van den Hof, is also used, as this improved the QCs' performance significantly. Compared to Van den Hof, our research can determine conflicts much earlier by route scheduling and it is deadlock-free because it prevents deadlocks by smart design of the AGV driving lanes and by using no-stop claims to guarantee that AGVs that entered a crossing, will be able to leave it as well.

Compact rows are similar to platoons in the fact that both strategies are developed to decrease the distance between vehicles. The inter-vehicle distance is larger for compact rows than for platoons, but the distance can theoretically still be decreased significantly, leading to higher track capacities. This larger distance is caused by the fact that AGVs do not have hardware to communicate directly with each other or to measure the distance to their predecessor. The reduction of the inter-AGV distance compared to the traditional claiming strategy as described by Van den Hof, is achieved by using not only the current position of the predecessor, but also its speed and deceleration properties.

4

Methods

This chapter describes what methods are used to answer the research questions. To develop and evaluate our solutions, we use a state of the art industrial simulation environment. First, we set up a benchmark in this simulation. Subsequently, we add a module that gives vehicles a priority value that determines what vehicle gets priority when routes intersect. Next, we investigate what the impact is of Automated Guided Vehicle (AGV) greediness. We investigate whether a vehicle should always drive as far as possible, or halt sometimes a bit earlier to prevent unnecessary blocking of other vehicles.

Finally, we also discuss the possibility of driving with shorter inter-AGV distance, which we expect to be a promising research direction, as a starting point for further research. Due to the limited scope of this thesis and the large deviation from the rest of this research, no simulations are run for this topic. Our discussion about this topic can therefore be considered as a starting point for future research. Together with the description the methods, our hypotheses for the results of the simulations are stated. These hypotheses are compared with the actual results in chapter 5.

4.1. Simulation model

The used simulation model TIMESquare is a Discrete Event Simulation (DES) environment provided by the TBA group. This section describes the concept of DES, some relevant properties of the used TIMESquare environment and the settings that are used for the executed simulations.

4.1.1. Discrete Event Simulation

In a Discrete Event Simulation (DES), the simulation consists of a sequence of discrete events that occur at a certain time. The controller maintains a list of future events, ordered by the time they should occur. When the system executes an event, the clock makes a time step to the next event in the event list. The system's state can only be modified by events and therefore, the system does not change during the time step. Besides changing the state of the system, events can also schedule other events, either at the current time stamp or in the future.

In a DES, randomness is required to simulate stochastic variables, that occur often in simulations. In contrast to real randomness, simulations generally use pseudo randomness to be able to run a simulation again with the same random values. Simulation results are obtained by logging statistics during the run of a simulation. The number, type and level of detail of these statistics, depend on the goal of the simulation and are the result of a trade-off between extensiveness and high level of detail at the one hand and simulation speed and reduced memory usage at the other hand.

The events should be created in such a way that the event list never gets empty, because that would cause the simulation to stop. A common way to prevent this unintentional finishing of the simulation, is that moving entities in the model always schedule a next 'move'-event during the execution of their current 'move'-event, even when their speed equals zero. Because such models will never have an empty event list, another halting criteria is added to decide when the simulation should stop. This can be either a time-based criteria or a certain state of the system.

4.1.2. TIMESquare: the simulation environment

For our simulations, we use TBA's TIMESquare environment which provides a realistic and very detailed simulation of real-world container terminals. The model includes the behavior of QCs, AGVs, yard cranes, trucks and trains. Furthermore, several properties of vessel bays, containers and AGV tracks are simulated. The level of simulation detail differs per object. Terminal equipment like QCs, AGVs and yard cranes, are modelled very detailed, but objects that only use the terminal, like vessels, trains and trucks, are modelled in less detail. For QCs, the movements between transfer points, twistlock platform and specific bay positions, are simulated. The time used for hoist and spreader movements, container grabbing and dropping, and twistlock handling, are implemented as stochastic values. For yard cranes, the movements of the hoist, spreader and gantry are explicitly modelled. The level of detail of AGVs is also very high, including speed, load-based acceleration and deceleration, exact position, orientation and battery level. During the simulations, AGVs with low power levels go to the battery station to change their empty battery pack for a fully charged one.

Trucks, trains and vessels are modeled as objects that come to bring or pick-up containers. All containers are unique. A train or vessel that wants to pick-up containers, has the ids of these containers, and a loading plan that describes the desired position of these containers in the train or vessel. The brought containers have a known final destination and there is a yard management module in the model that decides where every container should be stacked. The operation of the yard module however, is beyond the scope of this research. The determination of the container positions, leads to AGV drive requests, which form an input for our research.

Although the whole simulation runs on a single processor, TIMESquare simulates a distributed system where the equipment and the central system cannot use each other's local data. The data is therefore sent to each other, while taking a realistic communication delay into account. The GUI of the simulation does use the local information of all equipment and is therefore a representation of the real current situation and not of the state that is currently registered by the central processor.

4.1.3. Simulation settings

To investigate the performance for various solutions and parameters, we execute a number of simulation runs. In a simulation run, we analyze a single combination of a solution with certain parameters. A simulation run exists of a number of replications of a long term peak simulation. Division into replications is done to reduce the impact of an unfavorable initial situation, which sometimes leads to a snowball effect with much congestion and slow AGV driving.

In every replication, a new peak simulation is started with new seeds for the random value generator. In a replication, all AGVs start in an idle parking position, and therefore it takes some time before the situation on the terminal approaches a real peak situation. The data of the first 2 hours of every replication is therefore ignored. Occasionally, an error occurs, which leads to a performance drop. As the performance drops under a certain level, usually indicating that an error occurred, the system recognizes this and cuts off the replication. If the error however, occurs late in the replication, it might not be noticed by the simulation itself. Therefore, the data of the last hour of every replication is also ignored.

In every simulation, the number of available AGVs is equal to five times the number of QCs, but they are not dedicated to a single QC. In general, a loading QC is served by more than five AGVs and a discharging QC by less than five AGVs. In a situation with an equal number of loading and discharging QCs, the loading QCs will typically be served by seven AGVs and the discharging QCs by three.

4.2. Benchmark simulations

In order to evaluate the results of the simulations, a good benchmark simulation has to be created. For our situation, we create two benchmark simulations. The first benchmark is a run without route scheduling. The second benchmark is a run where the routes are scheduled, but where priorities are granted via the same First Come First Served strategy as in the first benchmark. The second benchmark is used to show what the effect is of using the scheduling module, without using an improved priority strategy.

Hypothesis 1. *We expect that, compared to the first benchmark, the second benchmark will show (a) some more AGV waiting time, (b) more QC waiting time and (c) a lower QC productivity. (d) We expect that all these effects will be greater if the scenario size increases.*

The use of the scheduling module in the second benchmark could result in a performance decrease, since it introduces extra restrictions on the AGV driving behavior. The scheduling module keeps track of the order in which AGVs pass a certain track area and it only gives a vehicle access to a segment if it noticed that the segment was left by the previous AGV. If the segments of the AGVs are too large, then the scheduling module might be unnecessarily block pieces of the track that are already free. This causes AGV driving delay, which can lead to decreasing QC performance. Another issue that could make the performance in the second benchmark lower than in the first, is that schedules are created based on approximations of future driving behavior. The deviations of the driving planning, could also lead to delays for other AGVs. If the roads are crowded, this could lead to a snowball effect.

If the second benchmark shows performance that is significantly worse than in the first benchmark due to inaccurate drive planning, the accuracy can be improved at the price of slower computations by making the segments in the scheduling module smaller or by lowering the deviation threshold that defines when a route schedule should be recalculated.

To react on deviations of the route plan schedules, the scheduling module has functionality to recalculate the schedules whenever a vehicle deviates more than a certain threshold from its initial schedule. This should reduce the number and impact of delayed vehicles that unnecessarily block other vehicles. To analyze the impact of rescheduling of routes, we run the second benchmark in two variants, with and without updating the original schedule.

Hypothesis 2. *We expect that, compared to the benchmark without updates, the benchmark variant with schedule updates will show (a) less AGV waiting time, (b) less QC waiting time and (c) higher QC performance. (d) We expect that the described differences grows as the number of QCs and AGVs increases.*

As the schedules are based on approximations of vehicle speeds, AGVs will experience deviations from the route schedules. For certain conflict situations, this will lead to extra waiting, as the AGV that was expected to enter the conflict area first, is actually later than the other vehicle, causing a longer total AGV waiting time for this situation. We assume that all vehicles are equally likely to be delayed, and therefore, increased AGV waiting will lead to some more QC waiting, which directly decreases the QC performance. If the number of active AGVs increases, the number of conflicts of an AGV during a drive also increases, which makes the effect of a delayed vehicle on the other vehicles larger. Therefore, we expect that for larger scenarios, the impact of rescheduling will increase.

4.3. AGV Priorities

To take advantage of the scheduling module, we introduce the `Advanced Priorities` strategy to improve the first-come, first-served strategy that is currently used to determine priorities. In the current model, vehicles or routes have no priority assigned to it. Priorities should therefore be added to the scheduling module. For `Advanced Priorities`, we add a priority to claims, which are defined in section 2.4, to be able to determine the vehicle driving order. Two decisions have to be taken: how to determine priorities of a route and how to apply these priorities to the route planning. The ideal situation would be to compute the final outcomes of all possible priority decisions, every time a future conflict is detected and evaluate the outcomes using a reward function. Theoretically, this would lead to optimal performance. In practice however, this exact approach is not a feasible possibility.

The reason to renounce this exact solution, is that the situation in practice is too unpredictable. AGVs deviate from their schedules regularly and therefore, the system should update its schedule very often. This is especially a problem because the calculation itself is also computationally hard. This implies that an exact approach does not satisfy the scalability requirement.

