Robust Integration of Acceleration and Deceleration Processes into the Time Window Routing Method

Thomas Lienert, M.Sc., Technical University of Munich, Chair for Materials Handling, Material Flow, Logistics, Germany

Dipl.-Ing. Florian Wenzler, Technical University of Munich, Chair for Materials Handling, Material Flow, Logistics, Germany

Prof. Dr.-Ing. Johannes Fottner, Technical University of Munich, Chair for Materials Handling, Material Flow, Logistics, Germany

1. Introduction

An automated guided vehicle system (AGVS) is a driverless transport system for moving materials horizontally (Vis 2006). It consists essentially of at least one automated guided vehicle (AGV), a guidance control system, devices for localization, and equipment for data transmission (VDI 2510). AGVSs are commonly used in manufacturing plants, warehouses, distribution centers, and transshipment terminals (Le-Anh & De Koster 2006).

Robotic mobile fulfillment systems (RMFSs) are a more recent AGVS application. RMFSs are a new type of automated storage and retrieval system used for part-to-picker order-picking systems (Lamballais et al. 2017). The products are stored on racks, which are arranged in storage aisles on the ground. The vehicles are considered to be mobile robots in this context. They move throughout a rectangular grid of paths within the storage area and can travel along the storage aisles as well as underneath the racks if the vehicles are empty. Once an order arrives and is assigned to a picking station, a vehicle moves under the rack containing the required item, lifts the rack, and brings it to the designated picking station, where the item is picked. A vehicle subsequently brings the rack back to an empty storage location.

Shuttle systems with tier-to-tier and aisle-to-aisle shuttles are another example of an automated vehicle based storage and retrieval system. In these systems, every shuttle can reach every storage location as the vehicles use lifts to change storage tier and cross-aisles to change storage aisle (Lienert & Fottner 2017a).

The main benefits relative to common stacker-crane-based storage and retrieval systems are simple scalability and good redundancy. The whole system can be run with a single vehicle. If needed, more vehicles can be added to achieve greater throughput. Should a single vehicle fail, the remaining vehicles continue to fulfill the storage and retrieval request and system's throughput is only slightly affected.

However, one such system requires a more complex control strategy to run robustly and efficiently. Dispatching and routing are the main issues that the control has to address: where and when should a vehicle travel? And which path should be taken to reach a designated position? Routing must not only choose the routes, but also take into account deadlocks-situations in which one or more vehicles are permanently blocked.

2. Time Window Routing Method

Three generic approaches to handle deadlocks are distinguishable: static deadlock prevention, detection and recovery, and deadlock avoidance, which is a dynamic approach that generally allows the greatest resource utilization (Liu & Hung 2001). One possibility for avoiding deadlocks is the time window routing method, which was introduced by Kim and Tanchoco for the conflict-free routing of automated guided vehicles in a bidirectional network (Kim & Tanchoco 1991). Ter Mors presented a generic model for routing agents through an infrastructure graph with better worst-case performance. A resource graph, the nodes of which correspond to the nodes and edges of the infrastructure graph, is generated to calculate the route. The edges of the resource graph can be interpreted as a successor relation (ter Mors 2009). Lienert and Fottner extended his approach by integrating crossings where vehicles can turn 90 degrees, and by enabling a sequencing procedure that occurs during routing. They integrated the algorithm into a simulation-based tool for analyzing AGVs' performance in the context of automated storage and retrieval systems (Lienert & Fottner 2017b).

The time window routing method is used to obtain deadlock-free routes for vehicles moving through infrastructures modelled by a graph. For each node, the algorithm maintains a list of free time windows through which vehicles can be routed (figure 1).

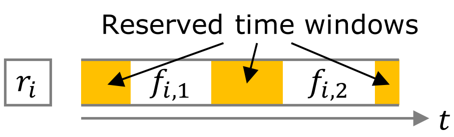

Figure 1: Free time windows, $f_{i,j}$, on a node, r_i

Starting with the initial time window on the start node, each iteration of the algorithm investigates the reachability of all free time windows on all neighboring nodes (see figure 2). Some conditions, such as sufficient overlapping, must be met for a free time window to be reachable.

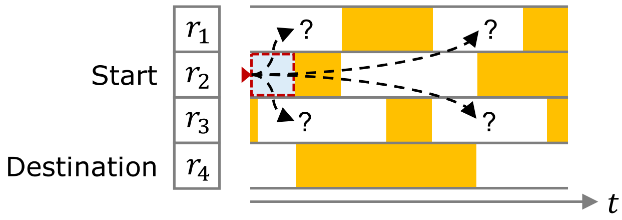

Figure 2: Which free time windows are reachable from the current free time window on node r_2?

The routing procedure results in the fastest path from the start node to the destination node through the free time windows (figure 3).

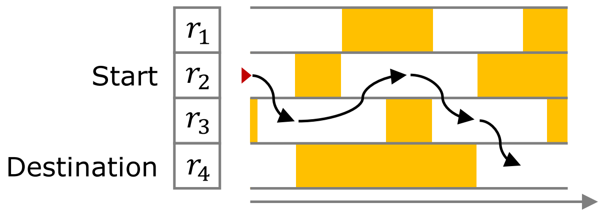

Figure 3: Fastest path from the start to the destination

3. Problem Statement

As Maza and Castagna showed, the absence of deadlocks is guaranteed as long as the nodes' crossing order is preserved, even if some vehicles are late and do not match their reserved time windows (Maza & Castagna 2005). Neglecting acceleration and deceleration processes is furthermore common in the context of routing (ter Mors 2009). Hence in a real world application, moving vehicles won't meet the reserved time windows exactly.

As a consequence, the vehicles are navigated only according to the sequence of reserved time windows on the nodes. Lienert and Fottner therefore introduced the concept of claiming nodes: A vehicle is only allowed to enter a node that it has previously claimed. A vehicle can claim a node only if the earliest reserved time window on that node was reserved by that vehicle. The nodes that a vehicle could traverse during the next movement are identified whenever the vehicle starts moving. (Lienert & Fottner 2017b). We call a segment the set of nodes that are claimed for the next movement. A segment's length is bounded by a node requiring a 90 degree turn, a maximum segment length, or a node where the vehicle hasn't reserved the earliest time window.

Figure 4 provides an example of this procedure. Since the vehicle reserved the earliest time windows on nodes $r_2, \ldots, r_7$, it's allowed to claim these nodes and to start the linear movement.

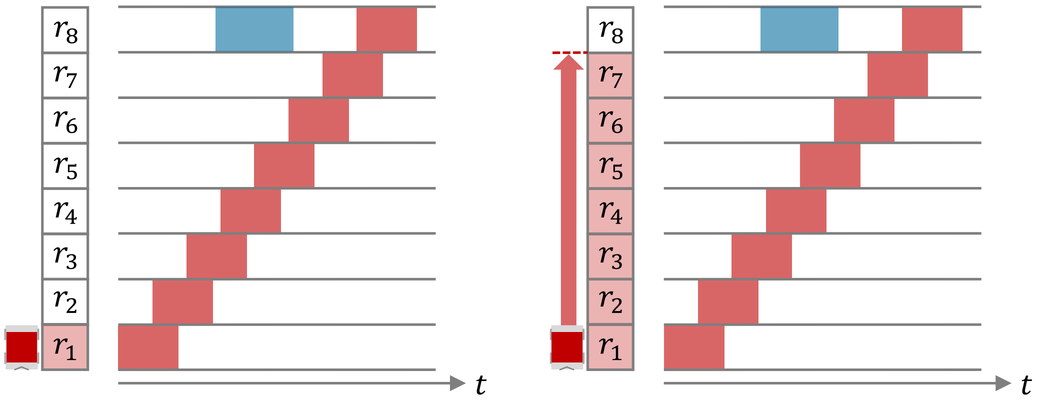

Figure 4: Claiming the next segment's nodes

Now we consider the following situation: Another vehicle's routing takes place during the movement and an earliest time window is reserved on node r_7, which is already claimed (figure 5).