Hypothesis 3. *We expect that, compared to the second benchmark, the `Advanced Priorities` strategy will (a) increase the total AGV waiting time, however, we expect that (b) the the waiting time of high prioritized vehicles in the `Advanced Priorities` strategy is smaller than the average AGV waiting time in the second benchmark, and that (c) the AGV waiting times of low prioritized AGVs is larger than the average in the second benchmark.*

Deviations from the priority determination of the first-come first-served strategy of benchmark 2, generally increase the total AGV waiting time for that conflict situation. The `Advanced Priorities` strategy is therefore expected to increase the total AGV waiting time. However, such changes are always made in favor of the AGV with the highest priorities, so if the distribution between low and high prioritized vehicles is balanced, than high prioritized vehicles will find decreased hindrance and low prioritized vehicles will have an increased waiting time. The expected difference between low and high prioritized AGVs, leads to the next hypothesis.

$$
\begin{aligned}
c_i(prio) &= c_i(urg) \cdot c_i(type) \\
c_i(urg) &= \begin{cases} \text{if slack } < -30 & 10 \\ \text{if } -30 \leq \text{ slack } \leq 170 & 10 - \sqrt{\frac{slack+30}{2}} \\ \text{else} & 0 \end{cases} \\
c_i(type) &= \begin{cases} \text{if loaded} & \& \quad \text{dest = QC} & 10 \\ \text{if empty} & \& \quad \text{dest = Stack} & 8 \\ \text{if empty} & \& \quad \text{dest = QC} & 6 \\ \text{if loaded} & \& \quad \text{dest = Stack} & 0 \\ \text{if} & \quad\;\; \text{dest = Battery} & 0 \end{cases}
\end{aligned}
$$

Figure 4.1: Used definitions to determine priority scores in the `Advanced Priorities` strategy.

Hypothesis 4. *We expect that, compared to the second benchmark, the* `Advanced Priorities` *strategy will (a) decrease the QC waiting time and (b) increase the QC performance with a few percents for 8 QCs and 40 AGVs. (c) We expect that these differences will be smaller for scenarios with less QCs and AGVs.*

The aim of the use of `Advanced Priorities`, is to decrease the delays for vehicles that are late, at the expense of waiting time for low prioritized AGVs. For QCs however, this means that the number of delay of AGVs that are to late, decreases, which to smaller waiting times and higher performance. As the priority implementation of our research is comparable to Van den Hofs approach, we expect a QC performance upgrade of several percents, compared to the second benchmark for the most realistic scenario size with 8 QCs, by using `Advanced Priorities`, which is similar to the improvement that was found by Van den Hof, when applying priority rules.

As stated in hypotheses 1 and 4, we expect that the second benchmark will perform worse than the first benchmark, and that `Advanced Priorities` will increase the performance of benchmark 2 in terms of QC productivity. Based on these two hypotheses, it is impossible to predict whether the `Advanced Priorities` will perform better or worse than the first benchmark. Although the aim of this study is to improve the terminal performance, we do have not enough information to make substantiated quantitative estimates of the impact of route scheduling, and therefore we cannot state a useful hypotheses about the differences in QC performance between benchmark 1 and `Advanced Priorities`.

Compared to the results of Van den Hof, our model decides in advance what AGV will get priority. At the one hand, this could lead to better results, as low prioritized vehicles can be scheduled in such a way that they never block higher prioritized vehicles. On the other hand, the information that is used to determine priorities far ahead, is inaccurate due to increasing deviations from initial driving schedules.

If the obtained results are worse than in Van den Hof's work, this is most likely caused by inaccurate time predictions, and better simulation results can then be obtained by taking more parameters into account for the route scheduling. It should however be avoided that the predictions are more accurate than the predictions in real terminals would be, because in real terminals, the driving times of AGV are stochastically distributed, and they will therefore deviate from the schedules. The best solution would be to make the predictions very precise and use (pseudo) random values to simulate stochastically distributed AGV driving times. Due to time limitations however, this strategy is left out of the scope of this thesis.

4.3.1. Determining priorities

For the determination of priorities, we extend the priority rules of Van den Hof [19] with heuristics that were provided by domain experts. Van den Hof uses the weighted average of 4 scores to determine a vehicle's priority for a certain claim, namely the urgency score, time score, speed score and type score. In the computation of our priority score, we remove the speed and time scores, change the computation of the other scores and use a non-linear calculation to determine the priority score. The speed score is removed as it is not applicable to our situation, because we calculate the priority for future situations, for which we do not know what the speed is of the concerned AGVs. The effect of the time score is tested separately in section 4.4. The used definitions of the scores are based on, but not exactly the same as the definitions used by Van den Hof, and they are described in figure 4.1.

Due to the urgency score, delayed vehicles get priority over vehicles that have some slack left. The urgency score grows quadratically when the slack decreases, which is done because the available slack becomes more important when in approaches 0, or is even negative. If the slack however, is smaller than -30, which means that the AGV is already substantially behind schedule, the urgency value is always equal to 10, as it is already

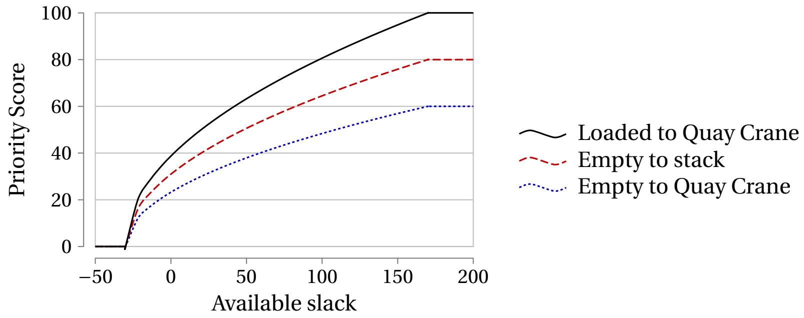


Figure 4.2: Priority score of AGVs based on the available slack for different driving types.

clear that the QC will have to wait for the AGV which means that every second of extra AGV delay directly leads to an extra second of QC delay, independent of how long the QC already has to wait.
The type score covers the destination of the AGV and whether it is loaded or not. The highest values are assigned to AGVs that are driving to the QCs, as QC waiting time reduction is the main goal of using priorities. AGVs carrying a container are prioritized over empty AGVs, as loaded AGVs need to come in the right sequence, but the sequence of empty AGVs can be swapped. AGVs that are driving empty to the stack, do this to pick up a container that has to be driven to the QCs, so these drives are also part of the process of bringing a container to a QC. The priority is however, lower than for AGVs that are already driving loaded to a QC as the drive to the stack is generally planned in such a way that the AGV can leave the stack with enough slack for its drive to the QC transfer point. The priority score as a function of the available slack and the driving type, is drawn in figure 4.2.
Both the urgency and the type scores have a range from 0 to 10, and contrast to Van den Hof, we do not use the sum, but the product of these two scores, to better express the cohesion of these scores in the priority. As it takes at least 48 hours of multiple cpu's to run a single simulation, it is infeasible to search for the optimal priority formula. Due to the extent of this thesis, we only use a single formula to calculate the priorities. Based on our results however, we provide some suggestions for further research on priorities in chapter 6.

4.3.2. Deadlock prevention mechanisms

The determined priorities of the jobs should not lead to deadlock situations. Therefore, no group of AGVs should be mutually waiting for each other. To avoid deadlocks, we aim for a consistent priority determination. First of all, priorities should be always unique. This is obtained by using the vehicle id as a tie breaker in case of equal priorities. Furthermore, vehicles that are blocked by the claiming system, should have a lower priority than the vehicle that blocks them. Among other situations, this happens when vehicles drive in a row, the front vehicle should then have a higher priority than the following vehicle(s). Thus, if two vehicles drive partly the same route, then the following should hold:

Heuristic. *For vehicles X and Y that drive partly the same route in the same direction, say from A to B, it holds that from the moment on that both vehicles have requested a claim beyond A, the vehicle whose claim was granted first, say X, should uninterruptedly hold a higher priority than Y, until X releases its last claim on a segment between A and B.*

Note that the time interval as described in this heuristic will be empty if X has released its last claim between A and B, before Y requests a claim between A and B, i.e. when the vehicles drive far from each other. The system may deviate from this heuristic if the distance between A and B is larger than a threshold α. In that case, the heuristic only holds when the distance between the two vehicles is smaller than threshold β.

4.3.3. Strategies to apply the priority value

The found priority value can be applied in different ways to determine the order in which vehicles should cross conflict areas. To decide what way is best applicable to our research, we use the following principle: Simple approaches are preferred over complex ones, provided that the difference in expected quality is not too large. It is worth noting that the choices that will be described below, do not affect the practical value of the solutions, but only the performance.
The determination of driving order could be done only between pairs of vehicles, for clusters of vehicles that cross the same area, or for the whole system at the same time. We choose to apply the priorities only to pairs

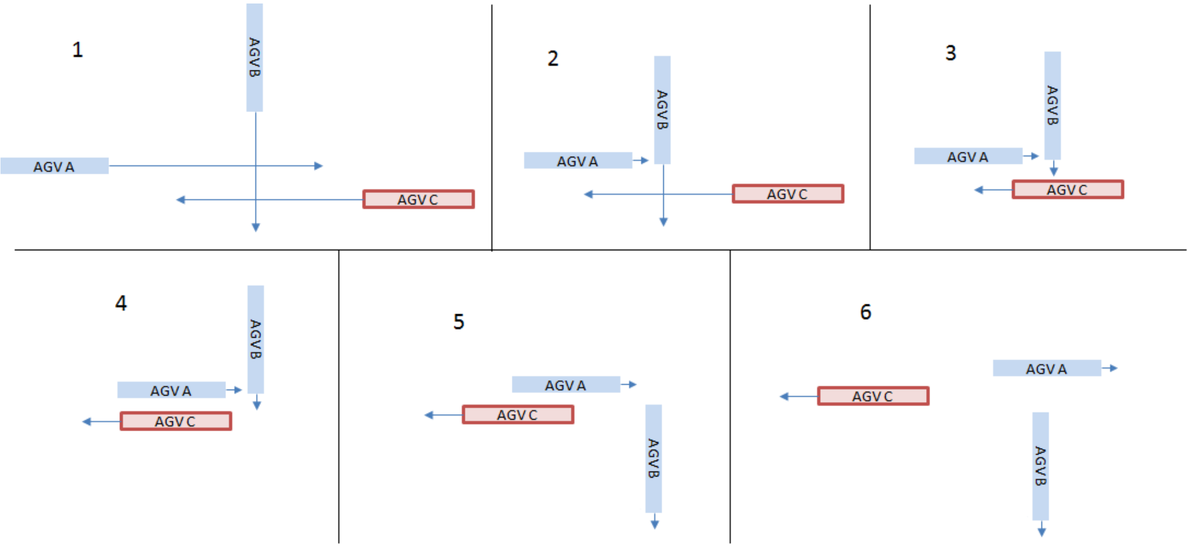


Figure 4.3: Schematic representation of a situation where a greedy strategy leads to unnecessary waiting. AGV *C* has a high priority and AGV *B* is blocking AGV *A* due to its `greedy` strategy

of AGVs as it takes less computation time. Furthermore, there are different ways to apply the priority values. The easiest application is to always give priority to the vehicle with the highest priority value. An alternative could be to use several priority classes and apply a First Come First Served strategy for sets of vehicles that are in the same priority class. The benefit of this approach is that the number of route updates could be reduced, which makes the computations faster, without large effects on the performance. Because speed is not yet very important for the simulation, and because the implementation of priority classes is relatively complex, we always give priority to the vehicle with the highest priority, independent of the difference.

4.4. AGV greediness

Due to the use of route scheduling, AGVs know in advance that they will have to wait in the future. AGVs can handle this information is different ways, ranging from greedy to defensive. The most greedy strategy is to drive always as far as allowed. The most defensive way is to always wait until your schedule does not interfere with any other driving plan. It is also possible to use strategies that combine greedy and defensive driving.

It is hard to predict how well a strategy will work as this heavily depends on the AGV's deviations from their route plan and therefore, we run simulations to see what strategy performs best. For an individual AGV, greedily driving is always the fastest strategy, for the system as a whole however, it might be better to drive more defensively. In this research, we compare two greediness strategies.

For first strategy, called `greedy`, all AGVs drive greedily, i.e. always drive as far as allowed. The second approach, `defensive`, is a less strict strategy, in which a vehicle always plans its drives greedy, but when it finds out that it will block another vehicle while it is waiting, it tries to wait at an earlier position, where it does not block the other vehicle.

The main benefit of using the `greedy` strategy, is its simplicity in both implementation and analysis. It is however, not hard to come up with realistic scenarios in which the greedy strategy clearly does not provide optimal results, of which an example is given in figure 4.3. In this example, the red AGV *C* has a high priority and AGV *A* and *B* have a low priority. AGV *B*'s route crosses the routes of both other vehicles, so two conflicts have to be handled: the conflict between *A* and *B* and the conflict between *B* and *C*. In the first conflict, *B* will arrive earlier at the conflict area, so it gets priority over *A* (figure 4.3 (2)), and *A* will have to wait until *B* has passed the conflict area, before it can continue (figure 4.3 (5-6)). For the crossing between *B* and *C* however, *C* gets priority, because it has a high priority (for example because it is loaded moving towards a QC). Therefore, AGV *B* has to wait until AGV *C* has left the crossing (figure 4.3 (4-5)).

If we consider the situation of figure 4.3 while using `defensive`, we obtain the situation as drawn in figure 4.4. Here, AGV *B*'s original waiting position blocks AGV *A*, and therefore it shifts its waiting position a bit backwards, to prevent AGV *A* from waiting. The benefit for *A* is much larger than the delay for *B*, so the system as a whole is has a faster AGV flow. Note however, that the decision of *B* to adjust its waiting position, is made well

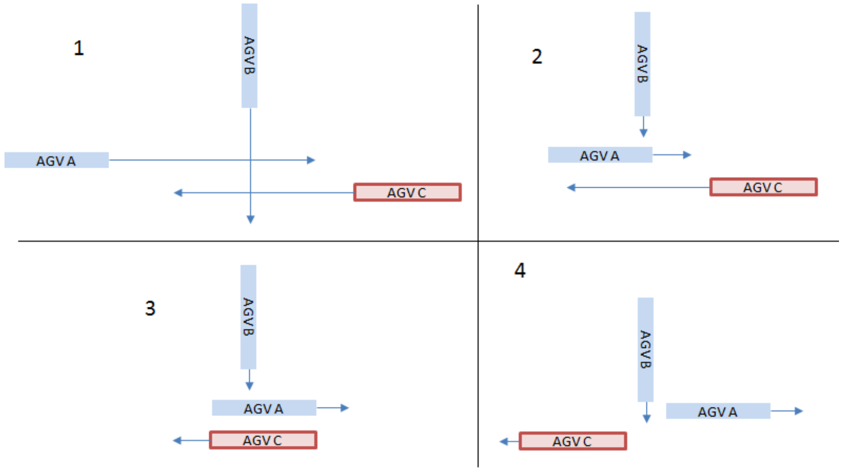


Figure 4.4: Schematic representation of a situation where the `defensive` strategy prevents unnecessary waiting. AGV *B* has to wait for AGV *C*, and halts some earlier to give AGV *A* enough space to pass.

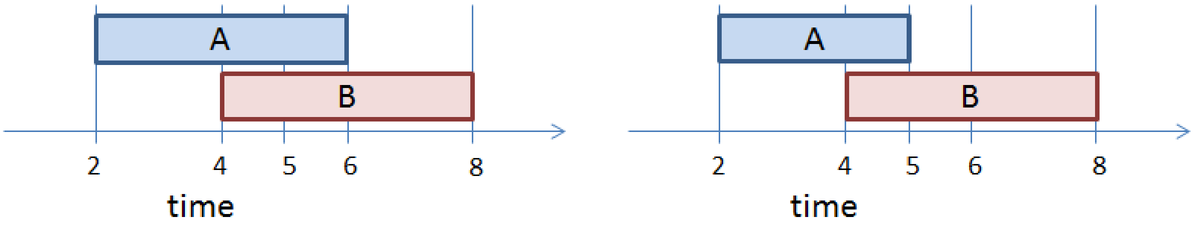


Figure 4.5: Schematic example of conflicting desired claim occupations of a certain area by A and B (left), and the same situation with a shorter occupation time of vehicle A (right). The shorter occupation of A leads to a higher relative priority value as the potential hindrance for B is decreased.

before the actual waiting is done. AGVs do not drive backwards to reach a better waiting position. A possible drawback of `defensive` is the possibility of starvation. If an AGV wants to cross a six-lanes highway, without hindering any other vehicle at a waiting position, this strategy could lead to long waiting times. By applying the priority strategy as defined in section 4.3 however, the priority of the vehicle steadily grows as its slack decreases, making it more likely that it will have a higher priority than the vehicles on the highway.
To determine priorities in the `defensive` strategy, the order of priority is not solely based on the priority score of the vehicles, but also on the time score that describes how much delay it costs an AGV if the other AGV would go first. The time score depends, contrary to the urgency- and type score, not only on the status of the AGV itself, but also of the conflicting AGV. Note that a shorter requested occupation time of a claim results in a lower time score of the conflicting AGV, causing an increased relative priority, as displayed in figure 4.5. At the left image, the time score of B is $c_B(time) = c_A(req\_out) - c_B(req\_in) = 6-4 = 2$, while at the right image, $c_B(time) = 5-4 = 1$. In practice, this means the expected speed, which is in our current implementation solely based on the type drive (straight, curve or crab), is implicitly taken into account in the computation of the priority scores.
We expect that the priority and greediness strategies do not show significant correlations and therefore, we do not consider all combinations to find the global optimum. Instead of finding the global optimum, we look for the best priority strategy using `greedy`, and then apply both greediness strategies on that priority strategy. The `greedy` strategy is used as the default strategy as the AGVs in the original system also drive greedily. Researching whether priority and greediness strategies interfere, can be done as further research and are out of scope of this thesis.

Hypothesis 5. *We expect that for the greediness strategies, (a)* `defensive` *will provide a significantly lower AGV waiting time than* `greedy`*, that (b)* `defensive` *has a little lower QC waiting times and (c) a little higher QC productivity.*

The `defensive` strategy is developed to decrease the total AGV waiting time of vehicles, which justifies our expectation of less AGV waiting time. The used time score has however, a limited influence on the priority decisions and acts therefore in practice often as a tie breaker for almost equally prioritized AGVs. Therefore,

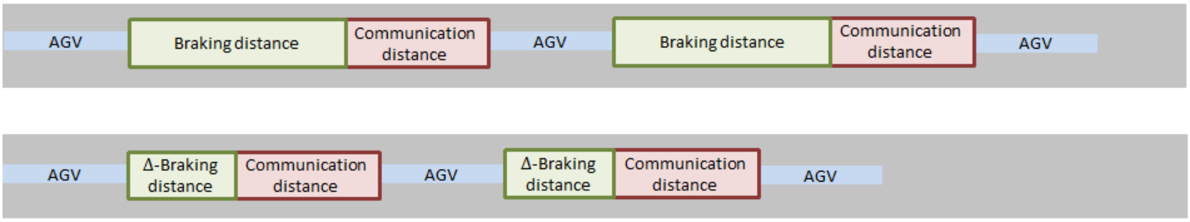


Figure 4.6: Schematic representation of the AGV between a normal row (top) and a compact row (bottom) of AGVs that drive at the same speeds. With compact rows, the required space on the track is significantly reduced.

the expected influence is small. Due to the limited influence of the time score, a large difference in expected delays is necessary to change the priority order. We expect that for these situations, it is useful to change the order of priority and that this will cause a slightly lower QC waiting time and hence a higher QC productivity.

4.5. Decreased inter-AGV distance: Compact Rows

Although the application of smaller inter-vehicle distances seems to be promising, the implementation and execution of these simulations is out of the scope of this thesis as it cannot be implemented as a part of the scheduling module. Shorter inter-vehicle distance requires a less restrictive strategy to grant claims and is therefore fundamentally different than the current route scheduling module as this only limits the granting of claims in some situations. As an extension of this research, we describe how compact rows could be implemented.

In Kavathekar's definition, a platoon is a group of vehicles that acts like a single unit [11]. This includes that vehicles are accelerating and braking at the same time. With the technology that is currently used on most terminals, there is no direct inter-AGV communication possible, nor do the AGVs have sensors to measure the distance to their predecessor. The information that AGV A has about AGV B, has to be provided by the central system. It takes therefore two times the communication delay between central system and AGV, before this information is shared. This makes it impossible for our research to apply platooning as Kavathekar defines it. The described driving with smaller inter-AGV distances, does therefore not fit in the platooning definition, however, to a certain extent, it is comparable to platooning, and therefore we use some ideas that come from platooning. To keep the distinction with platooning clear, we will use the term 'Compact Row' instead of 'Platoon' for our strategy to decrease the inter-vehicle distance.

4.5.1. Defining compact rows

Informally, a compact row is a group of vehicles that drive with reduced inter-AGV distance. The formal definition of compact rows is as follows:

Definition. *A compact row is a group of at least two vehicles, that drive along the same path and in the same direction, in which all vehicles, except the first, use both the position and speed of their predecessor to adjust their own speed.*

When AGVs drive in a normal row at constant speeds, the inter-vehicle speed consists of two parts, namely the braking distance and the communication distance. The breaking distance guarantees that the vehicle can halt before the last known position of the predecessor, while the Communication Distance ensures that the following vehicle can continue driving its constant speed until it receives a new position update of its predecessor. Under the assumption that the back AGV can brake at least as fast as the front AGV, it is possible to drive with just the Communication Distance as inter-vehicle distance.