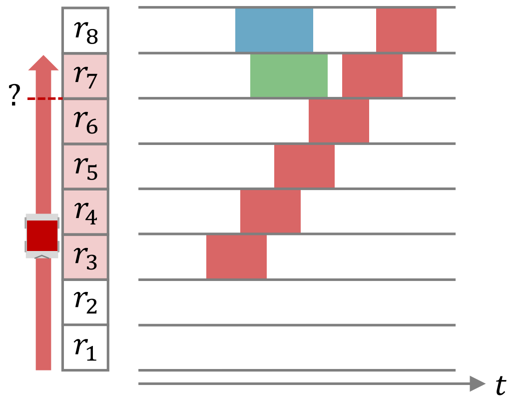

Figure 5: Another vehicle reserves an earliest time window on a node that is already claimed

The nodes' crossing order needs to be preserved to preclude deadlocks. If all of the vehicles were to move perfectly over the infrastructure without any delays, then routing would ensure the second vehicle's having left node r_7 before the currently moving vehicle enters it. But such behavior cannot be guaranteed in reality. Therefore, the currently moving vehicle has to stop on node r_6 to prevent deadlocks and collisions. However, this can't be guaranteed, be it due to the breaking distance or to communication delays between the overall control infrastructure and the vehicles.

The described situation can happen, because various processes can simultaneously access the data structure of the time windows. Routes are planned sequentially for the vehicles, but execution of movements takes place simultaneously. Figure 6 gives information about accesses to the data structure of the time windows.

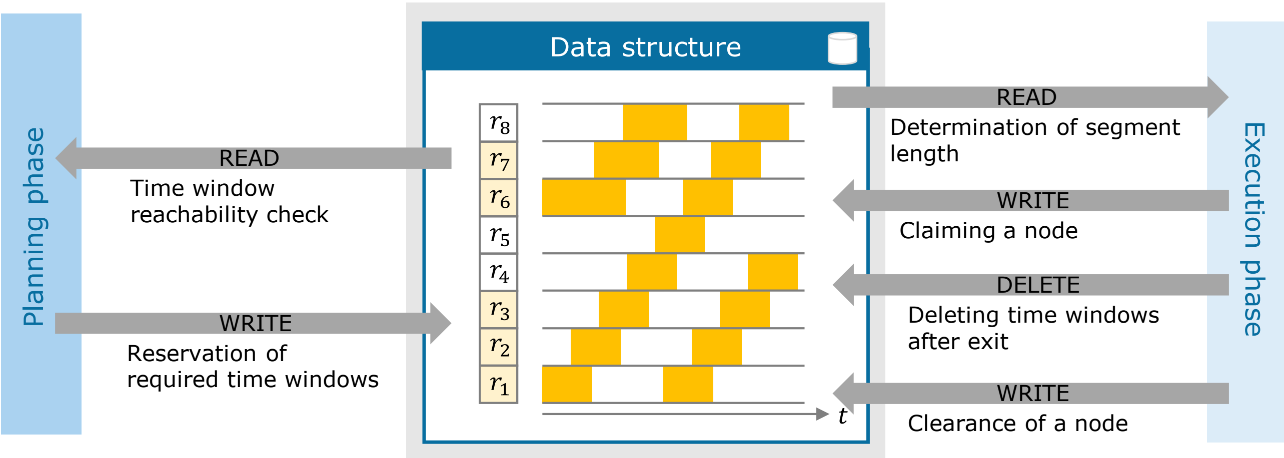

Figure 6: Read and write accesses to the data structure of the time windows

It can be seen that write accesses take place during both the planning phase (the routing process) and the execution phase (claiming nodes and executing movements). To solve the problem described above, we suggest that the planning and execution phases be strictly separated such that the data structure of the time windows is not write accessed while a calculated route is being executed.

4. Solution Approach

The idea behind this solution approach is to build the linear movement segments during the planning phase. The question about how far a vehicle is allowed to travel during a movement is not answered during the execution, but during the planning phase. Whenever a vehicle starts moving over the next n nodes, it will not only have reserved the earliest time windows on these n nodes, but the time windows on all these nodes will begin with the movement. As a result, the reservations will have a triangular shape as shown in figure 7, assuming that every single node can be released as soon as the vehicle has completely left that node. This approach not only leads to robust execution of calculated routes, but also allows routing to take acceleration and deceleration processes into account since the movements' lengths are already known during the planning phase.

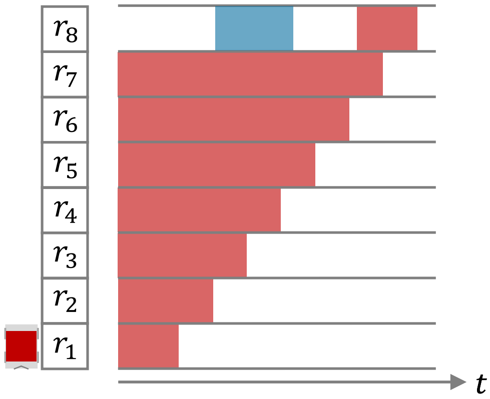

Figure 7: Reservation of a segment

The flow chart in figure 9 describes the underlying routing procedure. Lienert and Fottner provide a detailed description of this algorithm as well of the layout modeling (Lienert & Fottner 2017b). We will subsequently focus only on the modification concerning the extension of a movement during the planning phase. How far a movement can be extended in a straight line through several free time windows is examined instead of checking the reachability of every free time window on every neighboring node (figure 8).

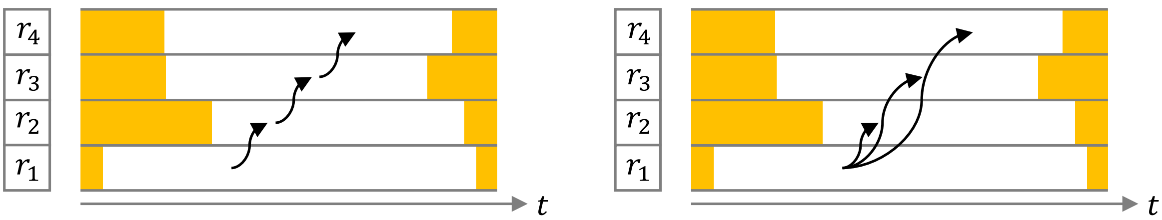

Figure 8: Routing free time window by free time window (left) and with movements through several free time windows (right)

The variables below are needed to describe the modified algorithm. We refer, as already mentioned, to (Lienert & Fottner 2017b) for further description.