In our model, as well as in reality, there are situations where the front AGV can decelerate faster than the back AGV, for example when the back AGV is loaded and the front AGV empty. To guarantee collision-free driving for this situation, we add some Δ-Braking Distance between the AGVs, which compensates for the difference in braking power. If AGV A drives in front of AGV B, the Δ-braking distance is calculated as:

$$\Delta\text{-Braking Distance} = \text{Braking distance } B - \text{ Braking distance } A.$$

In this computation, we use for the computation of the braking distance of the front vehicle, the maximal deceleration that is possible for any AGV on the terminal, leading generally to braking distances that are shorter than the AGV could achieve, forcing the following vehicle to drive on a larger distance than actually necessary. The result of using compact rows is shown schematically in figure 4.6.

4.5.2. Compact row strategies

For the implementation of compact rows, a number of decisions has to be taken. A first major decision is whether or not to use active formation strategies. In active row formation, the system considers the routes of the vehicles and decides what vehicles could form a compact row together. In contrast to active row formation, it is also possible to implement compact row driving as passive system. In that case, rows arise spontaneously when two vehicles drive in the same direction. The main principle of compact rows, driving with shorter inter-vehicle distance, can then still be applied. In case of passive rows, no further decisions have to be taken. For active row formation however, a number of other choices has to be made.

For active row construction, the central system decides what vehicles will form a row. To make this decision, it should consider what delay is acceptable, to be able to participate. If a vehicle has to wait too long before it can join a row, it might be better to drive individually to its next destination.

The order of the vehicles in a row is another relevant property. To drive smoothly, it is preferable that the vehicles leave the row in reversed driving order, i.e. The back vehicle leaves first and the front vehicle is the last one that leaves the shared track. As vehicles cannot overtake each other, this order should be achieved by the insertion strategy. If, by way of contrast, a middle vehicle Q leaves the group, a gap arises between the predecessor P and successor R of Q. This distance between P and R is even larger than before Q left, because Q will decrease its speed before leaving the track, which forces R to decelerate as well until Q fully left the track.

5

Results

This chapter gives an overview of the experimental setup of the simulation scenarios in section 5.1, followed by an analysis of the benchmark simulation results in section 5.2. Subsequently, the outcomes of the different priority simulations are described in section 5.3 and finally, the results of the greediness research are discussed in section 5.4.

5.1. Experimental setup

To investigate the effectiveness of our solutions, we run a number of simulations in the TIMESquare simulation environment, as summarized in table 5.1. In all simulations, we use the same terminal layout, equipment types and number of AGVs per QC. The only thing that differs in the scenarios, is the scenario size, in which the number of QCs and AGVs proportionally change, and the behavior of AGVs. Every scenario consists of at least 100 hours of simulation to reduce the width of the 95 % confidence interval.

5.1.1. Benchmark simulations

We execute simulations to test two separate ideas, namely priority determination and greediness. To reduce the number of simulations, we do not investigate all combinations of priorities, and greediness strategies.
The first simulations to run are the benchmarks simulations. Our first benchmark is a simulation in the original TIMESquare model without any modifications (A). Subsequently, we run a simulation run with the same terminal layout and equipment, but with the route scheduling module turned on (B & C). We run this simulation in two variants, in the first variant, we calculate a schedule and do not reschedule the drives when a vehicle deviates from its schedule (B). In the second variant, the the driving scheduled is updated when the vehicle deviates more than 30 seconds from its initial plan (C).
The used priority strategy in these runs is to assign priorities based on a first-come first-served strategy, which is also used in the original system. Therefore, the applied strategies of these benchmarks are equal. The only differences with the original system that occur in the simulations, are caused by AGVs' deviations from their route schedules, which are a result of inaccurate route scheduling and stochasticity of equipment speeds. These two benchmarks show the impact of using our implementation of route scheduling.

	Experiment Name	Route Scheduling	Update threshold	Priorities	Greediness
A.	Benchmark 1	off	n.a.	FCFS	Greedy
B.	Benchmark 2 no updates	on	n.a.	FCFS	Greedy
C.	Benchmark 2 with updates	on	30 s.	FCFS	Greedy
D.	Advanced Priorities	on	30 s.	Advanced Priorities	Greedy
E.	Defensive	on	30 s.	Advanced Priorities	Defensive

Table 5.1: Overview of the simulations to run

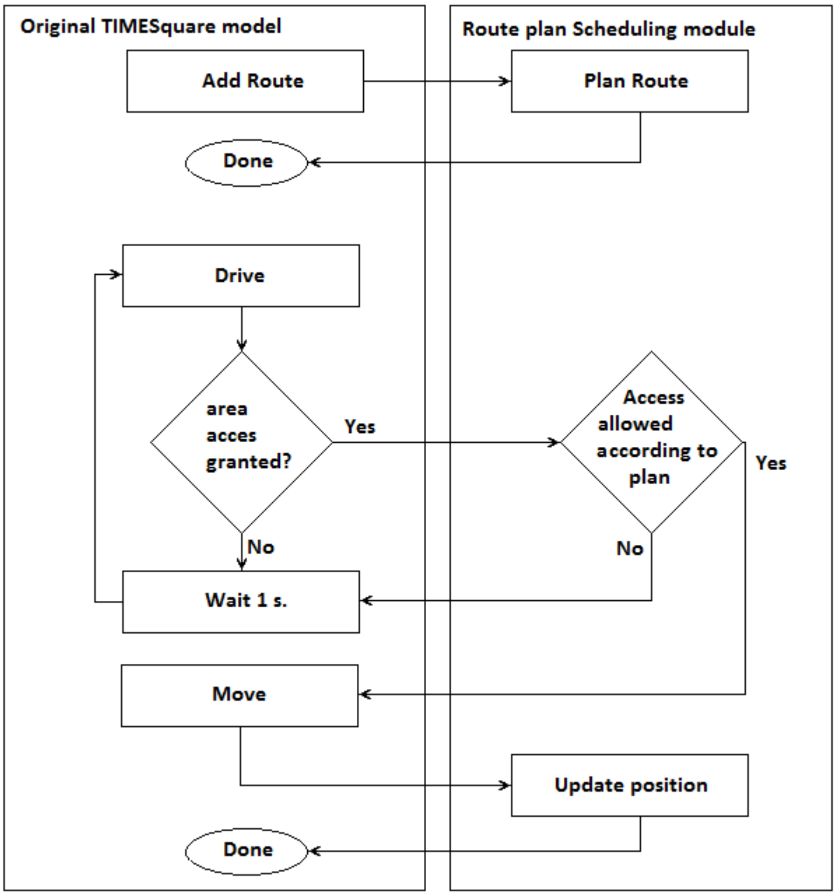


Figure 5.1: Simplified overview of the most important interactions between the original TIMESquare model and the developed Route Plan Scheduling module.

5.1.2. Priority and greediness strategies

To investigate the impact of using priorities for AGVs, we execute some extra simulation runs. For all these runs, we use a priority value that is a combination of the so called `urgency` and `type` score, which are described in detail in section 4.3.1 (D).

To research greediness, we implement two greediness strategies, `greedy` and `Defensive`, which are described in section 4.4. The default behavior of the system, with respect to greediness, is the `greedy` strategy. Therefore, we use run D as the greedy strategy and run a new simulation (E) with the less greedy `Defensive` strategy.

5.1.3. Implementation of Route Plan Scheduling module

The developed Route Plan Module is designed to be a separate module with a minimal number of interactions with the original system, to be able to connect the module to different versions of the TIMESquare model. The most important connections with the TIMESquare model are schematically displayed in figure 5.1. The first interaction is with the new module is used when an AGV receives a new route. The Route Plan Scheduler directly calculates the occupation times of the route's track segments. If future conflicts with other AGVs are detected, the planning module decides what vehicle should go first and delays the schedule of the other vehicle.

During the execution of the simulation, the central system regularly calls the Drive method of AGVs which first checks if the vehicle can access the desired area. If no access is granted, then the AGV has to wait some time before it retries. If the original model wants to grants access however, the route plan module compares this request with the created route plans to see if access can be granted definitively. In that case, the vehicle starts to move and the new driving position is sent to the scheduling module. The updated position is used to see if some track segments are left and can therefore be released.

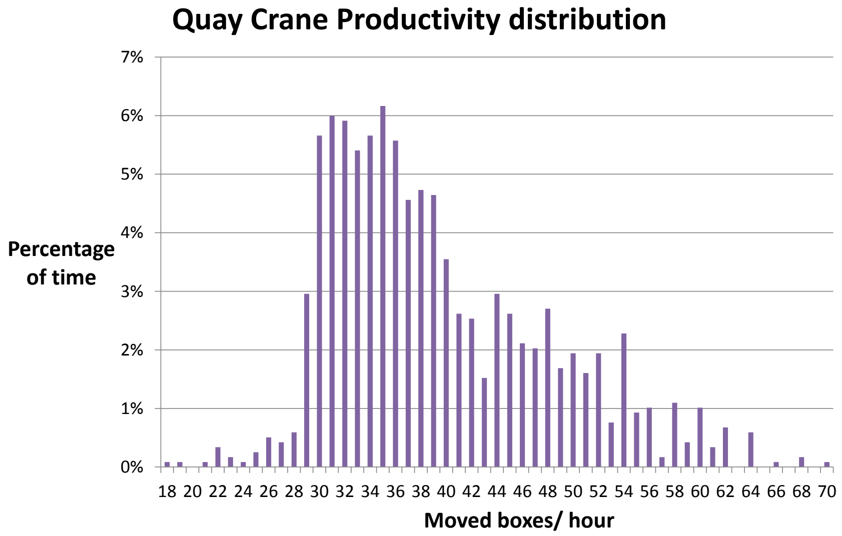


Figure 5.2: Quay Crane productivity distribution when serving 8 QCs with 40 AGVs in the original system.

5.2. Results of the benchmark simulations

This section gives an overview of the benchmark simulations, which are split up in a benchmark that runs without the `Route Plan Scheduling` module, a benchmark that runs with `Route Plan Scheduling` turned on, but without recalculation of schedules, and a benchmark with route scheduling with recalculation of schedules. All benchmark runs use the same First Come First Served strategy to determine priorities.

5.2.1. Benchmark 1 - Quay Crane productivity distribution

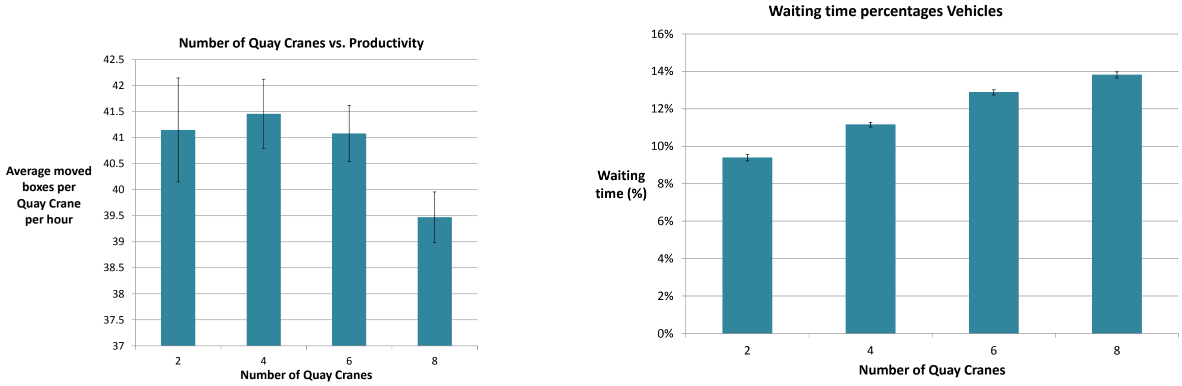
The type of jobs a QC is executing, loading or discharging a vessel, is stochastically determined. On average, discharging goes faster than loading, because no specific AGV is needed to drop a container to, but the loading of a vessel is performed in a predefined sequence, so the QC is sometimes waiting for a certain container while another container is already available for loading. Furthermore, an empty AGV can drive directly to the QC, while a loaded AGV has more risk of delay as it has to drive to the stack first, and also has to be loaded by a yard crane that can be delayed as well. The number of moves a loading QC performs in an hour is therefore often lower than the number of moves executed by a discharging QC.

The productivity of individual QC-hours shows relatively large deviations, as can be seen in figure 5.2. From this figure, it turns out that the largest fractions are found between 30 and 40 moved boxes per hour and that also a substantial number of hours, between 40 and 55 boxes are moved. Because the Quay Cranes can either handle a single 40 ft. container or two 20 ft. containers in a single move. On a vessel, the distribution between 20 feet containers and 40 feet containers is made per bay. As it takes on average a couple of hours to load or discharge a bay, QCs only occasionally change between 20- and 40 ft. containers.

Because handling two 20 ft. containers takes approximately the same amount of time as handling a single 40 ft. container, the QC productivity in terms of moved boxes is significantly higher for handling 20 ft containers. The hours in which the productivity is above 42 moves, the QCs were performing at least a part of their time 20 ft. containers. In figure 5.2, it can be seen that for 48 or more moved boxes per hour, the percentage of hours with an even number of moved boxes is significantly higher than the percentage of hours with an odd number of moved boxes, which is likely to be caused by QCs that are handling only 20 ft. containers for a full hour. The absence of hours in which an odd number of boxes, higher than 62, is moved, suggests that these performances are all reached in hours a QC was handling 20 ft. containers the full hour.

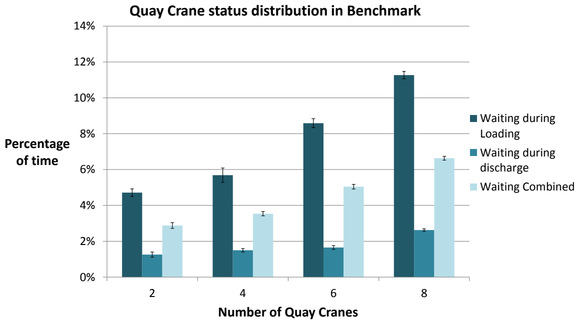
5.2.2. Benchmark 1 - Quay Crane productivity versus scenario size

In these benchmark runs, the terminal always has 5 AGVs per QC, but the number of QCs varies, ranging from 2 to 8, in steps of 2 added QCs and 10 added AGVs. All these simulations are run a number of times that is large enough to decrease the halfwidth of the 95 % confidence interval to at most 1,00 QC move per hour, and at least provides 100 net hours of simulation. In general, only for the scenarios with 2 QCs, the number of simulation hours exceeded 100 in order to decrease the size of the 95 % confidence interval of the QC pro-



(a) Relation between the number of available AGVs and QC moves per hour.

(b) Waiting times of the vehicles during their drive in the benchmark situation.



(c) Quay Crane waiting percentages for loading and discharge of vessels and combined.

Figure 5.3: Key performance indicators of the simulation in the original system.

ductivity. The average number of hours for these simulations is in the range between 160 and 180 net hours of simulation. As displayed in figure 5.3a, there exists a clear relation between the number of used QCs and the QC's performance when the number of QCs is in the range between 2 to 2 QCs. The number of QCs is not increased further than 8 active QCs as the space along the quay is too small to contain more than 8 QCs with sufficient space for the AGVs to drive around the QCs. The number of simulation runs in our benchmark was too small to discover any significant differences in performance for scenarios with 2 to 6 QCs. The measured number of QC moves per hour for these scenarios is around 41 box moves per hour. For 8 QCs however, the performance is significantly smaller than for 6 QCs, which is mainly caused by more crowded tracks that lead to driving delays.

The decreasing QC performance for increasing scenario sizes, is a direct consequence of the delayed AGVs that cause QC waiting times. In figure 5.3c, the waiting times of QCs for both loading and discharging QCs is shown for simulations with different numbers of QCs. The combined waiting time is also added and shows what the the percentage of the time a QC is waiting for an AGV, regardless of the type of operation it is executing. This value is typically not equal to the average of the waiting percentages for loading and discharging.

Loaded AGVs cause the QCs more waiting time than empty AGVs, which we expect to be a result of the fact that loaded AGVs are more likely to be delayed as a loaded AGV has to pick-up its container at a fixed point in the stack, while for an empty AGV, the nearest available AGV can be chosen. Furthermore, loading a vessel is performed in a predefined sequence, so the order of two loaded AGVs cannot be swapped if the second is already near to the QC, while the first is still further away. Empty AGVs however, are interchangeable, making it possible to decrease the QCs' waiting times in certain situations. Finally, there is also a chance of delay due to lateness of the stacking crane that has to deliver the requested container in time at a transfer point.

Waiting times	% of late vehicles	Contribution to total delay (%)
1 - 15 s.	36	3.4
16 - 30 s.	16	4.4
31 - 60 s.	17	8.9
61 - 90 s.	9.0	8.2
91 - 120 s.	5.9	7.6
121 - 180 s.	5.9	11
181 - 240 s.	3.5	9.1
241 - 300 s.	2.0	6.9
300 - 600 s.	2.6	14
> 600 s	2.4	27

Table 5.2: Distribution of Quay Crane waiting times for vehicles that are too late in the benchmark simulation with 8 Quay Cranes and 40 vehicles. Note that the time intervals are heterogeneous.

5.2.3. Benchmark 1 - Quay Crane and AGV waiting times

For both loading and unloading QCs, the QC waiting time percentage increases as the scenario size increases. For 4 to 8 QCs, the waiting time percentage for loading grows approximately linearly from 5.7 % to 11.3 %, while the growth between 2 and 4 QCs is a lot smaller. For discharging however, the waiting time percentage grows only from 1.3 % to 1.7 % in the range from 2 to 6 QCs, but shows a large increase from 1.7 to 2.6 % when 2 more QCs are added. For further analysis, the status distribution of AGVs is investigated.

Figure 5.3b shows that the added waiting time for AGVs when the number of AGVs grows from 30 to 40 AGVs (6 to 8 QCs), is smaller than the difference in waiting times between 20 and 30 AGVs (4 to 6 QCs). The relative large increase of QCs' waiting times for empty AGVs, can therefore not directly be explained by the increased traffic congestion. A possible explanation for the increased time QCs are waiting for empty AGVs, is that up to 6 QCs, the driving delay that is caused due to congestion, can be compensated by the slack that is added to the route plans. For 8 QCs and 40 AGVs however, this added slack is too small for a lot of AGVs, causing a large increase of vehicles that are too late at the QC and hence QC waiting times. As load moves are longer and have therefore in general more delay, the delay is already too short for a substantial part of the AGVs, when the number of QCs is still small. The described effect for empty vehicles is therefore not visible for loaded AGVs.

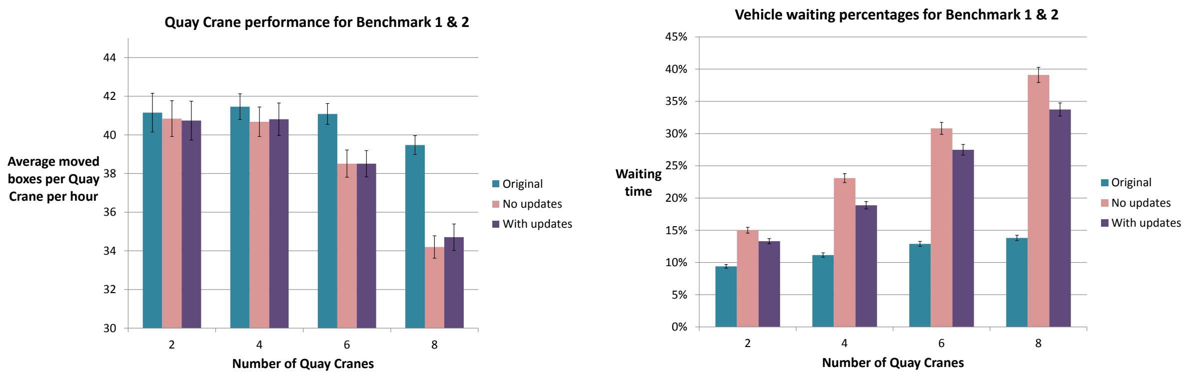
In the benchmark simulation with 8 QCs and 40 AGVs, more than 87 % of the AGVs was in time to be handled by the QCs, the 12.6 % of the vehicles that were too late at the QC transfer point, were together responsible for the 6.6 % waiting time of the QCs (figure 5.3b). To better understand the nature of the waiting times, table 5.2 shows the occurrences of the several waiting times. Less than 11 % of the delayed vehicles is more than 3 minutes delayed, however, they are responsible for 57 % of the total delay. Strategies to reduce the QC's waiting times, should therefore focus on reducing the number of vehicles with large delays.