r_i:	Node within the graph representing the infrastructure
Adj_{r_i}:	Adjacency list of node r_i
A_{r_i}:	Set of orientations along which a node can be traversed. We assume only vertical and horizontal movements as the vehicles are moving over an orthogonal grid of aisles and cross-aisles. $A_{r_i} \in \{\{X\},\{Y\},\{\{X,Y\}\}$.
$f_{i,l}$:	l-th free time window on node r_i
F_i:	Set of all free time windows on node r_i; $F_i = \{f_{i,1}, f_{i,2}, \dots, f_{i,l}, \dots\}$.
$\mathrm{Start}_{f_{i,l}}$:	Start of free time window $f_{i,l}$ on node r_i
$\mathrm{End}_{f_{i,l}}$:	End of free time window $f_{i,l}$ on node r_i
s_{r_i,r_j}:	Euclidean distance between nodes r_i and r_j
t_R:	Time needed to turn 90 degrees on a crossing node to change the orientation of the vehicle
$T_{\mathrm{De},f_{i,l}}$:	Departure time from time window $f_{i,l}$ on node r_i
$T_{\mathrm{Ex},f_{i,l}}$:	Exit time from time window $f_{i,l}$ on node r_i
$T_{\mathrm{Ar},f_{i,l}}$:	Arrival time in free time window $f_{i,l}$ on node r_i
s^A_{Ex,r_i}:	Exit distance that a vehicle has to cover from the centered arrival position in orientation A until the vehicle leaves the node completely
$t(s_{r_i,r_j})$:	Driving time between the nodes r_i and r_j
$t(s,l)$:	Driving time to cover partial distance s while performing an overall movement of length l (See appendix for further explanation.)

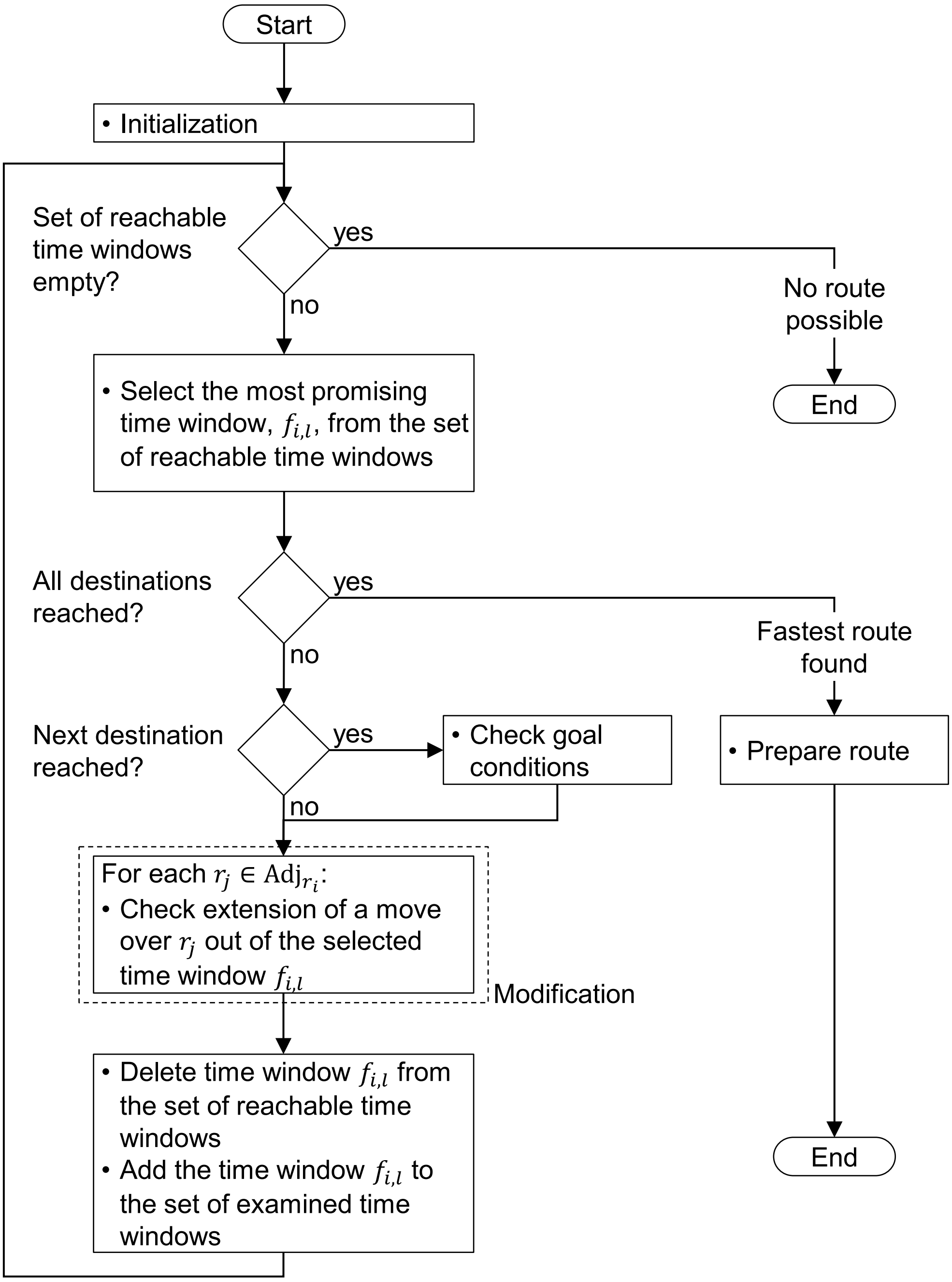

Figure 9: Overall routing procedure

The flow chart in figure 10 describes the extension of a movement. The selected time window is first added to the initial segment. Segment A's orientation has to be determined and the earliest possible departure time from time window $f_{i,l}$ calculated:

$$T_{\mathrm{De},f_{i,l}} = \begin{cases} max\{T_{\mathrm{Ar},f_{i,l}}, \mathrm{Start}_{j,m}\} & ,without\ 90°\ turn \\ max\{T_{\mathrm{Ar},f_{i,l}} + t_R, \mathrm{Start}_{j,m}\} & ,with\ 90°\ turn \end{cases} \quad (1)$$

Departure from the time window cannot start before arrival in time window $f_{i,l}$ has taken place and before free time window $f_{j,m}$ has started. Depending on current orientation, a 90° turn might be necessary to reach node r_j.