5.2.4. Benchmark 2 - Route scheduling with a first-come first-served strategy

The second benchmark shows the impact of our implementation of the route scheduling module. This module uses the same strategy to calculate priorities as the original system in benchmark 1. In these runs however, the priorities are calculated some time ahead, leading to inaccuracies due to stochasticity of equipment speeds and simplifications in the calculations. The second benchmark consists of two scenarios. A simple scenario, `FCFS without updates`, in which AGVs schedule their route plan and the corresponding priority decisions once, and do not update them, and a more complex scenario, `FCFS with updates`, in which AGVs recalculate their route schedule if the real position differs too much from the planned position.

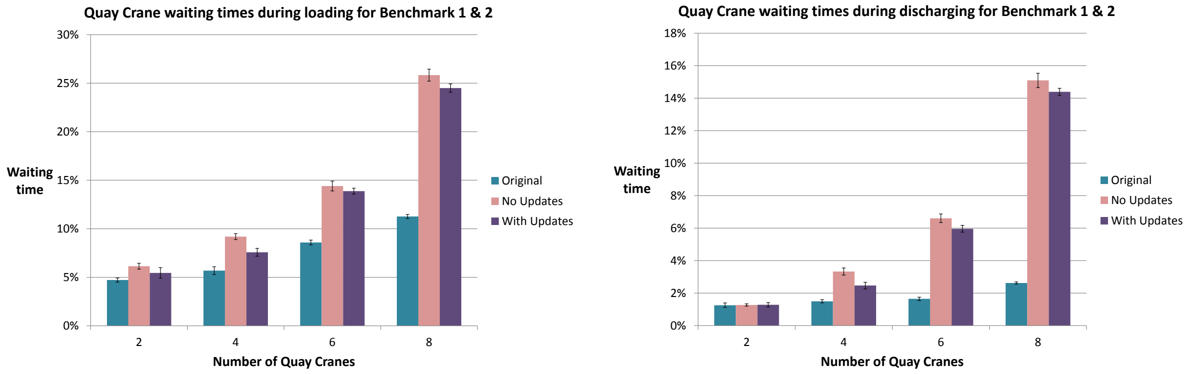
From figure 5.4a, we see that for up to 4 QCs and 20 AGVs, no significant difference in performance is found between the original system and the two scenarios of benchmark 2. Starting from 6 QCs and 30 AGVs however, the performance of the system with route scheduling is lower than the original system. The decreased performance for more crowded scenarios is in accordance with hypothesis 1c and 1d as the route scheduling in the benchmark does use the same priority strategy as the original system, but is less accurate due to the planning that is done some time in advance. This inaccuracy sometimes leads to inverted priority determination. As the original system always gives priority to the first AGV, a change of priority generally means a longer total waiting time. For scenarios with more AGVs on the roads, unnecessary waiting AGVs will also more often block other AGVs, which leads to an increased impact of the scheduling module.

Although the existence of decreased QC performance in benchmark 2 was predicted in hypotheses 1c, the ex-



(a) Quay Crane performance.

(b) Vehicle waiting times.



(c) Quay Crane Waiting times during vessel loading.

(d) Quay Crane Waiting times during vessel discharging.

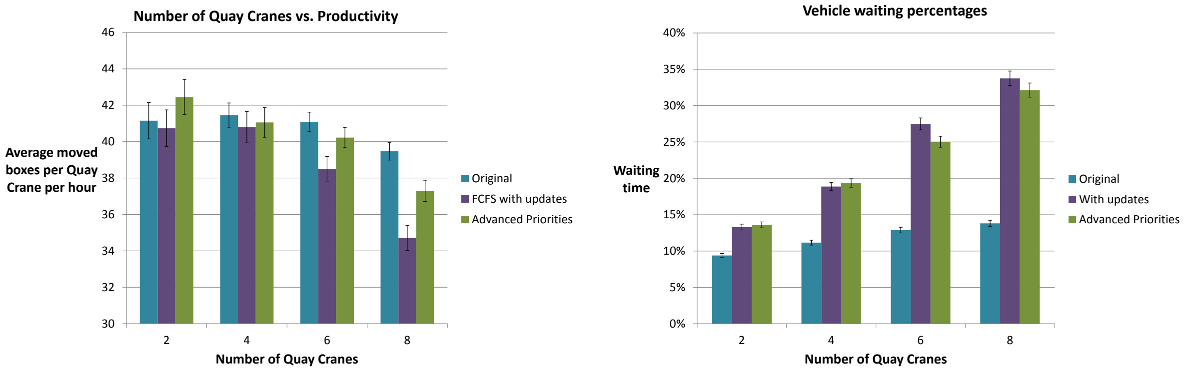
Figure 5.4: Impact of scheduling and rescheduling of routes on the vehicle waiting times, the Quay Crane waiting times and the Quay Crane performance for different scenario sizes.

tent of the measured performance drop is larger than expected. It would be highly desirable to use a scheduling implementation that performs better on benchmark 2, to be able to pronounce clear statements based on the results of the other scenarios. The performance of the route scheduling module could be improved by making the plan predictions more accurate. Unfortunately, the current implementation can already hardly satisfy the scalability constraints, and adding more precision without violating the scalability too much, requires major revisions of the module that are not feasible to be realized within the scope of this thesis project. The development of a more precise route scheduling implementation however, is a major recommendation for future work, both for scientific and industrial purposes. Our ideas about how this extra accuracy could be achieved, are described in chapter 6.

5.2.5. Benchmark 2 - Intermediate schedule updates

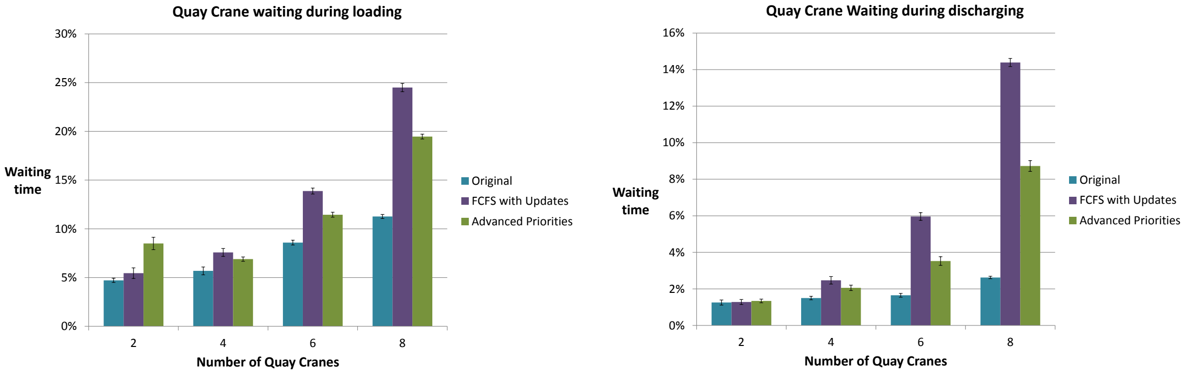
In contrast to our expectation that was described in hypothesis 2c, the result of benchmark 2 does not show a significant difference in productivity between the scenarios with and without rescheduling of the initial routes. As expected in hypotheses 1b, figure 5.4c shows that the waiting times of QCs in both scenarios of benchmark 2, are significantly larger than in benchmark 1, for both loading and discharging QCs. Furthermore, the waiting times of QCs is indeed, as predicted in hypotheses 2b, smaller when the schedules of the Route Scheduling module are recalculated when too much deviation is measured. The prediction of hypothesis 2d that this difference would be larger when the scenario increases, cannot be verified with the results that are displayed in figure 5.4.

If we compare the QC waiting times during loading (figure 5.4c) and discharging (figure 5.4d), it turns out that for the original system, the QC waiting times for increasing scenario sizes grow faster for loading QCs than for discharging QCs. In Benchmark 2 however, the QC waiting times growth for increasing scenario sizes, is stronger for discharging QCs. Informally, we could state that the Scheduling module has more impact on discharging QCs than on loading QCs, which could be explained by the fact that there already occurred delay in the original system for loading QCs, but that in the original system, QCs hardly had to wait for AGVs. For



(a) Quay Crane performance.

(b) Vehicle waiting times.



(c) Quay Crane Waiting times during vessel loading.

(d) Quay Crane Waiting times during vessel discharging.

Figure 5.5: Results of the original system, 'First Come First Served with priorities' and 'Advanced Priorities'.

both loading and unloading, the delay on the roads increases, but for the AGVs that handle discharging QCs, this delay is just too large to be in time at time QC transfer point, which causes relatively a lot of extra waiting for the QCs.

Similar to what we expected in hypothesis 2b, figures 5.4c and 5.4d shows that the second benchmark with recalculation of the route plans outperforms the simulation without intermediate route plan updates. The improvement is however hardly significant and in contrast to the expectation we stated in hypothesis 2d, the obtained results do not show an increase of the effect for larger scenarios.

Although the QC productivity does not show the expected performance increase for `FCFS with updates`, compared to `FCFS no updates` and the QC waiting time only shows a very minor waiting decrease, the AGV waiting percentage is significantly smaller for the scenario with intermediate schedule recalculation, as can be seen in figure 5.4b, which is in line with our expectation of hypothesis 2a. We expect that the effect is more significant for the AGVs waiting times, because AGVs are directly influenced by the new strategies, are only indirect affected.

5.3. Priority scenarios

To investigate the impact of applying a more advanced priority strategy, the first-come first-served strategy is replaced by the more advanced `Advanced Priorities` strategy that takes load, destination and slack into account. This strategy is described in more detail in section 4.3.

If we consider the waiting times of the QCs (see figures 5.5c and 5.5d), then the `Advanced Priorities` already starts to show better performance from four QCs on. This effect is significant for both loading and discharging QCs, and the waiting times are approximately halfway between the original system and the second benchmark. This effect was already expected in hypothesis 4a, and also the expected growing impact for larger scenarios, as stated in hypothesis 4c, was found in these graphs.