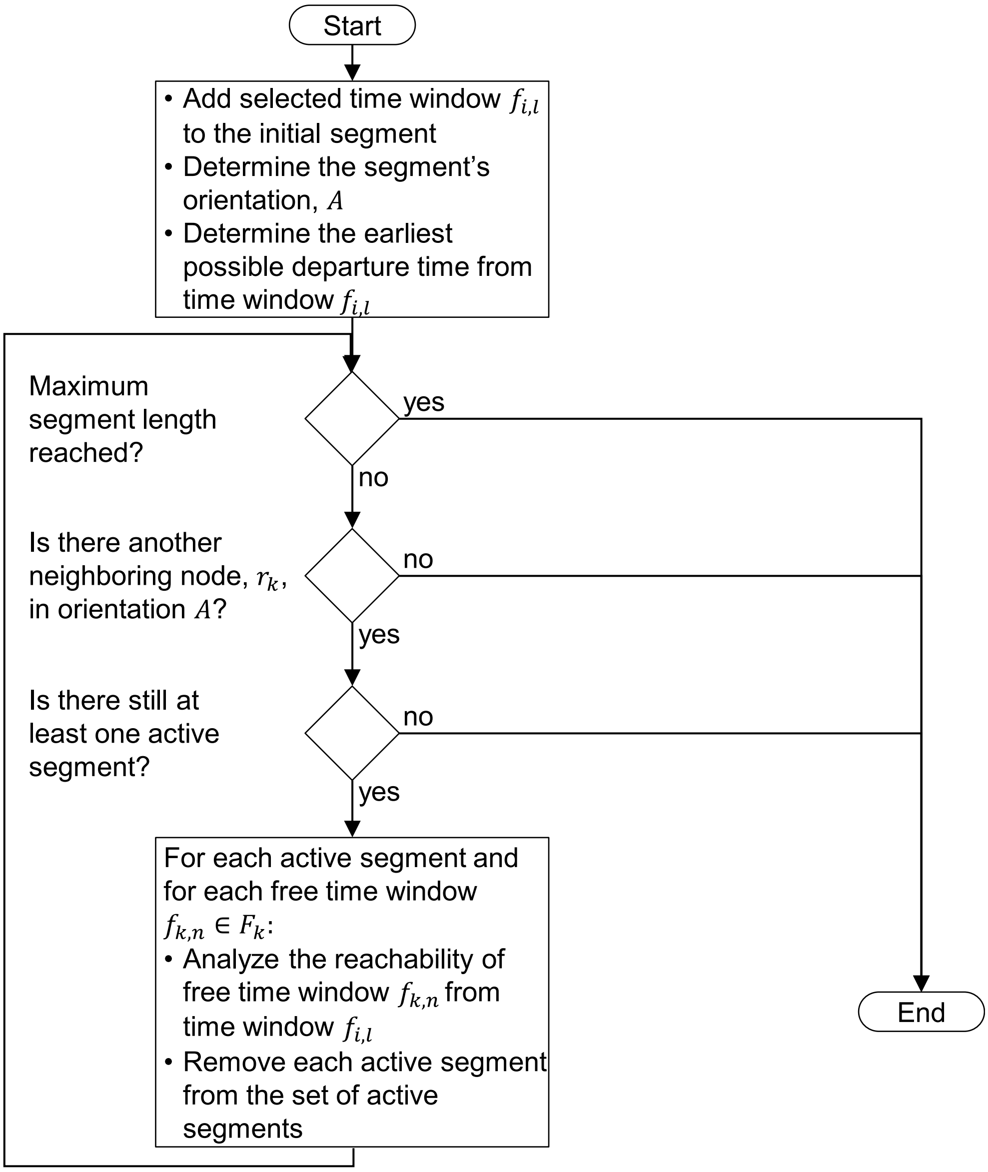

Figure 10: Extension of a movement

If the maximum segment length – measured by the number of nodes or travel distance – has not yet been reached, and there is another neighboring node in the moving orientation, r_k, so that the segment can be extended, and there is at least a single active segment, then the reachability of all free time windows on node r_k will be analyzed for each active segment.

In the example shown in figure 11, the segment currently consists of three free time windows: $f_{1,1}, f_{2,1}$, and $f_{3,1}$. Assuming that the maximum segment length is at least three nodes, the reachability of free time window $f_{4,1}$ on node r_4, which is the next neighboring node along the segment's orientation, can be analyzed.

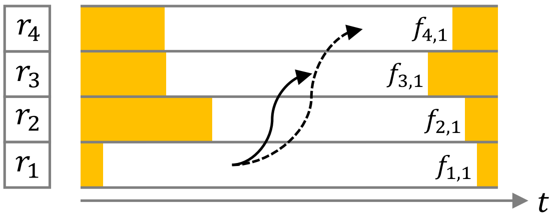

Figure 11: The segment with movement from node r_1 toward node r_3 comprises free time windows $f_{1,1}, f_{2,1}$, and $f_{3,1}$. The reachability of free time window $f_{4,1}$ will be analyzed during the next iteration.

The flow chart in figure 12 describes the procedure for checking whether free time window $f_{k,n}$ on node r_k is reachable from time window $f_{i,l}$.

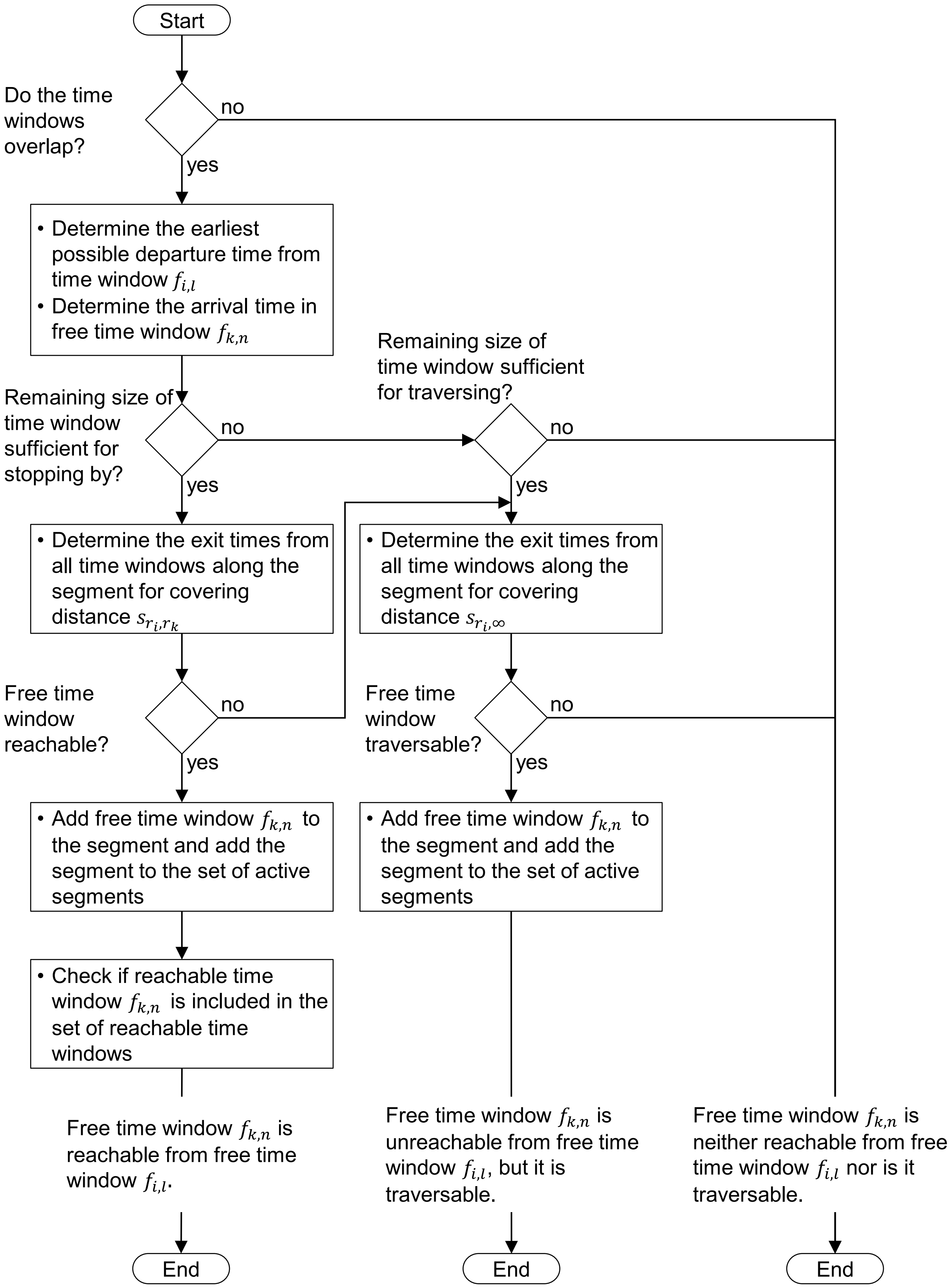

Figure 12: Reachability check of free time window $f_{k,n}$

If the departure time of segment $T_{\mathrm{De}_{i,l}}$ falls into free time window $f_{k,n}$ and both free time windows overlap each other,

$$T_{\mathrm{De}_{i,l}} < \mathrm{End}_{k,n} \text{ and } \mathrm{Start}_{k,n} < \mathrm{End}_{i,l}, \tag{2}$$

then the earliest possible departure time,

$$T_{\mathrm{De},f_{i,l}} = \max\{T_{\mathrm{De},f_{i,l}}, \mathrm{Start}_{k,n}\}, \tag{3}$$

and the arrival time in free time window $f_{k,n}$,

$$T_{\mathrm{Ar},f_{k,n}} = T_{\mathrm{De},f_{i,l}} + t(s_{r_i,r_k}), \tag{4}$$

can be determined. Next, the question about whether enough of the free time window remains so that the vehicle can stop at that node and leave it afterwards before the free time window ends,

$$T_{\mathrm{Ar},f_{k,n}} + t\left(s_{\mathrm{Ex},r_k}^{A_{f_{j,m}}}, s_{r_k,\infty}\right) \leq \mathrm{End}_{k,n}, \tag{5}$$

has to be answered. If enough time remains, then all of the exit times, T_{Ex}, from the segment's the free time windows are determined (see appendix for further explications). If none of the preceding free time windows is violated, the free time window is reachable and the segment can be extended (figure 13).

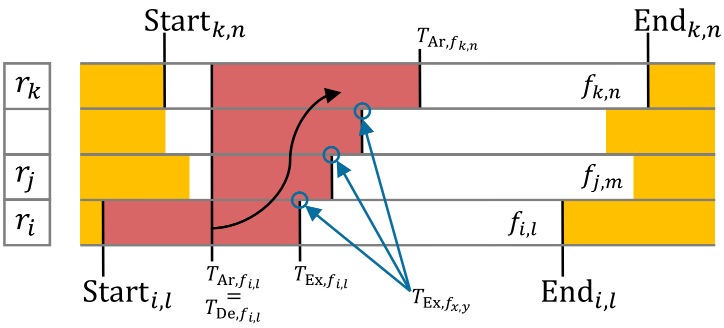

Figure 13: Free time window $f_{k,n}$ is reachable from time window $f_{i,l}$.

The free time window might be included in the set of reachable time windows (see Lienert & Fottner 2017b). Whether the free time window can be traversed will be examined if any of the preceding free time windows is violated or the remaining duration of the free time window is insufficient for stopping by. On the one hand, free time window $f_{k,n}$ must be vacated completely before it ends:

$$T_{\mathrm{De},f_{i,l}} + t\left(s_{r_i,r_k} + s_{\mathrm{Ex},r_k}^{A_{f_{j,m}}}, s_{r_i,\infty}\right) \leq \mathrm{End}_{k,n}. \tag{6}$$

On the other hand, once again, no preceding time window may be violated. The exit times will therefore be calculated assuming an infinitely long movement. If the free time window is traversable, then the segment will be extended, but the check for including the free time window in the set of reachable time windows is not done, as the free time-window is unreachable.

Figure 14 shows two examples of a reachable free time window. On the left-hand side, the segment can be extended easily. On the right-hand side, a waiting time is required on node r_i as the beginning of the free time window, $\mathrm{Start}_{k,n}$, is later than the arrival time, $T_{\mathrm{Ar},f_{i,l}}$, in time window $f_{i,l}$.

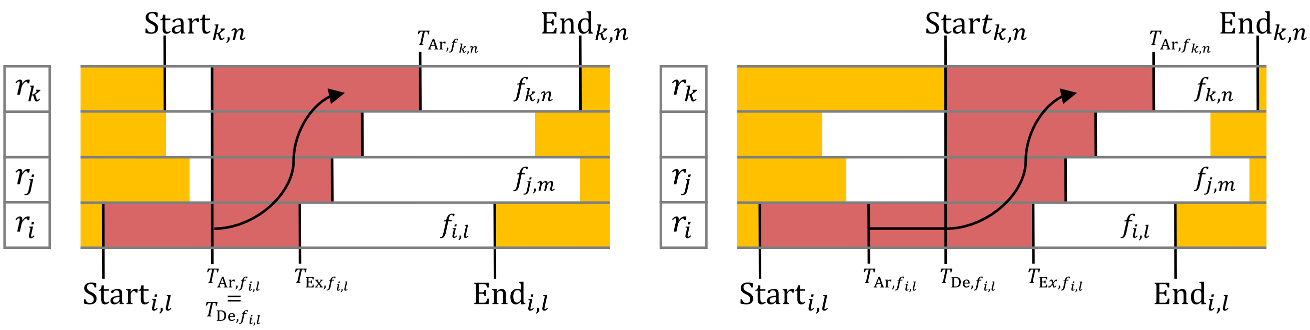

Figure 14: Reachability check of free time window $f_{k,n}$ from time window $f_{i,l}$. On the left-hand side, the free time window is reachable without any delay. On the right-hand side however, a waiting time ($T_{De,f_{i,l}} - T_{Ar,f_{i,l}}$) on node r_i is required as the free time window starts later than arrival in the current time window.

Because several free time windows on a single node might be reachable, several active segments, which need to be examined within the next iterations, can exist (see figure 15).

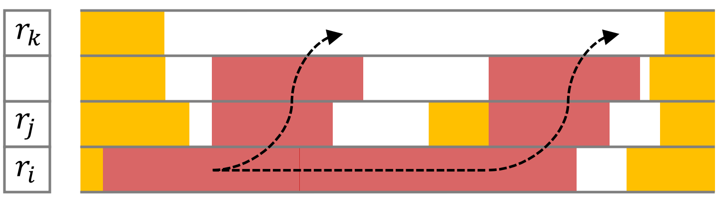

Figure 15: Two free time windows on node r_j are reachable. As a consequence, there are two active segments and a segment's extension has to be checked twice for every free time window on node r_k.

Applying the described reservation mechanisms obviates the concept of claiming nodes. Whenever a vehicle finishes a segment, two conditions must be fulfilled for traveling to the next segment. First, all the reserved time windows involved must start at least by the time the movement begins. Second, all the reserved time windows involved must be the earliest ones on their nodes. The vehicle starts moving when these conditions are met. Otherwise it will register as a waiting vehicle and wait for a notification that will be given as soon as the previous vehicle relinquishes its reserved time window on the respective node.

Figure 16 shows the read and write accesses to the data structure of time windows with modified routing procedure.