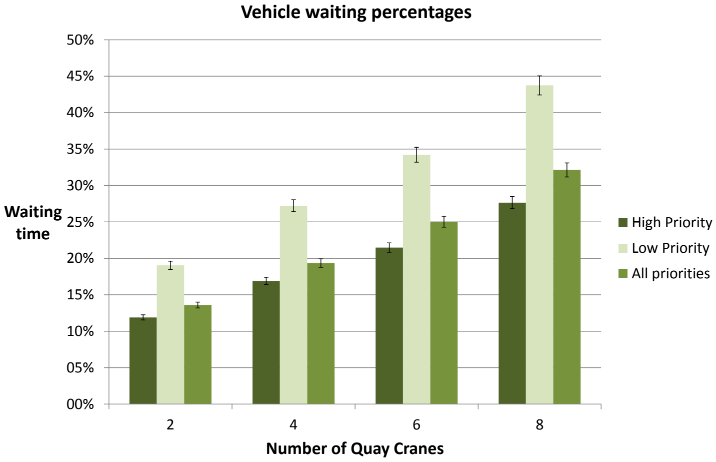


Figure 5.6: Waiting and driving time distribution of high and low prioritized vehicles in the `Advanced Priorities` scenario for different number of Quay Cranes. Note that the high and low priority class together do not cover all vehicle drives.

5.3.1. Quay Crane productivity for advanced priorities

It turns out that up to 4 QCs, the QC productivity in the `Advanced Priorities` scenario does not significantly deviate from the performance in the `FCFS with updates` scenario, as shown in figure 5.5a. For 8 QCs however, the `Advanced Priorities` strategy shows a significantly better performance than the route scheduling strategy with the first-come first-served strategy, which corresponds to our expectation of hypothesis 4b that the benefit of `Advanced Priorities` would increase when the traffic density increases. The difference compared to the `FCFS with updates` strategy, 7.5 %, is more than our expectation in hypotheses 4b of a few percent.

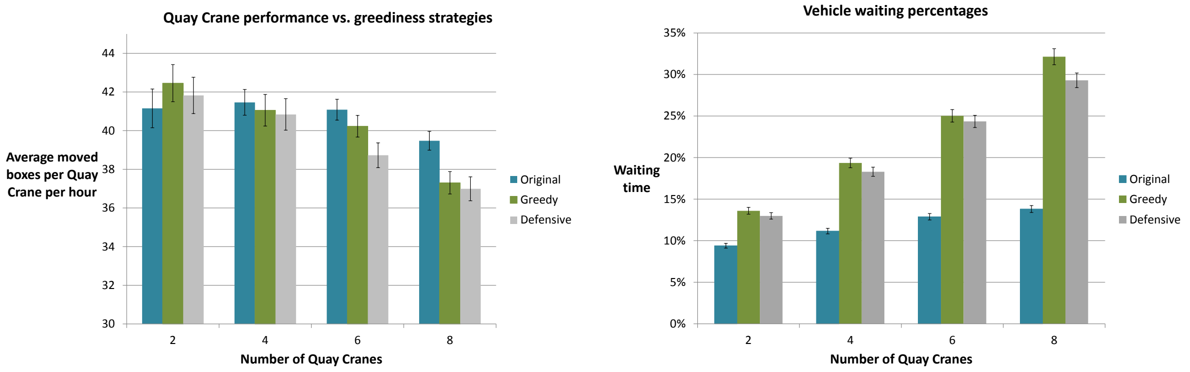
Despite of the large performance increase compared to the `FCFS with updates` strategy however, the number of performed QC moves, is still 5.5 % smaller than in the original system. Therefore, we cannot guarantee that it is possible to improve the Original system's performance, when a more accurate route scheduling module would be implemented.

5.3.2. Vehicle waiting for advanced priorities

Figure 5.5b shows that the expected increase of total AGV waiting time for the `Advanced Priorities` scenario, as stated in hypothesis 3a, was not found in our simulation results. This means that the benefit for the high prioritized vehicles has more impact on the total waiting time than the extra waiting time for the lower prioritized vehicles. A possible explanation is that the high prioritized AGVs have higher average speeds and hence, occupy the roads shorter, leading to a smaller traffic density, which improves the total traffic flow.

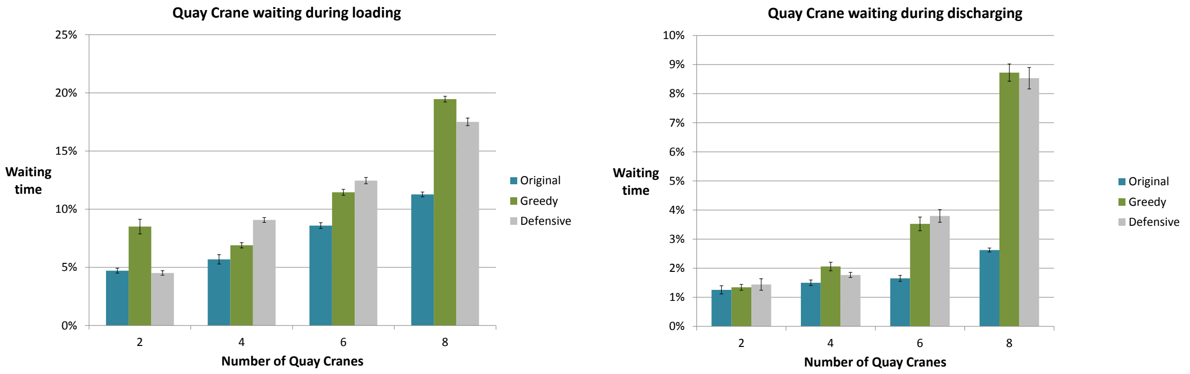
To visualize the effect of using more advanced priority rules, the difference in waiting times versus the priority class (high-low) is drawn for different numbers of QCs in figure 5.6. An AGV is waiting if it stands still and it is waiting until it receives access to its next segment. The status driving is used for the time an AGV is actually driving, either accelerating, decelerating or at constant speed. The time an AGV is (waiting to be) handled by a crane, or waiting for a new route, is ignored. AGVs are said to have a high priority when the available slack is negative and the AGV is moving (indirectly) to a QC. An AGV is classified as low prioritized if is driving towards the Battery Exchange Station or driving loaded to the stack. Note that low and high priority are disjunct, but do not cover all AGV drives. Furthermore, the priority of an AGV is a temporarily status, so AGVs can change over time between high and low priority.

From figure 5.6, it is clear that the `Advanced Priorities` has a significant influence on the behavior of AGVs, even for the small scenarios with 2 and 4 QCs, which confirms our expectations of hypothesis 3b and 3c that stated that high prioritized AGVs would have less AGV waiting times than average and low prioritized AGVs more. The figure shows that the further that the extra waiting time for the low prioritized vehicles is larger than the reduction of waiting time for the high prioritized vehicles. For all types of priorities however, the nature the waiting time growth is equal. All types grow approximately linearly when the scenario size increases, and the ratios between the different types stay constantly for increasing scenario sizes.



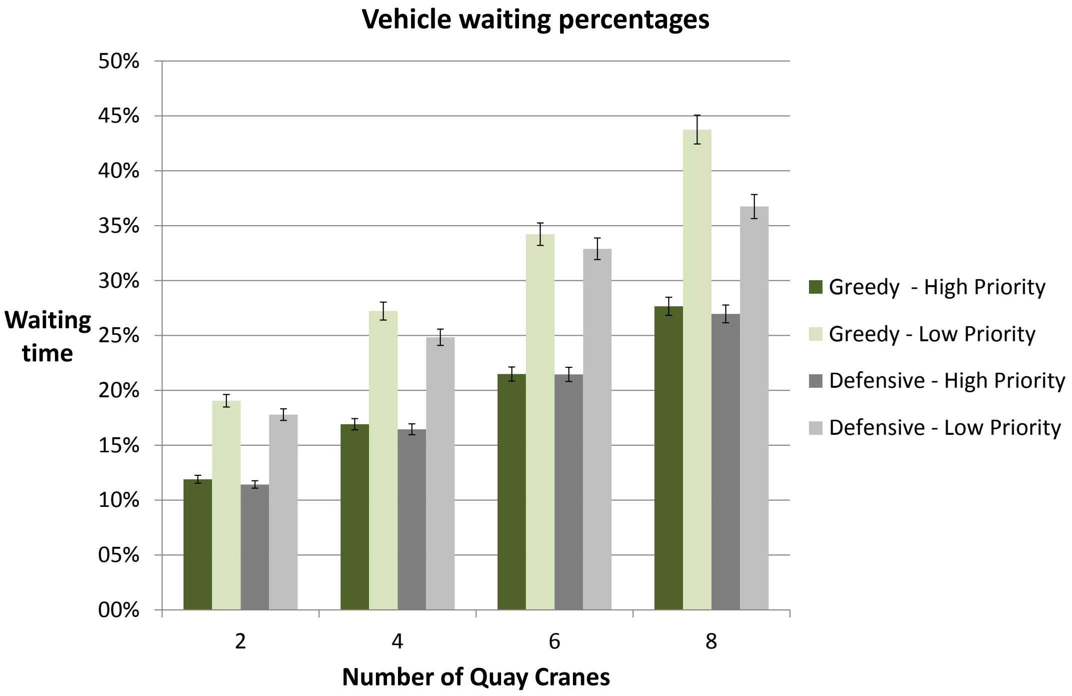
(a) Quay Crane performance.

(b) Vehicle waiting times.



(c) Quay Crane Waiting times during vessel loading.

(d) Quay Crane Waiting times during vessel discharging.



(e) Waiting of high and low prioritized vehicles.

Figure 5.7: Key performance indicators for two different greediness strategies.

5.4. Greediness scenarios

To investigate the impact of the scheduler's greediness, the current best performing strategy, which is the relatively greedy `Advanced Priorities` strategy, is adapted to a more defensive strategy, `Defensive`. To measure the impact of greediness, these two strategies are compared on a number of Key Performance Indicators.

Contrary to what we expected in hypothesis 5c, the QCs do not show a significant performance increase when the `Defensive` strategy is applied, except for the situation with 6 QCs (figure 5.7a). It is not clear why this improvement is only significant for the the scenario with 6 QCs and 30 AGVs, but we expect that the smaller scenarios do not offer enough situations where the defensive strategy can prove its value, and for the largest scenario, too much vehicles are delayed, leading to high priorities, which decreases the impact of defensive driving, because the time score's range, as described in section 4.4 is relatively small compared to the priority range.

The expected decrease of QC waiting times as described in hypothesis 5b, cannot be confirmed by the obtained results that are shown in figures 5.7c and 5.7d. To find the impact of the `Defensive` strategy, we should therefore analyze the most direct result of this strategy, the AGV waiting times, which are shown in figure 5.7b. For the AGV waiting times, the predictions of hypothesis 5a that `Defensive` will lead to less AGV waiting time, shows to be right, as can be seen in figure 5.7b. The difference is minor, and for 6 and less QCs, it is not significant, but for 8 QCs, the AGV waiting time does show a significant decrease. By investigating the AGV waiting times for high and low prioritized vehicles, which is done in figure 5.7e, it can be seen that especially the low prioritized vehicles show lower waiting times. This explains why the decreased waiting times hardly influence the QC behavior.