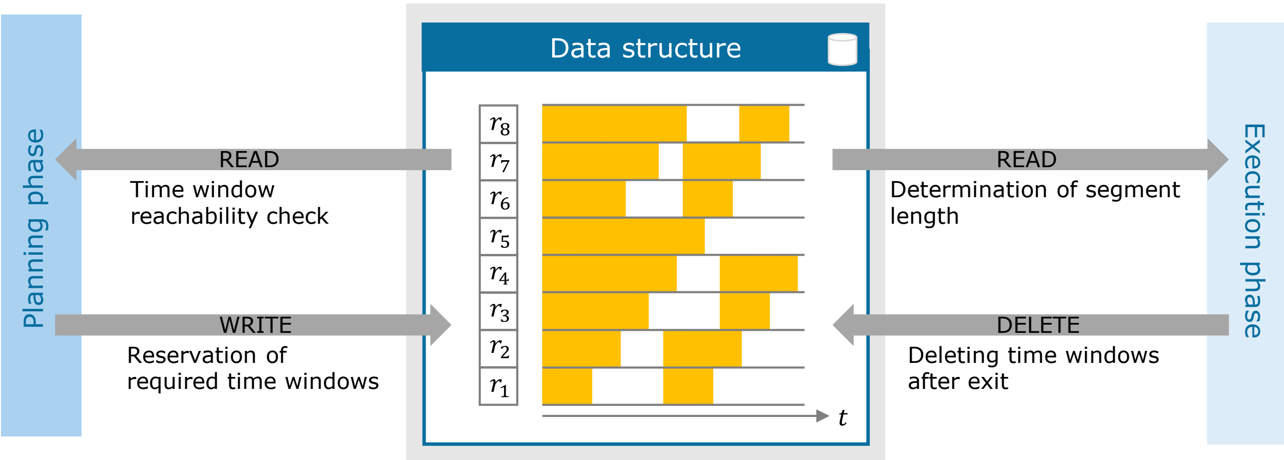

Figure 16: Read and write accesses to the data structure of the time windows with the modified routing algorithm

As can be seen, the data structure of the time windows is never write accessed during the execution phase. The previously described situation is hence prevented and robust execution of the calculated routes can be guaranteed.

5. Simulation Case Study

The described routing algorithm is implemented in a simulation-based tool for analyzing throughput of autonomous vehicle based storage and retrieval systems. The simulation environment *Tecnomatix Plant Simulation* is therefore used. In this section, we describe the results of a simulation study in which we consider an RMFS and vary the maximum segment length of straight movements. Figure 17 shows a visualization of a sample system.

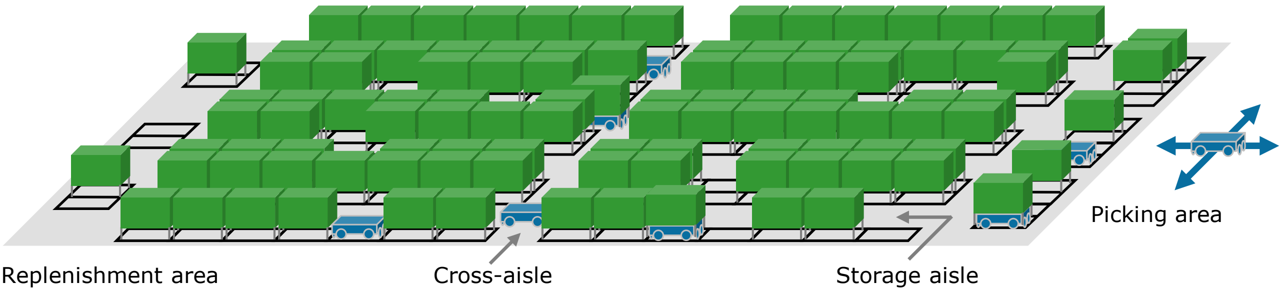

Figure 17: System example of an RMFS

The main section in the middle is used to store small racks (storage units) containing the storage items. They are assembled into two rows with aisles for vehicle movements in between. The AGVs can change aisles via cross-aisles. At one side of the system, there are picking locations to which the AGVs move racks for the picking process. If a rack is emptied, it has to be brought to the replenishment area on the opposite side of the layout. There, it will be refilled and subsequently stored at an empty storage location.

Maximum straight-movement length is a parameter affecting system performance. Short segments entail frequent AGV stops and accelerations. Longer segments enable AGVs to achieve maximum speed and reach their destinations with fewer intermediate stops. But in this case, the reserved time windows on the nodes are larger and the nodes are blocked longer for other vehicles.

Hence there exists a trade-off between segment length and reservation-time duration, which we analyze by performing a simulation study. Figure 18 shows a screenshot of the simulation model. The considered layout consists of six storage aisles, two cross-aisles, and four picking areas with five picking locations each.

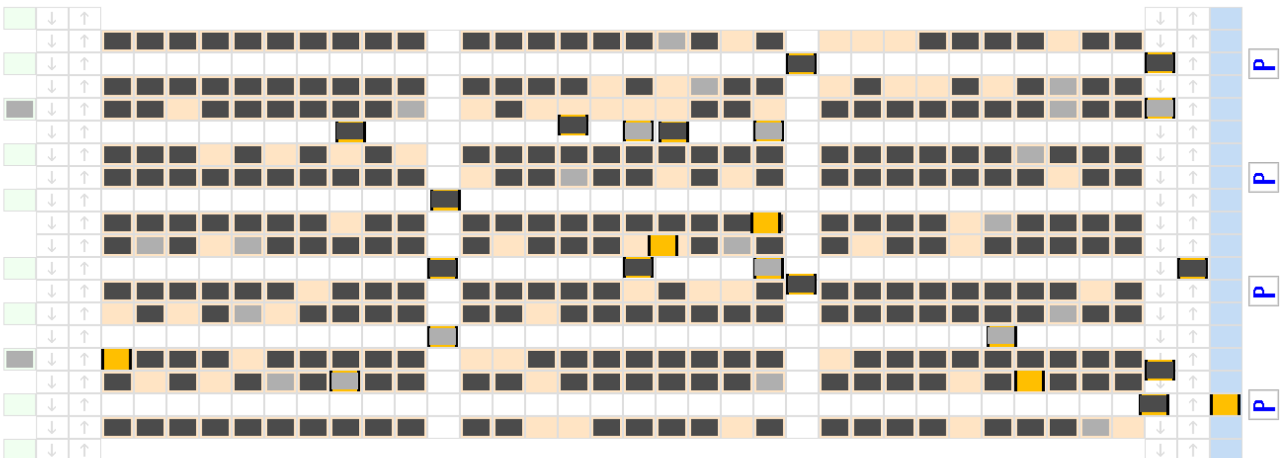

Figure 18: Simulation model of the considered layout

We vary the number of vehicles and the maximum segment length. Three observations with a simulation time of 12 hours are executed for each parameter configuration.

Figure 19 provides an overview of the simulation results. For all numbers of AGVs, the curves look similar. The performance is always worst with the maximum segment length "one" (which leads to a stop at each node). For greater maximum segment lengths, saturation is reached: greater maximum segment length no longer yields higher throughput. This behavior can be explained as follows: First, long segments are not always possible, because the next destination, a turning point, or a system boarder has been reached. Second, interference with routes already planned for other vehicles increases. Large time windows are necessary to extend a straight movement. These are hard to find, especially with a large number of AGVs in the system.

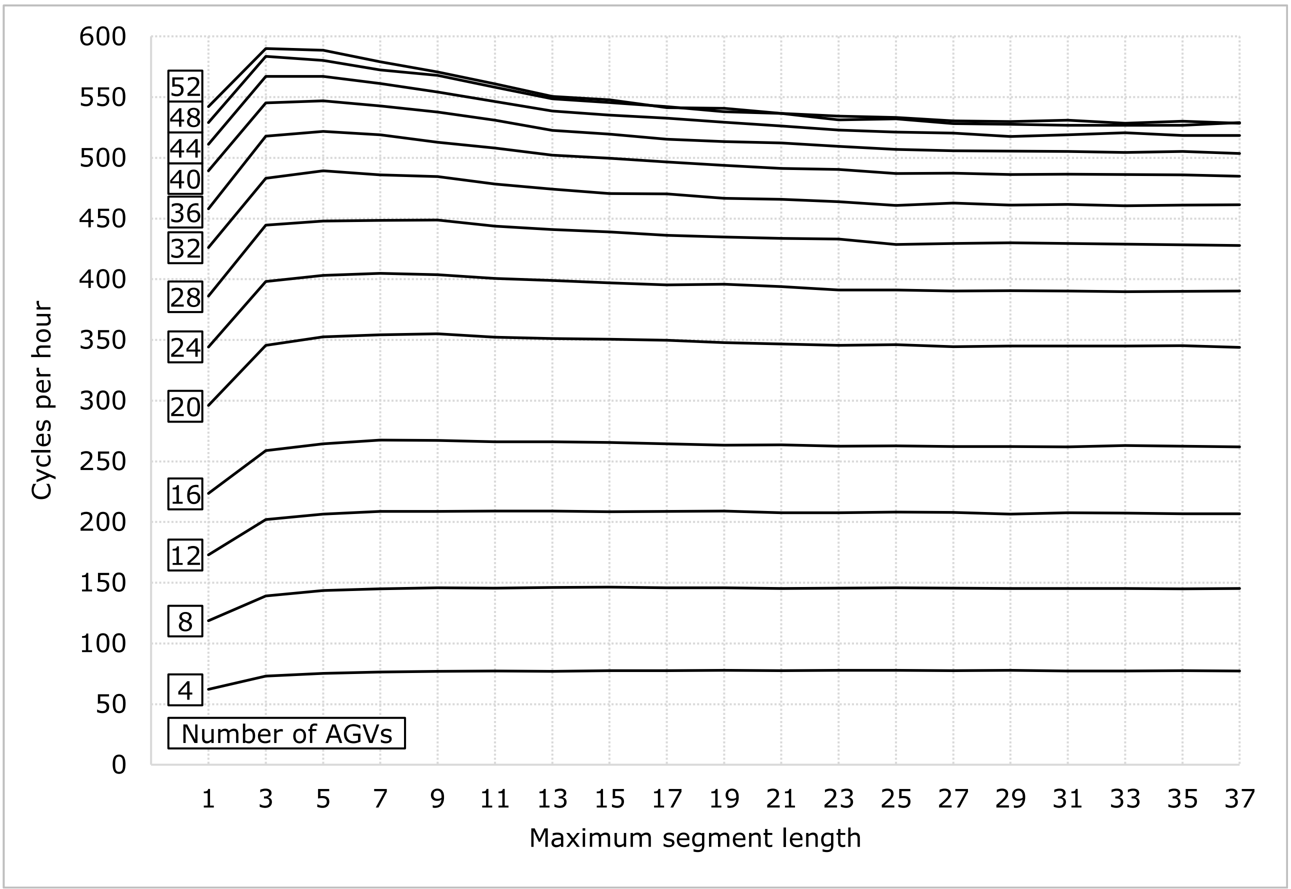

Figure 19: System throughput depending on the number of AGVs and the maximum segment length

A more specific look into the results yields figure 20. The chart provides information about which maximum segment length leads to the greatest throughput for each number of AGVs. For a small number of AGVs it is recommendable to allow longer movements.

Interference between vehicles is relatively infrequent under those conditions. In contrast, if a larger number of AGVs is operating in the system, greater throughput can be achieved using shorter segments.

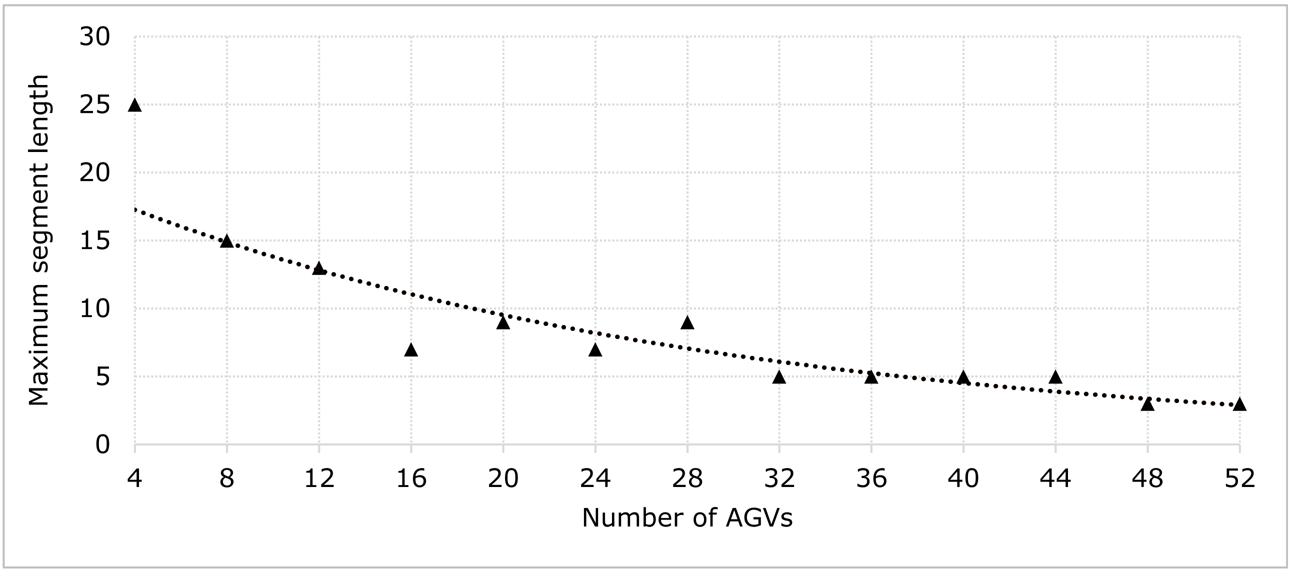

Figure 20: Optimal segment lengths for each number of AGVs and the respective trend line

The results discussed are valid only for the described system using a specific parameter set (e.g., speed, acceleration, and deceleration). In general the throughput of a vehicle-based automated storage and retrieval system can only be determined exactly by completing a simulation study. However, the impacts of the segments' lengths and the described trade-off between short and long segments will always influence the achievable throughput. In further work, we suggest analyzing different parameter settings, different layouts, and even different systems to gain more detailed information about the dependencies.

6. Summary

In this contribution, we identified the time window routing method as an option for deadlock avoidance in AGVSs. We described a problem that arises during execution of calculated routes and presented a solution approach that allows the consideration of acceleration and deceleration processes for planning time window routes. Applying modified time window routing can guarantee robust execution of the calculated routes. Finally, we conducted a simulation study to analyze the impact of different maximum segment lengths on an RMFS's achievable throughput. We showed that the optimal maximum segment length depends on the number of vehicles operating in the system.

We would like to conclude by remarking that one ends with the conventional time-window routing method if the maximum segment length is set to "one" and acceleration and deceleration to infinity.

References

Kim, C. W. & Tanchoco, J. (1991): Conflict-free shortest-time bidirectional AGV routeing. International Journal of Production Research 29 (12): 2377–2391.

Lamballais, T., Roy, D. & De Koster M.B.M. (2017): Estimating performance in a Robotic Mobile Fulfillment System. European Journal of Operational Research 256 (3): 976–990.

Le-Anh, T. & De Koster M. B. M. (2006): A review of design and control of automated guided vehicle systems. European Journal of Operational Research 171 (1): 1–23.

Lienert, T. & Fottner, J. (2017a): No more deadlocks – applying the time-window routing method to shuttle systems. Proceedings of 31. European Conference on Modelling and Simulation: 169–175.

Lienert, T. & Fottner, J. (2017b): Development of a generic simulation method for the time window routing of automated guided vehicles. Tagungsband 13. Fachkolloquium der WGTL: 307–322.