5.5. Discussion

To keep the simulations in this research as realistic as possible, both 20 ft. and 40 ft. containers are used. Quay Cranes and AGVs however, can handle two AGVs simultaneously, which increases the number of boxes that can be handled per hour. Pairs of containers that will be loaded by a single QC move, are not necessarily stored in the same stack module, which causes the serving AGV to drive via two stack modules. Using only 40 ft. containers in the simulation, is therefore undesirable as this changes the driving dynamics by removing a significant part of the traffic on the lower two highways, and it would remove a part of the drives that shows relatively often delays. Conform the common practice in the world of container terminals, the QC productivity is expressed in terms of moved boxes per hour rather than in QC moves per hour. The type of containers are stochastically determined per vessel bay, and by running the simulations long enough, the fraction of 20 ft. containers will converge to a fraction that is determined by the responsible parameter.

The number of simulation hours per experiment is based on two heuristics that are used for real-world terminal simulations. At least 100 hours of simulation should be executed for every scenario and the half width of the 95 % confidence interval of the average Quay Crane productivity should be at most 1.00 move. To satisfy these two rules, 100 hours of simulation were sufficient for the scenarios with at least 4 QCs, but for the small-sized scenarios with 2 QCs, almost 200 hours of simulation were necessary to sufficiently reduce the confidence interval. The used heuristics that are sufficient for scenarios with realistic properties, seem to be insufficient for the artificial situation with 2 AGVs, as for certain situation runs, both the QC waiting times and the QC productivity are significantly higher than in the larger scenarios. The number of Quay Crane hours however, is for these simulations significantly smaller than for the smallest scenarios that are executed for real-world terminals, which causes too large deviations in the performance to be fully reliable. We will therefore not draw conclusions solely based on the results of these small scenarios.

6

Conclusions

Based on the results that are described in chapter 5, this chapter describes the answers on the research questions in section 6.1. Next we state our recommendations for further research and development in an industrial setting in section 6.2, and finally we describe a number of ideas for future work for scientific purposes in section 6.3.

6.1. Answering the Research Questions

The main research question was:

Main research question. *To what extent can the QCs' waiting times be decreased by improving the AGVs' driving behavior?*

Before formulating an answer to this research question, the researched sub questions are answered.

Sub research question 1. *How much time are AGVs waiting during their drive and how much time are QCs waiting for an AGV?*

Answer : In the original system, QCs have a productivity percentage of 92 % for scenarios with a realistic equipment use, and a maximum measured productivity of 96 % for very simple scenarios with only a quarter of the initial AGVs and QCs. Furthermore, QCs are spending approximately 1 % of their time on sideways moves and the rest of the remaining part of their time is spend on waiting for AGVs that are too late. Most of the QCs' waiting times, 80 to 85 %, is caused by loaded AGVs that are too late.

Sub research question 2. *To what extent can the QCs' waiting times be decreased by using AGV priorities?*

Answer : For the most realistic scenario, with 8 QCs, the waiting times of QCs can be decreased from 20 % to 14 % compared to a situation where route plan scheduling is applied based on a first-come first-served strategy. However, this is still higher than the 7 % waiting time of the original system. For smaller scenarios with less AGVs and QCs, the same effect is found, but with smaller values.

Sub research question 3. *What is the effect of AGVs' greediness on the QCs' waiting times?*

Answer : Although a small decrease of AGV waiting time is found for the strategy with less greedy AGV performance, this does not lead to significant decrease of waiting times of the QCs.
Answer to the main research question : Improvement of the AGVs' driving behavior by applying the Advanced Priority strategy, can decrease the average QCs' waiting percentage of an AGV system with route schedule planning from 20 % to 14 %. This performance can be reached by applying either a greedy or a more defensive AGV driving strategy. This is however still larger than the 7 % waiting time that occurs in an AGV system with a standard claiming strategy.

6.2. Recommendations

As the performance of our implementation of route plan scheduling made it impossible to obtain a clear view of the potential of using advanced priorities, our main recommendation is to repeat the simulations of this thesis with a more accurate route planning module.

To make the route plannings more accurate, a number of improvements could be made. Firstly, the assumed speed of the AGV in a certain segment, should be made more accurate. This could be done by using the actual measured average driving speed of AGVs instead of a fixed speed that is added as a parameter to the model. Another improvement of the driving time calculations that we recommend, is to use a longer occupation time of segments that are entered after a waiting period as the AGV has to accelerate in this segment and will therefore occupy the segment longer. Another approach to make the planning more accurate is to use a smaller recalculation threshold to make the used predictions more precise.

These precision improvements however, will decrease the system's execution speed, which could lead to violation of the scalability requirements. Therefore, these improvements should be combined with modifications of the module to increase the calculation speeds. A large speedup could be reached by not calculating the priorities of the full route of an AGV in advance, but only to a certain point in the close future, for example at most 30 seconds ahead.

In this thesis, only one specific perpendicular terminal layout is investigated. It would be interesting to see how the described strategies perform in other perpendicular layout and in parallel layouts where the number of road intersections is larger.

6.3. Future work

A first recommendation for future work, is to investigate how a reduction of the inter-vehicle distance influences the Quay Crane productivity, which could be combined with route plan scheduling, but can be investigated apart as well. We suggest however to combine these strategies, as the potential benefit could be improved by determining a row sequence in advance. Our analysis of this concept is elaborated in more detail in section 4.5.

In this thesis we only investigated one formula to calculate AGV priorities, which was based on heuristics. It would be an interesting direction for further research to investigate what the optimal way is to calculate priorities for AGVs in order to minimize the QC waiting times. We expect that the major challenge for this research will be to find a way to evaluate possible computation methods a few orders of magnitude faster than the few days it takes to run 100 hours of simulation.

Another idea for further research is to base the priority not only on the properties of the two vehicles that want to pass the crossing at the same time, but also on the vehicles that drive directly after these vehicles. This approach will decrease the waiting time for high prioritized AGVs that drive directly behind a low prioritized AGV. The information about driving order, which is maintained in the scheduling module, can be used to determine what vehicles should be considered as successors. Key decisions that should be taken are how many successors are taken into account and how the successor score is exactly determined.

Bibliography

[1] Lift agv. http://www.konecranes.com/equipment/container-handling-equipment/automated-guided-vehicles/lift-agv. Accessed: 2017-07-31.

[2] Oocl reaches milestone with the christening of the oocl hong kong. http://www.oocl.com/eng/pressandmedia/pressreleases/2017/Pages/12may17.aspx. Accessed: 2017-06-19.

[3] Panagiotis Angeloudis and Michael GH Bell. An uncertainty-aware agv assignment algorithm for automated container terminals. *Transportation Research Part E: Logistics and Transportation Review*, 46(3):354–366, 2010.

[4] Christine G Co and Jose Mario Azaña Tanchoco. A review of research on agvs vehicle management. *Engineering Costs and Production Economics*, 21(1):35–42, 1991.

[5] Mark B Duinkerken and Gabriel Lodewijks. Routing of agvs on automated container terminals. In *Computer Supported Cooperative Work in Design (CSCWD), 2015 IEEE 19th International Conference on*, pages 401–406. IEEE, 2015.

[6] Burak Eksioglu, Arif Volkan Vural, and Arnold Reisman. The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57(4):1472–1483, 2009.

[7] Michael Erdmann and Tomas Lozano-Perez. On multiple moving objects. In *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, volume 3, pages 1419–1424. IEEE, 1986.

[8] Hamed Fazlollahtabar and Mohammad Saidi-Mehrabad. Methodologies to optimize automated guided vehicle scheduling and routing problems: a review study. *Journal of Intelligent & Robotic Systems*, 77(3-4):525, 2015.

[9] JOC Group. Berth productivity, the trends, outlook and market forces impacting ship turnaround times, 2014.

[10] Luka Kalinovcic, Tamara Petrovic, Stjepan Bogdan, and Vedran Bobanac. Modified banker's algorithm for scheduling in multi-agv systems. In *Automation Science and Engineering (CASE), 2011 IEEE Conference on*, pages 351–356. IEEE, 2011.

[11] Pooja Kavathekar and YangQuan Chen. Vehicle platooning: A brief survey and categorization. In *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 829–845. American Society of Mechanical Engineers, 2011.

[12] Nirup N Krishnamurthy, Rajan Batta, and Mark H Karwan. Developing conflict-free routes for automated guided vehicles. *Operations Research*, 41(6):1077–1090, 1993.

[13] JH Lee, BH Lee, and MH Choi. A routing table approach for minimum time traffic control of multiple agv systems. In *Industrial Electronics, Control and Instrumentation, 1997. IECON 97. 23rd International Conference on*, volume 3, pages 1222–1227. IEEE, 1997.

[14] Rolf H Möhring, Ekkehard Köhler, Ewgenij Gawrilow, and Björn Stenzel. Conflict-free real-time agv routing. In *Operations Research Proceedings 2004*, pages 18–24. Springer, 2005.

[15] Christopher Oboth, Rajan Batta, and Mark Karwan. Dynamic conflict-free routing of automated guided vehicles. *International Journal of Production Research*, 37(9):2003–2030, 1999.

[16] Dirk Steenken, Stefan Voß, and Robert Stahlbock. Container terminal operation and operations research-a classification and literature review. *OR spectrum*, 26(1):3–49, 2004.

[17] Fataneh Taghaboni-Dutta and Jose Mario Azana Tanchoco. Comparison of dynamic routeing techniques for automated guided vehicle system. *International Journal of Production Research*, 33(10):2653–2669, 1995.

[18] Koen H van Dam, Jaap A Ottjes, Gabriël Lodewijks, ZV Lukszo, and René W Wagenaar. Intelligent infrastructures: distributed intelligence in transport system control-an illustrative example. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 5, pages 4650–4654. IEEE, 2004.

[19] R. van den Hof. Improving the traffic performance of automated guided vehicles at automated container terminals. Master's thesis, Rijksuniversiteit Groningen, The Netherlands, 2008.

[20] A. Vos. Scheduling automated guided vehicles in container terminals using max-plus-linear systems. Master's thesis, Delft University of Technology, The Netherlands, 2015.