Liu, F.-H. & Hung, P.-C. (2001): Real-time deadlock-free control strategy for single multi-load automated guided vehicle on a job shop manufacturing system. International Journal of Production Research 39 (7): 1323–1342.

Maza, S. & Castagna, P. (2005): A performance-based structural policy for conflict-free routing of bi-directional automated guided vehicles. Computers in Industry 56 (7): 719-733.

ter Mors, A. W. (2009): The world according to MARP – Multi-Agent Route Planning. Dissertation, Technische Universiteit Delft.

VDI-Richtlinie 2510 (2005). Automated Guided Vehicle Systems. Beuth. Berlin.

Vis, I.F.A. (2006): Survey of research in the design and control of automated guided vehicle systems. European Journal of Operational Research 170 (3): 677–709.

Appendix

The nodes' exit times need to be calculated to extend a segment. In other words, when will a vehicle reach a certain position while moving from a start point to a destination point, taking acceleration, maximum speed, and deceleration into account? For simplicity's sake, we assume constant acceleration and deceleration. We furthermore suppose that acceleration equals deceleration in the formulas presented. The term $t(s,l)$ denotes the time needed to cover partial distance s while traversing overall distance l where $s \leq l$. This function is defined section-wise.

Two cases have to be distinguished. In one case, maximum speed is reached (figure 20) whereas in the other, the distance moved is too short to reach maximum speed (figure 21).

The function $s(t,l)$ can be derived from $v(t)$. The former's reversing function is the required $t(s,l)$ function.

s:	Partial distance
l:	Overall distance
t:	Time
$s(t,l)$:	Partial distance covered within time t while traversing overall distance l
$t(s,l)$:	Time needed to cover partial distance s while traversing overall distance l
v:	Speed
a:	Acceleration and deceleration (which are assumed equal)

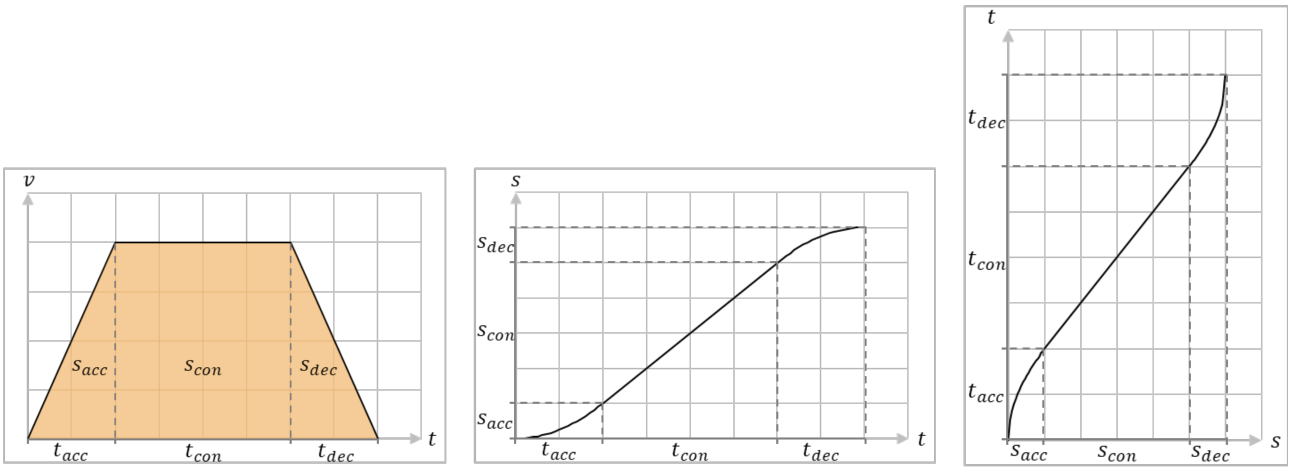

Figure 21: $v(t), s(t)$, and $t(s)$ for a movement during which the maximum speed is reached

$$s(t,l) = \begin{cases} \frac{1}{2}at^2 & ,t < \frac{v}{a} \\ -\frac{1}{2}\frac{v^2}{a} + vt & ,\frac{v}{a} \le t \le \frac{l}{v} \\ \frac{1}{2}\frac{v^2}{a} - \frac{1}{2}a\frac{l^2}{v^2} + vt - \frac{1}{2}at^2 + at\frac{l}{v} & ,\frac{l}{v} < t \end{cases} \tag{7}$$

$$t(s,l) = \begin{cases} \sqrt{\frac{2s}{a}} & ,s < \frac{1}{2}av^2 \\ \frac{1}{2}\frac{v}{a} + \frac{s}{v} & ,\frac{1}{2}av^2 \le s \le l - \frac{1}{2}\frac{v^2}{a} \\ \frac{l}{v} + \frac{v}{a} - \sqrt{\frac{2}{a}(l-s)} & ,l - \frac{1}{2}\frac{v^2}{a} < s \end{cases} \tag{8}$$

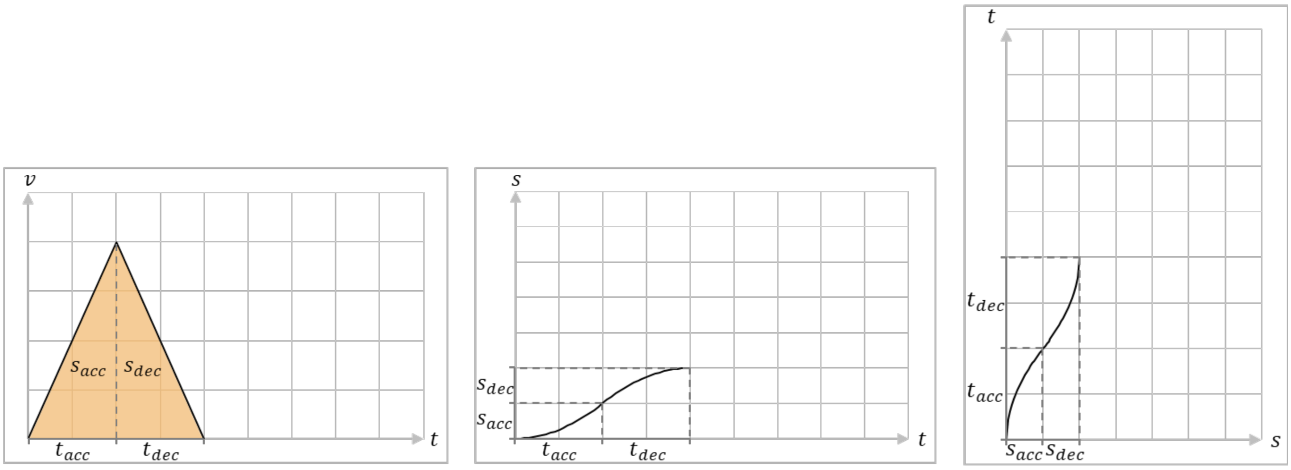

Figure 22: $v(t), s(t)$, and $t(s)$ for a movement during which the maximum speed is not reached

$$s(t,l) = \begin{cases} \frac{1}{2}at^2 & ,t < \sqrt{\frac{l}{a}} \\ 2t\sqrt{al} - \frac{1}{2}at^2 - l & ,t \ge \sqrt{\frac{l}{a}} \end{cases} \tag{9}$$

$$t(s,l) = \begin{cases} \sqrt{\frac{2s}{a}} & , s < \frac{l}{2} \\ 2\sqrt{\frac{l}{a}} - \sqrt{\frac{2}{a}(l-s)} & , s \geq \frac{l}{2} \end{cases} \tag{10}$$